DIPLOMARBEIT

A-POSTERIORI FEHLERSCHÄTZUNG FÜR DIFFERENTIALGLEICHUNGEN HÖHERER ORDNUNG

Ausgeführt am Institut für
Analysis und Scientific Computing, E101
der Technischen Universität Wien

unter Anleitung von
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Winfried Auzinger

durch
Lukas Exl

Stolberggasse 1-3/14
1050 Wien

Wien, am Unterschrift (Student)

Vorwort

Den Ausgangspunkt der vorliegenden Diplomarbeit bildet die Arbeit [1] '*Efficient collocation schemes for singular boundary value problems*' von W. Auzinger, O. Koch und E. Weinmüller, worin eine Fehlerschätzung für den globalen Fehler von Kollokationsverfahren als numerische Löser von singulären Randwertproblemen mit einer Singularität erster Art vorgestellt wird.
In dieser Arbeit wird die asymptotische Korrektheit der Fehlerschätzung im Falle nichtlinearer Systeme erster Ordnung mit linearen Randbedingungen bewiesen. Ausgehend von einem Kollokationsfehler der Ordnung m (Anzahl der Kollokationspunkte in jedem Teilintervall) beobachtet man hier die Ordnung $m + 1$ für den Fehler der Fehlerschätzung.

Das Ziel dieser Diplomarbeit war es, einen Fehlerschätzer für Kollokationslösungen linearer Randwertprobleme zweiter und speziell vierter Ordnung zu konstruieren.

Im Einführungsteil werden grundlegende Konvergenzresultate von Kollokationsverfahren vorgestellt, welche für die Beweise der Fehlerasymptotik benötigt werden.
Zu Beginn des zweiten Kapitels wird die Strategie der Fehlerschätzung gemäß [1] erläutert und die Asymptotik im linearen Fall bewiesen. Der Fehlerschätzer ist hier Lösung eines rückwärtigen Eulerverfahrens, mit 'integrierter Version' des punktweisen Defekts der Kollokationslösung bezüglich der gegebenen Differentialgleichung als Inhomogenität.
Dies bildet den Ausgangspunkt für die defektbasierte a-posteriori Fehlerschätzung linearer Zweipunkt-Randwertprobleme zweiter Ordnung, welche im folgenden behandelt wird. Hier ist es in Verallgemeinerung des Zuganges aus [1], der Defekt der Kollokationslösung bezüglich eines *exakten Differenzenschemas* (EDS), welcher als Inhomogenität eines einfachen Differenzenschemas

für den Fehlerschätzer auftritt.
Numerische Beispiele zeigen hier 'Kollokationsordnung+2' für den Fehler des Schätzers.
Im Anschluss wird das Verfahren auf Gleichungen vierter Ordnung verallgemeinert. Dazu wird zunächst das, für die Definition des Defektes, benötigte EDS und geeignete Diskretisierungen der Ableitungsrandbedingungen - erneut mit Hilfe eines EDS für die erste Ableitung - hergeleitet.
Numerische Beispiele zeigen auch hier 'Kollokationsordnung+2' für die Fehlerasymptotik.
Die entsprechenden Maple-Codes zu den numerischen Tests sind in Kapitel 2 zu finden.

Im Anhang wird die vorgestellte Methode, anstatt auf eine Kollokationslösung, auf die Finite-Differenzen-Lösung einer linearen Gleichung zweiter Ordnung mit Zwei-Punkt-Randbedingung angewandt. Im Unterschied zur Kollokation ist es hier erforderlich die Lösung zuerst geeignet zu interpolieren, um die gewünschte Fehlerordnung beobachten zu können.

Besonderer Dank ergeht an Ao.Univ.Prof. W. Auzinger für die hervorragende Betreuung, Bereitstellung von Materialien, hilfreiche Anregungen und Verbesserungsvorschläge.

Lukas Exl
Wien, April 2010

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Notation

Durchgehend wird mit $\mathbb{C}^n$ der Raum der komplexwertigen Vektoren der Dimension n bezeichnet und $|\ .\ |$,

$$|x| = \left|(x_1, x_2, \ldots, x_n)^T)\right| := \max_{1\leq i\leq n} |x_i|\,,$$

sei die Maximumsnorm in $\mathbb{C}^n$. Mit $C_n^p[0,1]$ wird der Raum der komplexwertigen, $p-$mal stetig differenzierbaren Funktionen auf [0,1] bezeichnet. Für Funktionen $y \in C_n^p[0,1]$ sei die Maximumsnorm durch

$$\|y\|_\infty := \max_{0\leq t\leq 1} |y(t)|$$

definiert.
$C_{n\times n}^p[0,1]$ bezeichne den Raum der $n \times n$ Matrizen mit Spalten in $C_n^p[0,1]$. Für $A \in C_{n\times n}^0[0,1]$ (Einträge a_{ij}) wird dadurch die Matrixnorm

$$\|A\|_{[0,1]} := \max_{t\in[0,1]} \|A(t)\|_\infty = \max_{t\in[0,1]} \left[\max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}(t)|\right]$$

induziert.

Für die numerische Analysis werden Gitter der Form

$$\Delta := (\tau_0, \tau_1, \ldots, \tau_N),$$

$h_i := \tau_{i+1} - \tau_i, i = 0 \dots N-1, \tau_0 = 0, \tau_N = 1, \mathbf{h} := \max_{0 \le i \le N-1} h_i$, und der zugehörige Gittervektor

$$u_\Delta := (u_0, u_1, \dots, u_N) \in \mathbb{C}^{(N+1)n}$$

definiert. Die Norm auf dem Raum der Gittervektoren sei gegeben durch

$$\|u_\Delta\|_\Delta := \max_{0 \le i \le N} |u_i| .$$

Für eine stetige Funktion $y(t) \in C[0,1]$ wird die punktweise Projektion auf den Raum der Gittervektoren durch

$$R_\Delta(y) := (y(\tau_0), y(\tau_1), \dots, y(\tau_N))$$

definiert.

Für die Kollokation wird jedes Teilintervall $J_i := [\tau_i, \tau_{i+1}]$ durch m zusätzliche Punkte äquidistant [1] unterteilt, was auf das feine Gitter

$$\Delta^m := \{t_{i,j} : t_{i,j} = \tau_i + j\delta_i, i = 0 \dots N-1, j = 0 \dots m+1\}$$

führt, wobei $\delta_i := \frac{h_i}{m+1}$. Zweckmäßigerweise wird τ_i auch mit $t_{i,0} \equiv t_{i-1,m+1}$, $i = 1 \dots N$ bezeichnet. Für ein Gitter Δ^m wird $u_{\Delta^m}, \|u_{\Delta^m}\|$ und R_{Δ^m} entsprechend den obigen Definitionen für das Gitter Δ definiert.

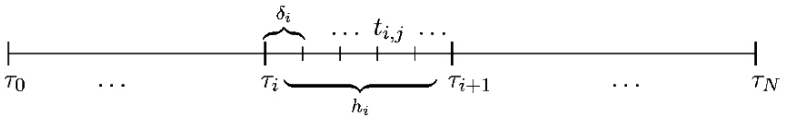

Abbildung 1.1: Das Gitter Δ^m.

[1] Die äquidistante Unterteilung ist für das Folgende nicht wesentlich; wir beschränken uns hier jedoch auf diese Wahl.

1.2 Kollokation

In diesem Abschnitt werden klassische Resultate der numerischen Lösung von (1.1) durch Kollokationsverfahren (siehe dazu auch [2, S.218-226]) und entsprechende Resultate der Kollokationslösung bei Gleichungen höherer Ordnung erwähnt (siehe hierzu auch [2, S.244-257]) .

Wir betrachten zuerst das Randwertproblem für ein System von n gewöhnlichen Differentialgleichungen 1. Ordnung mit linearen Randbedingungen:

$$\begin{aligned} &y'(t) = f(t, y(t)) \qquad t \in [0,1], \\ &B_0 y(0) + B_1 y(1) = \beta, \end{aligned} \tag{1.1}$$

wobei y und f vektorwertige Funktionen der Dimension n bezeichnen, und $B_0, B_1 \in \mathbb{R}^{r \times n}, \beta \in \mathbb{R}^r, r \leq n$ konstant sind. f wird durchgehend als ausreichend glatt angenommen.

Sei das Gitter Δ^m vorgegeben [2] und die Kollokationsfunktion $p(t) := p_i(t), t \in J_i, i = 0 \ldots N-1$, p_i Polynome vom Grad $\leq m$, welche folgende Bedingungen erfüllen:

$$\begin{aligned} &p_i'(t_{i,j}) = f(t_{i,j}, p_i(t_{i,j})), \qquad i = 0 \ldots N-1, \quad j = 1 \ldots m, \\ &p_i(\tau_i) = p_{i-1}(\tau_i), \qquad\qquad i = 0 \ldots N-1, \\ &B_0 p_0(0) + B_1 p_{N-1}(1) = \beta. \end{aligned} \tag{1.2}$$

Die erste Gleichung lässt sich in Verbindung zu m-stufigen Runge-Kutta Verfahren bringen. Durch Integration der Lagrange-Interpolierenden von p_i' über ein Intervall $[\tau_i, s]$ erhält man aus (1.2) (wir benutzen hier die Schreibweise $p(t_{i,j}) = p_{i,j}$):

$$\frac{p_{i,j} - p_{i,j-1}}{\delta_i} = \sum_{k=1}^{m} \alpha_{j,k} f(t_{i,k}, p_{i,k}), \quad j = 1 \ldots m \tag{1.3}$$

[2] Wir betrachten hier nur den Fall äquidistanter, innerer Kollokationsknoten $t_{i,j} = \tau_i + j\delta_i,\ j = 1 \ldots m$. Andere Varianten (nicht äquidistant bzw. mit Kollokation an einem Randpunkt) sind jedoch mit entsprechender Modifikation realisierbar.

für jedes Teilintervall J_i. [3] Die Koeffizienten $\alpha_{j,k}$ sind hierbei gegeben durch:

$$\alpha_{j,k} = \begin{cases} m\ a_{1,k}, & j = 1 \\ m(a_{j,k} - a_{j-1,k}), & j = 2 \dots m, \end{cases} \tag{1.4}$$

wobei die $a_{j,k}$ den Einträge im Butcher-Array des äquivalenten Runge-Kutta-Verfahrens entsprechen, (siehe dazu [2, S.69 und S.214]).
Die Gleichung (1.3) zusammen mit den Stetigkeits- und Randbedingungen in (1.2) bezeichnet man als *Kollokationsbedingungen in Integralschreibweise* und sind mit (1.2) äquivalent (siehe dazu [2, S.218]).

Definition 1. *Sei das Gitter Δ^m gegeben. Eine Kollokationslösung von* (1.1) *ist eine stückweise stetige Polynomfunktion $p(t)$ vom Grad $\leq m$, welche* (1.2) *erfüllt.*

Definition 2. *Eine Lösung von* (1.1) *heißt isoliert, falls die 'linearisierte, homogene Version' von* (1.1)

$$\begin{aligned} &z' = J(t)z, && t \in [0,1], \\ &B_0 z(0) + B_1 z(1) = 0, \end{aligned}$$

wobei $J(t) = J(t, y(t)) = \frac{\partial f(t,y(t))}{\partial y}$ die Jacobi Matrix von f bezeichnet, die eindeutige Lösung $z(t) \equiv 0$ hat.

Das folgende Konvergenzresultat ist für die späteren Betrachtungen von Bedeutung (siehe dazu [2, S.224-226]):

Satz 1. *Sei $z(t)$ die isolierte Lösung von* (1.1)*. Dann gibt es zu jedem Kollokationsverfahren der Form* (1.2) *Konstanten $\rho, h_0 > 0$, sodass für alle Gitter Δ^m mit $\boldsymbol{h} \leq h_0$ gilt:*

1. *Es existiert eine eindeutige Lösung $p(t)$ von* (1.2) *in einer Umgebung von Radius ρ um $z(t)$.*

[3] Die Summe in (1.3) entspricht einer Quadraturformel der Gestalt

$$\sum_{k=1}^{m} \alpha_{j,k}\ g(t_{i,k}) \approx \int_{\tau_{i-1}}^{\tau_i} g(\tau) d\tau.$$

Diese ist exakt für Polynome g vom Grad $< m$, also für $g = p'$. Siehe dazu auch Abschnitt 2.2.1.

2. *Diese Lösung kann mithilfe des Newton-Verfahrens berechnet werden und konvergiert quadratisch, sofern der Anfangsschätzer $p^0(t)$ hinreichend nahe an $z(t)$ ist.*

3. *Die folgenden Fehlerabschätzungen gelten:*

$$\begin{aligned} \|R_\Delta(p) - R_\Delta(z)\|_\Delta &= O(\boldsymbol{h}^{m+\nu}), \\ \|p - z\|_\infty &= O(\boldsymbol{h}^{m+\nu}), \\ \left\|p^{(l)} - z^{(l)}\right\|_\infty &= O(\boldsymbol{h}^{m+1-l}), \qquad l = 1 \dots m, \end{aligned} \tag{1.5}$$

wobei $\nu = 0$, falls m gerade und $\nu = 1$, falls m ungerade.

□

Im Folgenden sei m gerade gewählt, also $\nu = 0$.

Wir betrachten nun eine Gleichung der Ordnung k mit einer linearen 2-Punkt-Randbedingung:

$$\begin{aligned} &y^{(k)}(t) = f(t, y(t), ..., y^{(k-1)}(t)) \qquad t \in [0,1], \\ &B_0\mathbf{y}(0) + B_1\mathbf{y}(1) = \beta, \end{aligned} \tag{1.6}$$

wobei y und f skalarwertige Funktionen bezeichnen, $\mathbf{y} = (y, y', ..., y^{k-1})$ und $B_0, B_1 \in \mathbb{R}^{r\times k}, \beta \in \mathbb{R}^r, r \leq k$ konstant sind. f wird durchgehend als glatt angenommen.

Anmerkung. (1.6) lässt sich umschreiben zu einem äquivalenten System 1. Ordnung (1.1) für $y \in \mathbb{R}^k$. Wir beschäftigen uns im Weiteren jedoch mit der direkten Behandlung von (1.6) in der gegebenen Gestalt.

Eine Kollokationsmethode für (1.6) entspricht der Berechnung einer stückweisen Polynomfunktion $p(t) := p_i(t), t \in J_i, i = 0 \dots N-1$, p_i Polynome vom Grad $= m + k - 1$, welche an den Gitterpunkten stetige Ableitungen bis zur Ordnung $k-1$ hat (also $(p, p', \dots, p^{(k-1)}) \in C[0,1]$), die Randbedingungen erfüllt und der Differentialgleichung an den Kollokationspunkten genügt. Letzteres ergibt mN Gleichungen, die zusammen mit den $k(N-1)$ Stetigkeitsbedingungen der Ableitungen bis zur Ordnung $k-1$ und den k Randbedingungen, ein System für die $N(m+k)$ unbekannten Koeffizienten von $p(t)$

bestimmen.
Es gilt ein, dem Satz 1, entsprechendes Resultat (siehe [2, S.253- 257]), wobei anstatt von (1.5) gilt:

$$\begin{aligned} &\|R_\Delta(p) - R_\Delta(z)\|_\Delta = O(\mathbf{h}^m), \\ &\|p - z\|_\infty = O(\mathbf{h}^m), \\ &\left\|p^{(l)} - z^{(l)}\right\|_\infty = O(\mathbf{h}^{m+k-l}) + O(\mathbf{h}^m), \quad l = 1 \ldots m + k - 1. \end{aligned} \tag{1.7}$$

Es werden also Ableitungen bis zur Ordnung k der Differentialgleichung zumindest mit der Ordnung $O(\mathbf{h}^m)$ approximiert.

Kapitel 2

A-posteriori-Fehlerschätzung

2.1 Systeme 1. Ordnung

Die hier betrachtete Methode zur Schätzung des globalen Fehlers bei Kollokation für das analytische Problem (1.1) basiert auf Ideen von P. E. Zadunaisky (1976, [8]) und H. J. Stetter (1978, [9]). Hierfür wird ein zu (1.1), in ihren Grundzügen, benachbartes Problem betrachtet, dessen exakte Lösung und somit der Fehler, bei Lösen mittels eines numerischen Verfahrens niedriger Ordnung, bekannt ist. Der Fehler der Kollokation wird sodann mithilfe des Fehlers des benachbarten Problems, wie wir später sehen werden, von Ordnung mindestens $O(\mathbf{h}^{m+1})$ geschätzt.
Die Ideen aus [8] und [9] sind jedoch geeignet zu modifizieren, damit die Fehlerschätzung in der hier vorliegenden Situation erfolgreich funktioniert. Wir beschreiben zunächst diese Modifikation gemäß [1] für Systeme 1. Ordnung.

Definition 3. *Der punktweise Defekt $d(t)$ für (1.1) der Kollokationslösung $p(t)$ ist gegeben durch*

$$d(t) := p'(t) - f(t, p(t)). \tag{2.1}$$

Sei nun das zu (1.1) verwandte Problem

$$\begin{aligned} &y'(t) = f(t, y(t)) + d(t) \qquad t \in [0,1], \\ &B_0 y(0) + B_1 y(1) = \beta. \end{aligned} \tag{2.2}$$

gegeben.
Nach Konstruktion ist $p(t)$ die exakte Lösung von (2.2). Nehmen wir den

Fehler von $R_{\Delta^m}(p)$ als klein an, so können wir erwarten, dass sich auch die Lösungen von (1.1) und (2.2) nur gering voneinander unterscheiden. Die Information aus (2.2) wird dazu verwendet, um den Fehler von $R_{\Delta^m}(p)$ zu schätzen. Dafür lösen wir (1.1) und (2.2) mittels eines numerischen Verfahrens niedriger Ordnung (das sogenannte *Hilfsverfahren*), wie z.B. dem rückwärtigen Eulerverfahren:

$$\frac{\xi_{i,j} - \xi_{i,j-1}}{\delta_i} = f(t_{i,j}, \xi_{i,j}), \quad \text{und} \tag{2.3}$$

$$\frac{\pi_{i,j} - \pi_{i,j-1}}{\delta_i} = f(t_{i,j}, \pi_{i,j}) + \widetilde{d}_{i,j}, \tag{2.4}$$

$$\widetilde{d}_{i,j} := \frac{p(t_{i,j}) - p(t_{i,j-1})}{\delta_i} - \sum_{k=1}^{m+1} \alpha_{j,k} f(t_{i,k}, p(t_{i,k})), \tag{2.5}$$

mit $i = 0 \ldots N-1, j = 1 \ldots m+1$ und zugehörige Randbedingungen.
Hier ist $\widetilde{d}_{i,j}$ eine lokal gemittelte Version von Def. 3 (Integralmittel). Die $\alpha_{j,k}$ entsprechen denen in (1.4), allerdings mit $m+1$ verwendeten Kollokationspunkten, d.h. (2.5) ist der Defekt bezüglich der aufintegrierten Version (im Sinne von (1.3)) bezüglich eines genaueren Kollokationsverfahrens, bei dem der rechte Intervallendpunkt als Kollokationspunkt mit eingeht.
Ein algorithmischer Vorteil der Verwendung von $\widetilde{d}_{i,j}$ in (2.4) anstatt $d(t_{i,j})$ liegt darin, dass ja $d(t)$ an den Kollokationspunkten verschwindet, wie aus (1.2) und Definition 3 ersichtlich ist, und daher dort keine Information trägt.[1] Dies macht auch den Unterschied zum Verfahren mittels klassischer Definition des Defekts (siehe dazu auch dem Beweis von Satz 2 nachstehende Bemerkung).
Weiters wurde in obiger Definition von $\widetilde{d}_{i,j}$ für das Kollokationspolynom p die Bedingung (1.2) auch für den rechten Randpunkt $\tau_{i+1} = t_{i,m+1}$ gefordert.
Wir ersetzen nun den Fehler von ξ_{Δ^m} für (1.1) durch den Fehler von π_{Δ^m} für (2.2) und erhalten dadurch für den gesuchten Fehler $R_{\Delta^m}(p) - R_{\Delta^m}(z)$:

$$\begin{aligned} R_{\Delta^m}(p) - R_{\Delta^m}(z) &= (R_{\Delta^m}(p) - \xi_{\Delta^m}) + (\xi_{\Delta^m} - R_{\Delta^m}(z)) \\ &\approx (R_{\Delta^m}(p) - \xi_{\Delta^m}) + (\pi_{\Delta^m} - R_{\Delta^m}(p)) \qquad (2.6) \\ &= \pi_{\Delta^m} - \xi_{\Delta^m}. \end{aligned}$$

Dies zeigt, wenn auch nur heuristisch, dass $\pi_{\Delta^m} - \xi_{\Delta^m}$ eine erwartungsgemäß gute Schätzung für den globalen Fehler der Lösung höherer Ordnung $R_{\Delta^m}(p)$

[1] Hätte man jedoch die Kollokationsgleichungen nur näherungsweise gelöst (z.B. durch abgebrochene Newton-Iteration), so wäre es natürlich, die so entstehenden Defekte $d(t_{i,j})$ in die Berechnung von $\widetilde{d}_{i,j}$ mit einzubeziehen.

ist. Wir werden die Fehlerasymptotik des Fehlerschätzers im Falle eines linearen Randwertproblems beweisen (siehe Satz 2).

Für den Beweis des nachfolgenden Satzes benötigen wir noch zwei Resultate:

Lemma 1. *('Stufenordnung' der Kollokationsgleichung in Integralschreibweise, (siehe [3])) Sei $z(t)$ hinreichend oft differenzierbar, dann gilt die Ordnungsaussage:*

$$\frac{z(t_{i,j}) - z(t_{i,j-1})}{\delta_i} = \sum_{k=1}^{m+1} \alpha_{j,k} f(t_{i,k}, z(t_{i,k})) + O(\boldsymbol{h}^{m+1}) \tag{2.7}$$

für alle $i = 1 \dots N-1, j = 1 \dots m+1$.

Beweis. Dies ist nichts anderes als die Asymptotik der zugrundeliegenden Quadraturformel:
Betrachtet man ein Teilintervall J_i und sei $q(t)$ das Polynom vom Grad $\leq m$, welches $z'(t)$ an den Punkten $t_{i,j},\ j = 1 \dots m+1$ von J_i interpoliert. Für die Lagrange- Interpolation gilt bekanntlich (siehe z.B. [5])

$$\|q - z'\|_\infty = O(\mathbf{h}^{m+1}).$$

Dann gilt

$$\begin{aligned}
&\tfrac{z(t_{i,j})-z(t_{i,j-1})}{\delta_i} - \textstyle\sum_{k=1}^{m+1} \alpha_{j,k} f(t_{i,k}, z(t_{i,k})) \\
&= \tfrac{1}{\delta_i} \textstyle\int_{t_{i,j-1}}^{t_{i,j}} z'(t)dt - \sum_{k=1}^{m+1} \alpha_{j,k} z'(t_{i,k}) \\
&= \tfrac{1}{\delta_i} \textstyle\int_{t_{i,j-1}}^{t_{i,j}} z'(t)dt - \sum_{k=1}^{m+1} \alpha_{j,k} q(t_{i,k}) \\
&= \tfrac{1}{\delta_i} \textstyle\int_{t_{i,j-1}}^{t_{i,j}} (z'-q)(t)dt = \tfrac{1}{\delta_i}\delta_i O(\mathbf{h}^{m+1}).
\end{aligned}$$

□

Vom Hilfsverfahren wird Stabilität gefordert:

Lemma 2. *(Stabilität des rückwärtigen Euler- Verfahrens) Es bezeichne π_{Δ^m} die Lösung von* (2.3)*. Für das gestörte Schema*

$$\frac{\widetilde{\pi}_{i,j} - \widetilde{\pi}_{i,j-1}}{\delta_i} = f(t_{i,j}, \widetilde{\pi}_{i,j}) + \sigma_{i,j}, \quad i = 0 \dots N-1, \quad j = 1 \dots m+1$$

$$B_0 \widetilde{\pi}_{0,0} + B_1 \widetilde{\pi}_{N-1,m+1} = \beta,$$

existieren Konstanten S und σ_0, sodass $\widetilde{\pi}_{\Delta^m}$ eindeutig existiert, mit

$$\|\widetilde{\pi}_{\Delta^m} - \pi_{\Delta^m}\|_{\Delta^m} \leq S \|\sigma_{\Delta^m}\|_{\Delta^m} \tag{2.8}$$

für Störungen σ_{Δ^m} mit $\|\sigma_{\Delta^m}\|_{\Delta^m} \leq \sigma_0$ und Gitter Δ^m mit $\boldsymbol{h} \leq h_0$.

Beweis. Der Beweis stützt sich auf die Annahme, dass das zugrundeliegende Randwertproblem korrekt gestellt und stabil (gut konditioniert) ist, und mit Hilfe der Konsistenz des Euler-Verfahrens kann man dessen Stabilität dazu in (asymptotische) Beziehung setzen. Der Stabilitätsbeweis für Kollokationsmethoden funktioniert in ähnlicher Weise, siehe [2]. □

Wir betrachten nun konkret das folgende lineare Randwertproblem erster Ordnung:

$$\begin{aligned} &y' = f(t, y(t)) := A(t)y(t) + g(t), \qquad t \in [0, 1], \\ &B_0 y(0) + B_1 y(1) = \beta, \end{aligned} \tag{2.9}$$

wobei y und g vektorwertige Funktionen der Dimension n bezeichnen, A eine glatte $n \times n$ Matrix und $B_0, B_1 \in \mathbb{R}^{r \times n}, \beta \in \mathbb{R}^r, r \leq n$ konstant sind.

Wegen der Linearität von f kann (2.3) und (2.4) mittels $\eta_{\Delta^m} := \pi_{\Delta^m} - \xi_{\Delta^m}$ zu

$$\frac{\eta_{i,j} - \eta_{i,j-1}}{\delta_i} = A(t_{i,j})\eta_{i,j} + \widetilde{d}_{i,j} \tag{2.10}$$

mit homogenen Randbedingungen zusammengefasst werden.
Es folgt die zuvor angesprochene Asymptotik der Fehlerschätzung im linearen Fall. Der Beweis von Satz 2 ist eine Spezialisierung der Argumente aus [1] für den linearen Fall.

Satz 2. *Es habe* (2.9) *eine genügend glatte eindeutige Lösung. Dann gilt die folgende Abschätzung:*

$$\|(R_{\Delta^m}(p) - R_{\Delta^m}(z)) - \eta_{\Delta^m}\|_{\Delta^m} = O(\boldsymbol{h}^{m+1}). \tag{2.11}$$

Beweis. Im Folgenden verwenden wir die Kurzschreibweise $u_{i,j} := u(t_{i,j})$ für Gitterfunktionen.
Bezeichnen wir weiters mit $\nu_{\Delta^m} := (R_{\Delta^m}(p) - R_{\Delta^m}(z) - \eta_{\Delta^m})$ die Abweichung, d.h. den Fehler des Fehlerschätzers, der abzuschätzen ist, dann ist

$$\frac{\nu_{i,j}-\nu_{i,j-1}}{\delta_i}=\frac{p_{i,j}-p_{i,j-1}}{\delta_i}-\frac{z_{i,j}-z_{i,j-1}}{\delta_i}-(A(t_{i,j})\eta_{i,j}+\widetilde{d}_{i,j})=\dots$$

Mit (2.7) in Lemma 1 und der Definition des Defekts in (2.5) erhalten wir für obigen Ausdruck

$$\dots=\sum_{k=1}^{m+1}\alpha_{j,k}A(t_{i,k})(p_{i,k}-z_{i,k})+O(\mathbf{h}^{m+1})-A(t_{i,j})\eta_{i,j}=\dots$$

Nun folgt wegen $A(t_{i,j})\eta_{i,j}=A(t_{i,j})(p_{i,j}-z_{i,j})-A(t_{i,j})\nu_{i,j}$ weiter

$$\dots=A(t_{i,j})\nu_{i,j}-[A(t_{i,j})(p_{i,j}-z_{i,j})-\sum_{k=1}^{m+1}\alpha_{j,k}A(t_{i,k})(p_{i,k}-z_{i,k})+O(\mathbf{h}^{m+1})].$$

Unser Ziel ist es nun die Inhomogenität [.] durch $O(\mathbf{h}^{m+1})$ abzuschätzen, um mit dem Stabilitätsargument aus Lemma 2 den Beweis zu schließen.
Durch Taylorentwicklung in t sehen wir ($\Delta_{j,k}(t):=t_{i,k}-t_{i,j}$)

$$A(t_{i,k})(p_{i,k}-z_{i,k})=A(t_{i,j})(p_{i,j}-z_{i,j})+(A'(t_{i,j})(p_{i,j}-z_{i,j})+A(t_{i,j})(p'_{i,j}-z'_{i,j}))\Delta_{j,k}(t)$$

Wegen $\sum_{k=1}^{m+1}\alpha_{j,k}=1$ gilt insgesamt für die Inhomogenität [.]

$$[.]=(A'(t_{i,j})(p_{i,j}-z_{i,j})+A(t_{i,j})(p'_{i,j}-z'_{i,j}))\sum_{k}\Delta_{j,k}(t)+O(\mathbf{h}^{m+1}) \quad (2.12)$$

und somit

$$|(2.12)|\leq K\delta_i\max_{\tau_i\leq t\leq\tau_{i+1}}|A'(t)(p-z)(t)+A(t)(p'-z')(t)|+O(\mathbf{h}^{m+1}) \quad (2.13)$$

mit einer Konstante K. Aufgrund von Glattheitsannahmen an A und z folgt

$$\|(2.12)\|_\infty\leq\widetilde{K}\mathbf{h}\left(\|p-z\|_\infty+\|p'-z'\|_\infty\right)+O(\mathbf{h}^{m+1})=O(\mathbf{h}^{m+1}) \quad (2.14)$$

mit $\widetilde{K}\propto\max\left\{\|A'(t)\|_{[0,1]},\|A(t)\|_{[0,1]}\right\}$.
Letzte Gleichheit in (2.14) folgt aus den Fehlerabschätzungen (1.5) aus Satz 1. Insbesondere haben wir die Tatsache ausgenutzt, dass auch $p'-z'$ die volle Ordnung $\mathbf{h}^{m+1}$ besitzt.
Aus der Stabilität des rückwärtigen Eulerverfahrens (Lemma 2) folgt jetzt unmittelbar (2.11), wobei in diesem Argument ν_{Δ^m} mit der Nullösung verglichen wird, welche sich durch Inhomogenität und Randwerte 0 ergibt. □

Bemerkung. Verwendet man in (2.10) die 'klassische' Definition 3 des Defekts, so würde in der Inhomogenität [.] der Ausdruck

$$(p' - z')(t_{i,j}) - \frac{(p-z)(t_{i,j}) - (p-z)(t_{i,j-1})}{\delta_i}.$$

auftreten. Als Folgerung hätte man in der Schranke (2.14) den zusätzlichen Term $\|p'' - z''\|_\infty = O(\mathbf{h}^{m-1})$ (siehe Satz 1 (1.5), $l = 2$), und insgesamt eine Ordnung verloren.
Dies zeigt auch die wesentliche Rolle der Definition des Defekts gemäß (2.5): Der Differenzenquotient in (2.5) ist analog zum Differenzenquotienten im Hilfsverfahren (2.3), (2.4) bzw. (2.10), und Inspektion des Bewises von Satz 2 zeigt, dass dadurch das Auftreten eines von $p'' - z''$ abhängigen Terms vermieden wurde.

2.2 Gleichungen höherer Ordnung

Gleichungen der Ordnung k können bekanntlich in äquivalente Systeme 1. Ordnung umgeschrieben werden.
Im Folgenden werden jedoch solche numerische Verfahren betrachtet, welche die Gleichung höherer Ordnung 'direkt' lösen, wie z.B. Kollokationsmethoden für welche (1.7) gilt.
Um den globalen Fehler bei Lösen von Randwertproblemen wie (1.6) durch Kollokationsmethoden analog zum Vorangegangen zu schätzen, verallgemeinern wir die Methode des letzten Abschnitts.

2.2.1 Probleme 2. Ordnung

Für die Definition des Defekts in (2.5) sind wir im Prinzip von der lokal integrierten Differentialgleichung ausgegangen, was zu einem *'exakten Differenzenschema'* führt:

$$\frac{z(t_{i,j} + \delta_i) - z(t_{i,j})}{\delta_i} = \int_0^1 z'(t_{i,j} + \zeta\delta_i)d\zeta = \int_0^1 f(t_{i,j} + \zeta\delta_i)d\zeta \qquad (2.15)$$

wobei für die rechte Seite von (1.1) die Kurzschreibweise $f(t) \cong f(t, z(t))$ verwendet wird und z die eindeutige Lösung von (1.1) bezeichnet.
Um den Defekt zu definieren, haben wir eine Quadratur auf ganz J_i für das rechte Integral (wobei z durch die Kollokationslösung p ersetzt wird) verwendet, von welcher wir Exaktheitsgrad $m+1$ gefordert haben, um die Gewichte der Quadraturformel zu bestimmen (diese entsprechen in der Notation des letzten Abschnitts den $\alpha_{j,k}$ in (2.5)).
Der Defekt $\widetilde{d}_{i,j}$ ist dann das Residuum beim Ersetzen von z durch die Kollokationslösung p in der Variante von (2.15), mit Quadratur für das Integral der rechten Seite f.

Wir gehen nun für eine Gleichung 2. Ordnung entsprechend vor:

$$\begin{aligned} y''(t) &= f(t, y(t), y'(t)) \qquad t \in [0,1], \\ y(0) &= \beta_1, \quad y(1) = \beta_2. \end{aligned} \tag{2.16}$$

Wir setzen des Weiteren die eindeutige Lösbarkeit von (2.16) voraus.
Um den Aufwand in der Notation gering zu halten, betrachten wir zunächst ein Intervall der Form $I = [t-h, t+h]$.
Die 2. Ableitung einer Funktion $u \in C^4$ an der Stelle t wird durch den symmetrischen Differenzenquotienten wie folgt approximiert:

$$u''(t) = \frac{u(t-h) - 2u(t) + u(t+h)}{h^2} + O(h^2) =: \Delta_h^2(u(t)) + O(h^2). \tag{2.17}$$

Es gilt folgender Zusammenhang:[2]

Lemma 3. *Für eine Funktion $u \in C^1(I) \cap C^2(I \setminus \{t\})$ gilt*

$$\Delta_h^2(u(t)) = \int_{\mathbb{R}} K_2(\zeta) u''(t+\zeta h) d\zeta = \int_{-1}^{1} K_2(\zeta) u''(t+\zeta h) d\zeta \tag{2.18}$$

mit $K_2(\zeta) = \max\{0, 1-|\zeta|\}$.

Beweis. Berechnung des Integrals über $[-1, 0]$ und $[0, 1]$ durch partielle Integration liefert:

$$\int_{\mathbb{R}} K_2(\zeta) u''(t+\zeta h) d\zeta = \int_{-1}^{0} (1+\zeta) u''(t+\zeta h) d\zeta + \int_{0}^{1} (1-\zeta) u''(t+\zeta h) d\zeta$$

[2] (2.18) ist ein Analogon zu (2.15), das man auch in folgender Form schreiben könnte:

$$\Delta_h^1(u(t)) = \frac{u(t+h) - u(t)}{h} = \int_0^1 K_1(\zeta) u'(t+\zeta h) d\zeta, \quad K_1(\zeta) \equiv 1 \text{ auf } [0,1].$$

$$= \frac{u'(t)}{h} - \frac{1}{h}\int_{-1}^{0} u'(t+\zeta h)d\zeta - \frac{u'(t)}{h} + \frac{1}{h}\int_{0}^{1} u'(t+\zeta h)d\zeta$$

$$= -\frac{1}{h^2}(u(t) - u(t-h)) + \frac{1}{h^2}(u(t+h) - u(t)) = \Delta_h^2(u(t)).$$

□

Bemerkungen.

1. Die Identität in Lemma 3 werden wir später auf das Kollokationspolynom anwenden. Man beachte deshalb, dass die Voraussetzung auf solche Funktionen zugeschneidert ist (Das Kollokationspolynom und seine Ableitung sind stetig auf $[0,1]$, die 2. Ableitung jedoch im allgemeinen nicht.)

2. Wir wollen festhalten, dass der Kern K_2 kompakten Träger hat (i.Z. $\mathrm{Tr}[K_2] = \{\zeta \in \mathbb{R} \mid K_2(\zeta) \neq 0\}$ kompakt), stetig ist und bis auf die Stellen ± 1 und 0 differenzierbar ist.

3. Es gilt weiters
$$K_2(\zeta) = \Delta_{h=1}^2(U(\zeta)), \tag{2.19}$$
wobei $U(\zeta) = \frac{|\zeta|}{2}$ die Fundamentallösung des Differentialoperators $Lu = u''$ ist, also für U gilt
$$\langle LU, \phi\rangle := \langle U, \phi''\rangle = \phi(0) =: \langle \delta_0, \phi\rangle \tag{2.20}$$
für alle $\phi \in C_0^\infty(\mathbb{R}) := \{\phi \in C^\infty(\mathbb{R}) \mid \mathrm{Tr}\,[\phi] \text{ kompakt}\}$.
U wird dabei, da eine $L^1_{loc}(\mathbb{R})$- Funktion, als reguläre Distribution verstanden, also
$$\langle U, \phi\rangle = \int_{\mathbb{R}} U(x)\phi(x)dx \quad \text{für} \quad \phi \in C_0^\infty(\mathbb{R}).$$
δ_0 bezeichnet die Dirac'sche Delta-Distribution mit Singularität an der Stelle 0. Man schreibt für (2.20) auch kurz $LU = \delta_0$ und man erhält sie einfach durch partielle Integration:
$$\langle LU, \phi\rangle = \int_{\mathbb{R}} U(x)\phi''(x)dx = \left[\frac{|x|}{2}\phi'(x)\right]_{-\infty}^{\infty} - \int_{\mathbb{R}} \frac{\mathrm{sgn}(x)}{2}\phi'(x)dx$$

$$= -\left[\frac{\mathrm{sgn}(x)}{2}\phi(x)\right]_{-\infty}^{0} - \left[\frac{\mathrm{sgn}(x)}{2}\phi(x)\right]_{0}^{\infty} = -\frac{1}{2}(-\phi(0) - \phi(0)) = \phi(0).$$

Die Identität in (2.19) ergibt sich nun einfach aus:

$$\Delta^2_{h=1}(U(\zeta)) = \frac{1}{2}(|\zeta+1| - 2|\zeta| + |\zeta-1|) = \max\{0, 1-|\zeta|\}\,.$$

Diese Bemerkungen werden die spätere Vorgehensweise für Differentialgleichungen von z.B. Ordnung 4 motivieren.

Ist das Gitter Δ nicht äquidistant (h_i nicht konstant), so kann man in Verallgemeinerung zu Obigem analog vorgehen. Sei hierzu ein Intervall $\widetilde{I} = [t-h_L, t+h_R]$ vorgegeben mit $h_L > 0, h_R > 0$.
Wir definieren

$$\widehat{h} := \frac{h_R + h_L}{2}, \quad h_L = \alpha\widehat{h},\ h_R = \beta\widehat{h}, \tag{2.21}$$

dann ist $\alpha + \beta = 2$ und $\alpha, \beta \in (0,2)$.
Es sei weiters

$$\Delta^2_{\alpha,\beta} u(t) := \frac{1}{\widehat{h}^2}\left[\frac{u(t+h_R)-u(t)}{\beta} - \frac{u(t)-u(t-h_L)}{\alpha}\right]. \tag{2.22}$$

Im Falle $h_L = h_R = h$ stimmt $\Delta^2_{\alpha,\beta}$ mit Δ^2_h überein.
Für $\alpha \neq \beta$ (das entspricht $h_L \neq h_R$) gilt $\Delta^2_{\alpha,\beta} u(t) - u''(t) = O(h)$, wie man mittels Taylorentwicklung sehen kann. Der globale Fehler bei Approximation der 2. Ableitung mittels obigem asymmetrischen Differenzenquotienten stellt sich allerdings wieder als $O(\mathbf{h}^2)$ heraus [3], was die Verwendung für unsere Methode rechtfertigt.
Mit der Bezeichnung $S_a u(t) := u(t+a)$ für den *Shiftoperator* schreibt sich (2.22) wie folgt:

$$\Delta^2_{\alpha,\beta} = \frac{1}{\widehat{h}^2}\left[\frac{1}{\beta}(S_{h_R} - I) - \frac{1}{\alpha}(I - S_{-h_L})\right]. \tag{2.23}$$

Mit (2.23) erhalten wir für den *formal adjungierten Operator* $(\Delta^2_{\alpha,\beta})^*$:

$$(\Delta^2_{\alpha,\beta})^* = \frac{1}{\widehat{h}^2}\left[\frac{1}{\beta}(S_{-h_R} - I) - \frac{1}{\alpha}(I - S_{h_L})\right], \tag{2.24}$$

wobei hier $(S_a)^* = S_{-a}$ eingeht.[4]
Mit $K_2^{\alpha,\beta}(\zeta) = (\Delta^2_{\widehat{h}=1})^* U(\zeta) = (\Delta^2_{\alpha=h_L,\ \beta=h_R})^* U(\zeta)$ (wobei wieder $U(\zeta)$ die

[3] siehe dazu [10] bzw. [11] für eine detailierte Analyse (optimale Ordnung; nicht äquidistante Gitter) bzw. [12] (allgemeine Randbedingungen; Matlab-Implementierung auf beliebigen Gittern)

[4] Die Bildung der Adjungierten erfolgt hier formal auf einem 'unendlichen' Gitter, bezüglich eines Skalarprodukts $\langle u, v\rangle = \int_{\mathbb{R}} u(\zeta)v(\zeta)d\zeta$.

Fundamentallösung von $Lu = u''$ ist) erhalten wir wieder einen stetigen Kern $K_2^{\alpha,\beta}$:

$$K_2^{\alpha,\beta}(\zeta) = \begin{cases} 1 + \frac{\zeta}{\alpha}, & \zeta \in [-\alpha, 0], \\ 1 - \frac{\zeta}{\beta}, & \zeta \in [0, \beta], \\ 0, & \text{sonst.} \end{cases} \tag{2.25}$$

Dieser Kern hat kompakten Träger $[-\alpha, \beta] \subset \mathbb{R}$, ist stetig und bis auf die Stellen $-\alpha,\ \beta$ und 0 differenzierbar.

Analog zu Lemma 3 gilt folgendes

Lemma 4. *Für eine Funktion $u \in C^1(\widetilde{I}) \cap C^2(\widetilde{I} \setminus \{t\})$ gilt*

$$\int_{-\alpha}^{\beta} K_2^{\alpha,\beta}(\zeta) u''(t + \zeta\widehat{h}) d\zeta = \Delta^2_{\alpha,\beta}(u(t)). \tag{2.26}$$

Beweis. Mittels partieller Integration berechnen wir

$$\begin{aligned} \int_{-\alpha}^{\beta} K_2^{\alpha,\beta}(\zeta) u''(t+\zeta\widehat{h}) d\zeta &= \int_{-\alpha}^{0} (1 + \frac{\zeta}{\alpha}) u''(t+\zeta\widehat{h}) d\zeta + \int_{0}^{\beta} (1 - \frac{\zeta}{\beta}) u''(t+\zeta\widehat{h}) d\zeta \\ &= \frac{u'(t)}{\widehat{h}} - \frac{1}{\widehat{h}\alpha} \int_{-\alpha}^{0} u'(t + \zeta\widehat{h}) d\zeta - \frac{u'(t)}{\widehat{h}} + \frac{1}{\widehat{h}\beta} \int_{0}^{\beta} u'(t + \zeta\widehat{h}) d\zeta \\ &= \frac{1}{\widehat{h}^2\beta}(u(t + \beta\widehat{h}) - u(t)) - \frac{1}{\widehat{h}^2\alpha}(u(t) - u(t - \alpha\widehat{h})) = \Delta^2_{\alpha,\beta}(u(t)). \end{aligned}$$

□

Bemerkungen.

- Für $h_L = h_R$ gilt aus Symmetriegründen $\Delta^2_{\alpha,\beta} = (\Delta^2_{\alpha,\beta})^*$. Wir hätten also auch schon $K_2(\zeta)$ mittels $(\Delta^2_h)^*$ definieren können.

- Um besser zu verstehen, warum der Kern mittels formal adjungierten Operator zu bilden ist, rechnen wir Lemma 4 für Funktionen $u \in C_0^\infty(\mathbb{R})$ nach:
  Wir bezeichnen im Folgenden kurz:

$$D := \Delta^2_{\widehat{h}=1} \quad \text{und} \quad D^* := (\Delta^2_{\widehat{h}=1})^* = ((\Delta^2_{\alpha,\beta})^*)_{\widehat{h}=1}. \tag{2.27}$$

Mit obigen Bezeichnungen schreiben wir für die linke Seite von (2.26)

$$\langle D^*U, Lu\rangle := \int_{\mathbb{R}} D^*U(\zeta)u''(t+\zeta\widehat{h})d\zeta.$$

Zunächst stellen wir fest, dass offenbar gilt:

$$\frac{1}{\widehat{h}^2}\langle LU, Du\rangle = \frac{1}{\widehat{h}^2}\langle \delta_0, Du\rangle$$

$$= \frac{1}{\widehat{h}^2}\left[\frac{u(t+(\zeta+\beta)\widehat{h})-u(t+\zeta)}{\beta} - \frac{u(t+\zeta)-u(t+(\zeta-\alpha)\widehat{h})}{\alpha}\right]_{\zeta=0}$$

$$= \Delta^2_{\alpha,\beta}(u(t)).$$

Da $LDu = DLu$ gilt, ergibt sich mittels partieller Integration:

$$\Delta^2_{\alpha,\beta}(u(t)) = \frac{1}{\widehat{h}^2}\langle LU, Du\rangle \overbrace{=}^{part.Int.} \langle U, LDu\rangle$$

$$= \langle U, DLu\rangle = \langle D^*U, Lu\rangle .$$

Die erhaltene Identität entspricht Lemma 4 für Funktionen $u \in C_0^\infty(\mathbb{R})$.

Um Notation zu sparen, beschreiben wir im Folgenden die defektbasierte Fehlerschätzung für den Fall eines äquidistanten Gitters Δ ($\delta_i \equiv \frac{h}{m+1}$). Im Falle eines nichtäquidistanten Gitters Δ, käme in folgenden Ausführungen Lemma 4 statt Lemma 3 zu tragen (Details: siehe [11]/[12]).

Die lokal integrierte Version von (2.16) über dem Intervall J_i mit Gewicht $K_2(\zeta)$ (*exaktes Differenzenschema (EDS)*) lautet nun

$$\Delta^2_{\delta_i}(z(t_{i,j})) = \int_{-1}^{1} K_2(\zeta)f(t_{i,j}+\zeta\delta_i)d\zeta, \tag{2.28}$$

wobei z wieder die exakte Lösung bezeichnet und die Kurzschreibweise $f(t) \cong f(t, z(t), z'(t))$ verwendet wurde.
Um den Defekt zu definieren verwenden wir das EDS (2.28), mit $z(t)$ ersetzt durch die Kollokationslösung $p(t)$, und approximieren das Integral auf der rechten Seite durch eine geeignete Quadraturformel. So erhalten wir:

$$\Delta^2_{\delta_i}(p(t_{i,j})) = \int_{-1}^{1} K_2(\zeta)f(t_{i,j}+\zeta\delta_i)d\zeta \approx Q_m(f(t_{i,j}, p_{i,j}, p'_{i,j})), \tag{2.29}$$

wobei $f(t)$ im Integral für $f(t, p(t), p'(t))$ steht.
$Q_m(f(t_{i,j}, p_{i,j}, p'_{i,j}))$ sei eine Quadratur von Exaktheitsgrad $m+1$ für obenstehendes Integral mit Gewichtsfunktion K_2. Für die Quadratur denkt man sich, analog zu (2.5), sämtliche Punkte in J_i beteiligt ($m+2$ Stück). Es stehe hier und im Weiteren '$\approx$' für eine (Quadratur-) Approximation der Ordnung $m+1$ oder höher. Naheliegend ist es hier, beide Randpunkte des Kollokationsintervalls als zusätzliche Quadraturknoten zu nützen. Man erhält damit eine Quadraturformel genau vom Genauigkeitsgrad $m+2$.
Wir verwenden interpolatorische Quadratur auf J_i, also

$$Q_m(f(t_{i,j}, p_{i,j}, p'_{i,j})) = \sum_{k=0}^{m+1} \alpha_{k,j} f(t_{i,k}) \tag{2.30}$$

mit Quadraturgewichten

$$\alpha_{k,j} = \int_{-1}^{1} K_2(\zeta) L_{i,k}(t_{i,j} + \zeta\delta_i) d\zeta, \quad j = 0 \dots m+1.$$

Für (2.30) wurde das Lagrange-Interpolationspolynom $q(t)$ exakt integriert ($\hookrightarrow$ Exaktheitsgrad $\geq m+1$):

$$q(t) = \sum_{k=0}^{m+1} L_{i,k}(t) f(t_{i,k}) \in \mathbb{P}_{m+1}, \tag{2.31}$$

$$L_{i,k}(t) = \prod_{\substack{l=0 \\ l \neq k}}^{m+1} \frac{t - t_{i,l}}{t_{i,k} - t_{i,l}}, \quad L_{i,k}(t_{i,l}) = \delta_{k,l}. \tag{2.32}$$

($\delta_{k,l}$ bezeichnet hier das Kronecker-Delta.)

Nun kann der *Defekt* der Kollokationslösung $R_{\Delta^m}(p)$ von (2.16) durch (2.29) wie folgt definiert werden:

$$D_{i,j} := \Delta^2_{\delta_i}(p_{i,j}) - Q_m(f(t_{i,j}, p_{i,j}, p'_{i,j})). \tag{2.33}$$

Im Folgenden lassen wir den Subscript δ_i weg und schreiben z.B. für (2.33) kurz $D_{\Delta^m} = \Delta^2 p_{\Delta^m} - Q_m f(p_{\Delta^m})$.

Bemerkung. Da Q_m mindestens Exaktheitsgrad $m+1$ hat und $\text{grad}(p'') = m-1$, gilt:

$$D_{\Delta^m} = Q_m(p''_{\Delta^m} - f(p_{\Delta^m})) = Q_m(d_{\Delta^m}),$$

wobei der *punktweise Defekt* d_{Δ^m} (vgl. mit (2.1) aus Definition 3) an den Kollokationspunkten verschwindet. In der Praxis muss also für die Berechnung von (2.33) der punktweise Defekt nur an den Teilungspunkten t_i ausgewertet werden.

Analog zu (2.3) und (2.4) wird nun das *Hilfsverfahren*[5] festgelegt:

$$\Delta^2 \xi_{\Delta^m} = f(\xi_{\Delta^m}), \quad \text{und} \tag{2.34}$$

$$\Delta^2 \pi_{\Delta^m} = f(\pi_{\Delta^m}) + D_{\Delta^m}. \tag{2.35}$$

Die in f auftretenden 1. Ableitungen denken wir uns z.B. durch einen symmetrischen Differenzenquotienten der Form:

$$y'(t) \approx \frac{y(t+h) - y(t-h)}{2h}$$

diskretisiert (damit ist Konsistenzordnung gleich 2 gewährleistet).

Für den Fehlerschätzer $E_{\Delta^m} := \pi_{\Delta^m} - \xi_{\Delta^m}$ gilt also das Differenzenschema:

$$\Delta^2 E_{\Delta^m} = f(\pi_{\Delta^m}) - f(\xi_{\Delta^m}) + D_{\Delta^m}, \tag{2.36}$$

welches sich für lineares f zu

$$\Delta^2 E_{\Delta^m} = f(E_{\Delta^m}) + D_{\Delta^m} \tag{2.37}$$

vereinfacht. Sowohl (2.36) als auch (2.37) gelten mit homogenen Randbedingungen.

Wir möchten nun die Wahl des Fehlerschätzers motivieren.
Mit (2.28) und einer entsprechenden Quadratur der rechten Seite, analog zur oben Beschriebenen, gilt für die exakte Lösung

$$\Delta^2 z_{\Delta^m} \approx Q_m f(z_{\Delta^m}),$$

und für die Kollokationslösung $R_{\Delta^m}(p) \cong p_{\Delta^m}$ nach (2.33)

$$\Delta^2 p_{\Delta^m} = Q_m f(p_{\Delta^m}) + D_{\Delta^m}.$$

Wir benutzen hier und im Folgenden die Kurzschreibweise $Q_m f(p_{\Delta^m}) \cong Q_m(f(t_{i,j}, p_{i,j}, p'_{i,j}))$ bzw. $Q_m(f(z_{i,j})) \cong Q_m(f(t_{i,j}, z_{i,j}, z'_{i,j}))$.

[5]Das einfachste Differenzenschema mit analoger Diskretisierung der zweiten Ableitung.

Für den zu schätzenden Fehler $e_{\Delta^m} := p_{\Delta^m} - z_{\Delta^m}$ erhalten wir demnach das Differenzenschema

$$\Delta^2 e_{\Delta^m} \approx (Q_m f(p_{\Delta^m}) - Q_m f(z_{\Delta^m})) + D_{\Delta^m}, \tag{2.38}$$

Wir sehen bereits, dass das Differenzenschema für den Fehlerschätzer (2.36) gewisse Ähnlichkeiten mit (2.38) aufweist. Im linearen Fall wollen wir nun plausibel machen, dass E_{Δ^m} ein geeigneter Schätzer für den globalen Fehler e_{Δ^m} ist.
Zunächst haben wir nach dem EDS (2.28):

$$\Delta^2 e_{\Delta^m} = \Delta^2 p_{\Delta^m} - \Delta^2 z_{\Delta^m} = \Delta^2 p_{\Delta^m} - \int_{-1}^{1} K_2(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta$$

$$= \left(Q_m f(p_{\Delta^m}) - \int_{-1}^{1} K_2(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta \right) + \overbrace{\left(\Delta^2 p_{\Delta^m} - Q_m f(p_{\Delta^m})\right)}^{=D_{\Delta^m}}$$

$$= (Q_m f(p_{\Delta^m}) - Q_m f(z_{\Delta^m})) + \left(Q_m f(z_{\Delta^m}) - \int_{-1}^{1} K_2(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta \right) + D_{\Delta^m}$$

$$= Q_m f(e_{\Delta^m}) + D_{\Delta^m} + O(h^{m+1}).$$

Man beachte, dass in der letzten Gleichheit zum einen die Linearität von f eingeht[6] (vergleiche hierzu auch mit (2.31)), und zum anderen wurde die Quadratur als $O(h^{m+1})$-Approximation angenommen.
Vergleicht man dies nun mit dem im linearen Fall erhaltenen Schema (2.37) so sieht man, dass für $\epsilon_{\Delta^m} := E_{\Delta^m} - e_{\Delta^m}$ offenbar gilt

$$\Delta^2 \epsilon_{\Delta^m} \approx f(E_{\Delta^m}) - Q_m f(e_{\Delta^m})$$

$$= f(\epsilon_{\Delta^m}) + [f(e_{\Delta^m}) - Q_m f(e_{\Delta^m})]. \tag{2.39}$$

Ist das Hilfsverfahren (vgl. (2.34)) stabil, die Quadratur Q_m, wie oben verwendet, eine Approximation der Ordnung $m+1$, sowie auch der Ausdruck [.] in (2.39), so ist auch ϵ zumindest von Ordnung $m+1$ (vgl. hierzu mit dem Beweis zu Satz 2, speziell die dort auftretende Inhomogenität [.])[7]

[6] nichtlinearer Fall: siehe [11]

[7] Details: Dissertation [11]; insbesondere ist der Term $[.] = f(e_{\Delta^m}) - Q_m f(e_{\Delta^m})$ sauber abzuschätzen, basierend auf bekannte Aussagen über die Approximationsqualität der Kollokationslösung, $e^{(l)} = O(h^m),\ l = 0, 1, 2.$

h	esterr	ord	const	collerr	collord
5.0000E-01	1.3113E-04	-	-	8.7302E-03	-
2.5000E-01	7.5092E-06	4.1262	2.29E-03	2.4726E-03	1.8199
1.2500E-01	4.7221E-07	3.9916	1.90E-03	6.3120E-04	1.9699
6.2500E-02	2.9673E-08	3.9922	1.90E-03	1.6045E-04	1.9759
3.1250E-02	1.8554E-09	3.9993	1.94E-03	4.0104E-05	2.0003
1.5625E-02	1.1588E-10	4.0010	1.95E-03	1.0027E-05	1.9999

Tabelle 2.1: Ordnung des Fehlers der Fehlerschätzung für Beispiel (2.40), $m = 2$.

h	esterr	ord	const	collerr	collord
5.0000E-01	9.2937E-06	-	-	3.3220E-04	-
2.5000E-01	1.5788E-07	5.8794	5.47E-04	2.2270E-05	3.8989
1.2500E-01	2.5363E-09	5.9600	6.12E-04	1.4311E-06	3.9599
6.2500E-02	3.9522E-11	6.0039	6.70E-04	8.9216E-08	4.0037
3.1250E-02	6.1398E-13	6.0083	6.78E-04	5.5492E-09	4.0070
1.5625E-02	9.5639E-15	6.0044	6.69E-04	3.4611E-10	4.0030

Tabelle 2.2: Ordnung des Fehlers der Fehlerschätzung für Beispiel (2.40), $m = 3$.

2.2.2 Numerische Beispiele

Wir wollen hier zunächst folgendes Beispiel einer linearen DGL 2. Ordnung betrachten:

$$\begin{aligned} &Lu = u'' + xu' + (1+x)u = g(x), \quad x \in [0,1], \\ &u(0) = u(1) = 0, \end{aligned} \tag{2.40}$$

mit der eindeutigen exakten Lösung: $z(x) = x(1-x)e^{-x^2}$, also $g = Lz$.

Tabelle 2.1 zeigt die Fehlerordnung für den Fall eines äquidistanten Gitters in $[0,1]$ und zwei Kollokationspunkten in jedem Teilintervall ($m = 2$).
Tabelle 2.2 entsprechend für den Fall dreier Kollokationspunkte ($m = 3$).

Für $m = 2$ zeigt sich zunächst Ordnung 2 für das verwendete Kollokationsverfahren (vgl. letzte Spalte von Tabelle 2.1, '*collord*').
Weiters beobachten wir in der 2. und 3. Spalte von Tabelle 2.1 ('*esterr*'

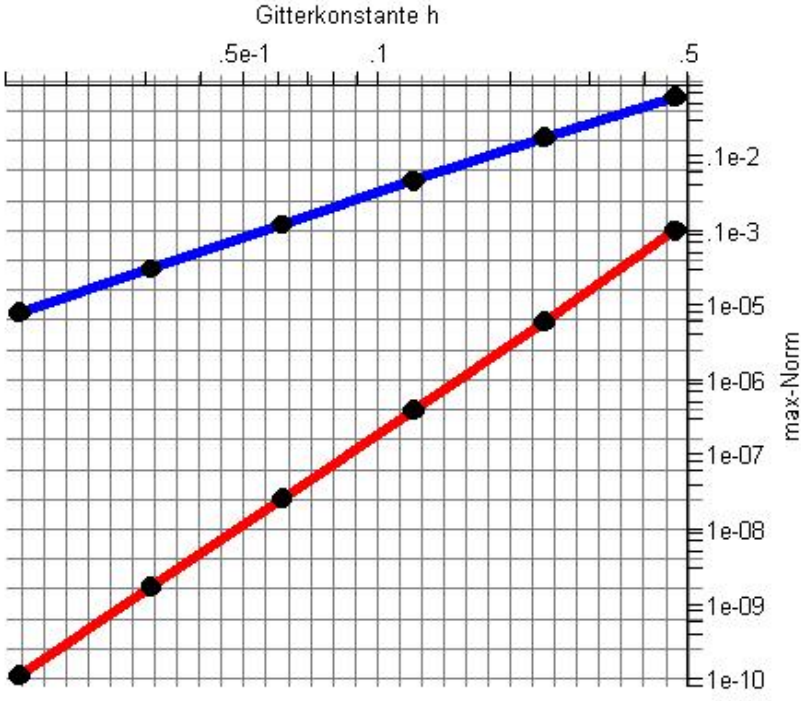

Abbildung 2.1: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot); $m = 2$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 2, die des Fehlerschätzers ca. 4.

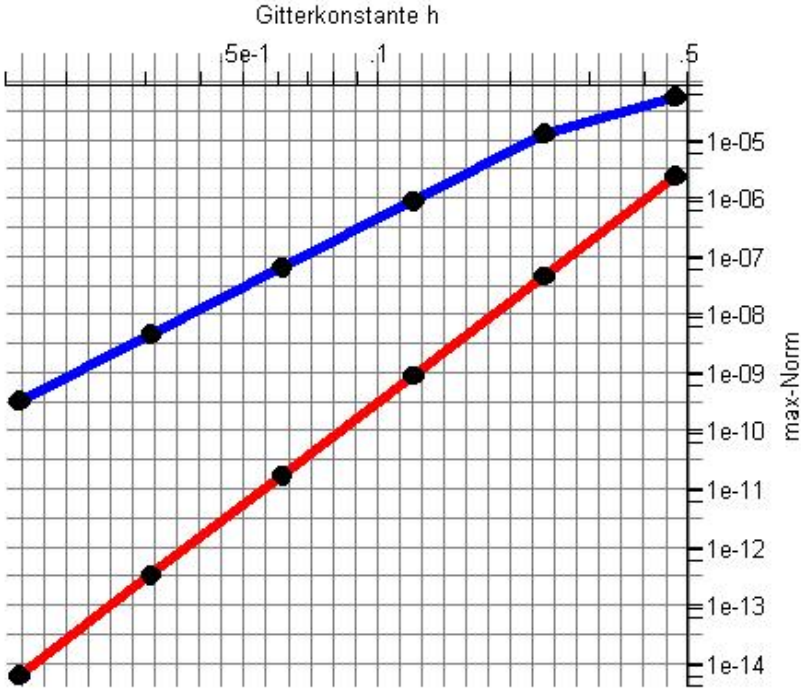

Abbildung 2.2: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot); $m = 3$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 4, die des Fehlerschätzers ca. 6.

bezeichnet den globalen Fehler der Fehlerschätzung in der Maximumsnorm, '*ord*' die zugehörige Fehlerordnung) sogar die Ordnung 4 ($m+2$).

Für $m = 3$ ist die Kollokationslösung sogar eine $O(h^{m+1})$ Approximation (siehe Tabelle 2.2). Wir beobachten in diesem Fall, dass der globale Fehler des Schätzers sogar Ordnung $m + 3 = 6$ besitzt (Tabelle 2.2).

Wir beobachten für den Fehlerschätzer die Ordnung 'Kollokationsordnung + 2'.[8]

2.2.3 Probleme 4. Ordnung

In diesem Abschnitt wollen wir die oben behandelte defektbasierte a-posteriori-Fehlerschätzung auf den Fall einer Differentialgleichung der Ordnung 4 verallgemeinern.
Wir betrachten zunächst ein einfaches Beispiel einer linearen DGL 4. Ordnung mit 2-Punkt Randbedingung:

$$\begin{aligned} &y^{(4)}(t) - c(t)y(t) = h(t), \quad t \in [0,1], \\ &y(0) = a_0, \quad y'(0) = b_0, \\ &y(1) = a_1, \quad y'(1) = b_1, \end{aligned} \tag{2.41}$$

mit $c(t) \geq 0, \quad \forall t \in [0,1]$.

Unter obigen Voraussetzungen hat die homogene Version von (2.41) ($h(t) \equiv 0$, homogene Randbedingungen) die eindeutige Lösung $y(t) \equiv 0$, da

$$y(t) = \sum_{j=0}^{3} C_j e^{\lambda_j t} \quad \text{mit} \quad C_j \in \mathbb{R} \quad \text{und} \quad \lambda_j = \sqrt[4]{c(t)} e^{i\frac{\pi}{2}j}, \; j = 0 \ldots 3,$$

wobei die Randbedingungen auf ein homogenes LGS für die Koeffizienten C_j mit regulärer Systemmatrix führen.
Die Green'sche Funktion $G(x,t)$ zu dem Problem (2.41) ist daher eindeutig auf $[0,1]^2$ definiert, und (2.41) (homogene RB) hat die eindeutige Lösung

$$z(t) = \int_0^1 G(x,t)h(x)dx \quad \text{(siehe dafür z.B. [2, S.96/97])}.$$

[8] Die komplette Theorie mit optimalen Abschätzen findet sich in [11], Matlab-Implementierung und weitere Beispiele in [12].

Bemerkung. Die Gleichung (2.41) entspricht dem 1-dimensionalen physikalischen Modell eines, an den Enden eingespannten und in einem Medium eingebetteten, Balkens unter Gewichtsbelastung.

Wir wollen nun in Analogie zum Fall der Ordnung 2 verfahren und einen entsprechenden Fehlerschätzer für die Kollokationslösung von (2.41) bei äquidistanten Gitter Δ herleiten.
Dazu betrachten wir den symmetrischen 5-Punkt Differenzenquotienten auf dem Intervall $I = [t-h, t+h]$:

$$\begin{aligned} u^{(4)}(t) &= h^{-4}\left[u(t+2h) - 4u(t+h) + 6u(t) - 4u(t-h) + u(t-2h)\right] + O(h^2) \\ &=: \Delta_h^4 u(t) + O(h^2), \end{aligned} \tag{2.42}$$

für eine Funktion $u \in C^6$.
Um für obiges Problem (2.41) ein exaktes Differenzenschema zu erhalten, berechnen wir mit der Fundamentallösung $U(\zeta) = \frac{|\zeta|^3}{12}$ des gewöhnlichen Differentialoperators $Lu = u^{(4)}$ den entsprechenden Kern $K_4(\zeta)$:

$$\begin{aligned} \Delta_{h=1}^4 U(\zeta) &= 1/12\left(|\zeta+2|^3 - 4\,|\zeta+1|^3 + 6\,|\zeta|^3 - 4\,|\zeta-1|^3 + |\zeta-2|^3\right) \\ &= \left\{ \begin{array}{ll} \frac{1}{6}\zeta^3 + \zeta^2 + 2\zeta + \frac{4}{3}, & \zeta \in [-2,-1], \\ -\frac{1}{2}\zeta^3 - \zeta^2 + \frac{2}{3}, & \zeta \in [-1,0], \\ \frac{1}{2}\zeta^3 - \zeta^2 + \frac{2}{3}, & \zeta \in [0,1], \\ -\frac{1}{6}\zeta^3 + \zeta^2 - 2\zeta + \frac{4}{3}, & \zeta \in [1,2], \\ 0, & |\zeta| \geq 2. \end{array} \right\} =: K_4(\zeta). \end{aligned} \tag{2.43}$$

Die Funktion $K_4(\zeta)$ ist in $C^2(\mathbb{R})$, die 2. Ableitung ist an $\pm 2, \pm 1$ und 0 jedoch nicht mehr differenzierbar.
Es gilt analog zu Lemma 3 das folgende

Lemma 5. *Für eine Funktion $u \in C^3(I) \cap C^4(I \setminus \{t, t+h, t-h\})$ gilt*

$$\int_{\mathbb{R}} K_4(\zeta) u^{(4)}(t+\zeta h) d\zeta = \Delta_h^4(u(t)). \tag{2.44}$$

Beweis. Die Identität folgt analog zum Beweis von Lemma 3 durch partielle Integration der linken Seite von (2.44) getrennt über die Intervalle $I_{1-} := [-2,1],\ I_{2-} := [-1,0],\ I_{2+} := [0,1]$ und $I_{1+} := [1,2]$:

$$\int_{I_{1\pm}} K_4(\zeta) u^{(4)}(t+\zeta h) d\zeta = \mp\frac{u^{(3)}(t\pm h)}{6h} - \frac{u^{(2)}(t\pm h)}{2h^2} \mp \frac{u'(t\pm h)}{h^3} + \frac{u(t\pm 2h) - u(t\pm h)}{h^4}.$$

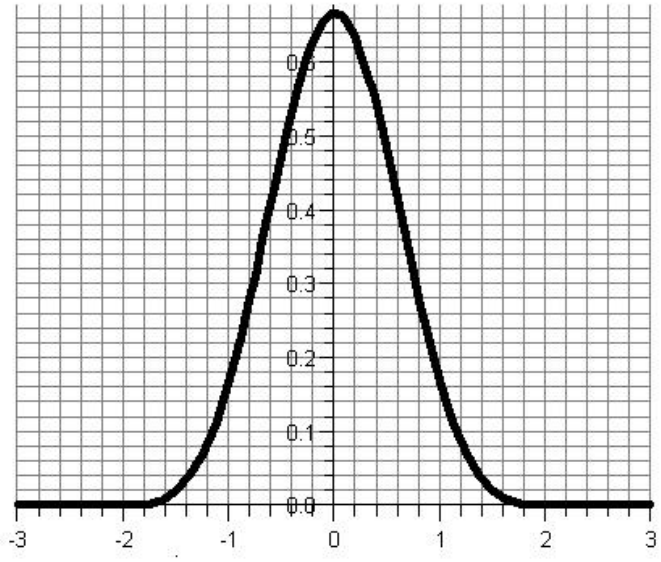

Abbildung 2.3: $K_4(\zeta)$

$$\int_{I_{2\pm}} K_4(\zeta)u^{(4)}(t+\zeta h)d\zeta = \frac{\mp 4u^{(3)}(t) \pm u^{(3)}(t \pm h)}{6h} + \frac{u^{(2)}(t \pm h)}{2h^2} + \frac{\pm 2u'(t) \pm u'(t \pm h)}{h^3} - \frac{3u(t \pm h) - 3u(t)}{h^4}.$$

Addition der vier Integrale ergibt $\Delta_h^4 u(t)$. □

Bemerkung. Die Voraussetzung an die Funktion u in Lemma 5 ist wieder derart, dass wir später (für die Definition des Defekts) in Identität (2.44) das Kollokationspolynom des Problems 4. Ordnung (2.41) einsetzen können (vgl. Kapitel *Kollokation*).
Weiters sieht man durch Einsetzen der Funktion $u(x) = \frac{x^4}{4!}$ in Lemma 5, dass gilt[9]:

$$\int_{-2}^{2} K_4(\zeta)d\zeta = 1.$$

[9]Der Differenzenquotient Δ_h^4 differenziert Polynome vom Grad 4 exakt; $\frac{d^4}{dx^4}\frac{x^4}{4!} = 1$.

Im Folgenden beschreiben wir die defektbasierte a-posteriori Fehlerschätzung des globalen Kollokationsfehlers von (2.41) für den Fall eines äquidistanten Gitters Δ ($\delta_i \equiv \frac{h}{m+1}$).

Für die Lösung $z(t)$ von (2.41) gilt nach Lemma 5 das *exakte Differenzenschema (EDS)*

$$\Delta^4_{\delta_i}(z(t_{i,j})) = \int_{-2}^{2} K_4(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta, \tag{2.45}$$

mit der Kurzschreibweise $f(t) \cong f(t, z(t)) \cong h(t) + c(t)z(t)$.

Wir verwenden (2.45), mit $z(t)$ ersetzt durch die Kollokationslösung $p(t)$ für die Defektdefinition, wobei wir das Integral auf der rechten Seite durch eine Quadraturformel approximieren. So erhalten wir:

$$\Delta^4_{\delta_i}(p(t_{i,j})) = \int_{-2}^{2} K_4(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta \approx Q_m(f(t_{i,j}, p_{i,j})), \tag{2.46}$$

mit einer interpolatorischen Quadratur, wo sämtliche Punkte in J_i verwendet werden. Die Quadratur ist dann mindestens von Genauigkeitsgrad $m+1$.

Nun kann der Defekt der Kollokationslösung $p(t)$ von (2.41) auf $\Delta^m \setminus \left\{0, \frac{h}{m+1}, 1 - \frac{h}{m+1}, 1\right\}$ definiert werden:

$$D_{i,j} := \Delta^4_{\delta_i}(p_{i,j}) - Q_m(f(t_{i,j}, p_{i,j})). \tag{2.47}$$

Im Folgenden lassen wir den Subscript δ_i weg und schreiben z.B. für (2.47) kurz $D_{\Delta^m} = \Delta^4 p_{\Delta^m} - Q_m f(p_{\Delta^m})$.

Bemerkung. Da Q_m mindestens Exaktheitsgrad $m+1$ hat und $\mathrm{grad}(p^{(4)}) = m-1$, gilt:

$$D_{\Delta^m} = Q_m(p^{(4)}_{\Delta^m} - f(p_{\Delta^m})) = Q_m(d_{\Delta^m}),$$

wobei der *punktweise Defekt* d_{Δ^m} (vgl. mit (2.1) aus Definition 3) an den Kollokationspunkten verschwindet. In der Praxis muss also für die Berechnung von (2.47) der punktweise Defekt nur an den Teilungspunkten t_i ausgewertet werden.

Bevor wir das Differenzenschema für den Fehlerschätzer festlegen können, müssen wir uns noch eine geeignete Diskretisierung der Ableitungsrandbedingungen in (2.41) überlegen. Die Idee ist, ein *exaktes Differenzenschema für die erste Ableitung* an den Intervallendpunkten zu konstruieren.
Um für den Ableitungswert $u'(0)$ einer Funktion $u(t)$ ein EDS zu erhalten, verwenden wir Taylorentwicklung bis zur Ordnung 3 mit Integralrestglied[10] :

$$u(h) = u(0)+hu'(0)+\frac{h^2}{2}u''(0)+\frac{h^3}{6}u^{(3)}(0)+\frac{h^4}{6}\int_0^1(1-\zeta)^3u^{(4)}(h\zeta)d\zeta. \quad (2.48)$$

$$u(2h) = u(0)+2hu'(0)+2h^2u''(0)+\frac{8h^3}{6}u^{(3)}(0)+\frac{8h^4}{3}\int_0^1(1-\zeta)^3u^{(4)}(2h\zeta)d\zeta. \quad (2.49)$$

$$u(3h) = u(0)+3hu'(0)+\frac{9h^2}{2}u''(0)+\frac{9h^3}{2}u^{(3)}(0)+\frac{27h^4}{2}\int_0^1(1-\zeta)^3u^{(4)}(3h\zeta)d\zeta. \quad (2.50)$$

Nun eliminieren wir in obigen Entwicklungen die Terme mit zweiter und dritter Ableitung $(18*(2.48)-9*(2.49)+2*(2.50))$ und erhalten:

$$\Delta_h^1u(0) := \frac{-11u(0)+18u(h)-9u(2h)+2u(3h)}{6h} = u'(0)+R_1(u^{(4)}), \quad (2.51)$$

mit

$$R_1(u^{(4)}) := \frac{h^3}{2}\int_0^1(1-\zeta)^3u^{(4)}(h\zeta)d\zeta$$

$$-4h^3\int_0^1(1-\zeta)^3u^{(4)}(2h\zeta)d\zeta+\frac{27h^3}{6}\int_0^1(1-\zeta)^3u^{(4)}(3h\zeta)d\zeta.$$

Bemerkung. Der Differenzenquotient $\Delta_h^1u(0)$ approximiert $u'(0)$ mit Ordnung 3.

Aus (2.51) folgern wir für die exakte Lösung $z(t)$ von (2.41) das *EDS*:

$$\Delta_h^1z(0) = z'(0)+R_1(z^{(4)}) = z'(0)+R_1(f(z(t))). \quad (2.52)$$

[10]Für eine hinreichend glatte Funktion $u(t)$ gilt die Taylorformel:

$$u(x) = \sum_{j=0}^{n}\frac{u^{(j)}(a)}{j!}(x-a)^j+\int_a^x\frac{(x-t)^n}{n!}u^{(n+1)}(t)dt$$

$$=\sum_{j=0}^{n}\frac{u^{(j)}(a)}{j!}(x-a)^j+\frac{(x-a)^{n+1}}{n!}\int_0^1(1-\xi)^nu^{(n+1)}(a+\xi(x-a))d\xi.$$

Analoges Vorgehen liefert auch ein entsprechendes *EDS* für $u'(1)$:

$$u(1-h) = u(1) - hu'(1) + \frac{h^2}{2}u''(1) - \frac{h^3}{6}u^{(3)}(1) - \frac{1}{6}\int_{1-h}^{1}(1-h-\zeta)^3 u^{(4)}(\zeta)d\zeta. \tag{2.53}$$

$$u(1-2h) = u(1) - 2hu'(1) + 2h^2u''(1) - \frac{8h^3}{6}u^{(3)}(1) - \frac{1}{6}\int_{1-2h}^{1}(1-2h-\zeta)^3 u^{(4)}(\zeta)d\zeta. \tag{2.54}$$

$$u(1-3h) = u(1) - 3hu'(1) + \frac{9h^2}{2}u''(1) + \frac{9h^3}{2}u^{(3)}(1) - \frac{1}{6}\int_{1-3h}^{1}(1-\zeta)^3 u^{(4)}(\zeta)d\zeta. \tag{2.55}$$

Dieselbe Linearkombination wie oben liefert:

$$\Delta_h^1 u(1) := \frac{11u(1) - 18u(1-h) + 9u(1-2h) - 2u(1-3h)}{6h} = u'(1) + R_2(u^{(4)}), \tag{2.56}$$

mit

$$R_2(u^{(4)}) := \frac{1}{2h}\int_{1-h}^{1}(1-h-\zeta)^3 u^{(4)}(\zeta)d\zeta$$

$$-\frac{1}{4h}\int_{1-2h}^{1}(1-2h-\zeta)^3 u^4(\zeta)d\zeta + \frac{1}{18h}\int_{1-3h}^{1}(1-3h-\zeta)^3 u^{(4)}(\zeta)d\zeta.$$

Bemerkung. Der Differenzenquotient $\Delta_h^1 u(1)$ approximiert $u'(1)$ mit Ordnung 3.

Aus (2.56) folgern wir für die exakte Lösung $z(t)$ von (2.41) das *EDS*:

$$\Delta_h^1 z(1) = z'(1) + R_2(z^{(4)}) = z'(1) + R_2(f(z(t))). \tag{2.57}$$

Wir definieren nun den *Defekt von* $p(t)$ *bezüglich des EDS* (2.52) *bzw.* (2.57) ($z'(0) = b_0,\ z'(1) = b_1$):

$$D_0 := \Delta_{\delta_i}^1 p(0) - \left[b_0 + Q_m(R_1(f(p(t))))\right], \tag{2.58}$$

$$D_1 := \Delta_{\delta_i}^1 p(1) - \left[b_1 + Q_m(R_2(f(p(t))))\right], \tag{2.59}$$

wobei wir interpolatorische Quadratur auf J_i verwenden.

Das Differenzenverfahren für den Fehlerschätzer $E_{\Delta^m} \approx R_{\Delta^m}(p-z)$ legen wir nun folgendermaßen fest:

$$\Delta^4 E_{\Delta^m} = f(E_{\Delta^m}) + D_{\Delta^m}, \tag{2.60}$$

$$\Delta^1 E_{\Delta^m}(0) = D_0, \tag{2.61}$$
$$\Delta^1 E_{\Delta^m}(1) = D_1, \tag{2.62}$$

mit $E_{0,0} = E_{N-1,m+1} = 0$.

Für den Kollokationsfehler $e_{\Delta^m} = R_{\Delta^m}(p - z) = p_{\Delta^m} - z_{\Delta^m}$ gilt nach (2.52)

$$\Delta^1 e_{\Delta^m}(0) = \Delta^1 p_{\Delta^m}(0) - \Delta^1 z_{\Delta^m}(0) = \Delta^1 p_{\Delta^m}(0) - \overbrace{z'(0)}^{=b_0} - R_1(f(z)), \tag{2.63}$$

und damit für den Fehler des Fehlerschätzers $\epsilon_{\Delta^m} = E_{\Delta^m} - e_{\Delta^m}$ am linken Rand

$$\Delta^1 \epsilon_{\Delta^m}(0) = -Q_m(R_1(f(p))) + R_1(f(z)) \tag{2.64}$$
$$= \underbrace{[Q_m(R_1(f(z))) - Q_m(R_1(f(p)))]}_{=Q_m(R_1(f(e)))} + \underbrace{[R_1(f(z)) - Q_m(R_1(f(z)))]}_{\|.\|_\infty = O(h^{m+1})}. \tag{2.65}$$

Wegen der h^3-Terme in R_1 und der Approximationsgüte der Kollokation $\|e\|_\infty = O(h^m)$ gilt insgesamt für die Inhomogenität in (2.64) eine $O(h^{m+1})$-Abschätzung in der Maximumsnorm [11]. Die zugrundeliegende Stabilität des verwendeten Differenzenquotienten lässt also zumindest auf Ordnung $m + 1$ für den Fehlerschätzer am linken Rand schließen.
Analog zeigt sich dieselbe Ordnung am rechten Rand.

Im Inneren gilt für $e_{\Delta^m} = R_{\Delta^m}(p - z) = p_{\Delta^m} - z_{\Delta^m}$ nach dem EDS (2.45)

$$\Delta^4 e_{\Delta^m} = \Delta^4 p_{\Delta^m} - \underbrace{\Delta^4 z_{\Delta^m}}_{=\int_{-2}^{2} K_4(\zeta) f(t_{i,j}+\zeta\delta_i) d\zeta} \tag{2.66}$$

$$= \underbrace{\left[\Delta^4 p_{\Delta^m} - Q_m f(p_{\Delta^m})\right]}_{=D_{\Delta^m}} + \left[Q_m f(p_{\Delta^m}) - \int_{-2}^{2} K_4(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta\right] \tag{2.67}$$

$$= Q_m f(e_{\Delta^m}) + \left[Q_m f(z_{\Delta^m}) - \int_{-2}^{2} K_4(\zeta) f(t_{i,j} + \zeta\delta_i) d\zeta\right] + D_{\Delta^m}. \tag{2.68}$$

Nimmt man das, der Quadraturformel zugrundeliegende, Lagrange-Interpolationspolynom q der rechten Seite f zu Hilfe[12], so sieht man, dass für den Quadraturfehler [.] gilt:

$$\|.\|_\infty = O(h^{m+2}) \int_{-2}^{2} K_4(\zeta) d\zeta = O(h^{m+2}). \tag{2.69}$$

[11] $\|Q_m(R_1(f(e)))\|_\infty = O(h^{m+3})$.
[12] Da $m + 2$ Stützstellen verwendet werden gilt $\|q - f\|_\infty = O(h^{m+2})$.

In der Notation des letzten Abschnitts gilt also

$$\Delta^4 e_{\Delta^m} \approx Q_m f(e_{\Delta^m}) + D_{\Delta^m}. \tag{2.70}$$

Für den Fehler der Fehlerschätzung ϵ_{Δ^m} gilt somit

$$\begin{aligned}\Delta^4 \epsilon_{\Delta^m} &\approx f(E_{\Delta^m}) - Q_m f(e_{\Delta^m}) \\ &= f(\epsilon_{\Delta^m}) + [f(e_{\Delta^m}) - Q_m f(e_{\Delta^m})].\end{aligned} \tag{2.71}$$

Wir wollen nun die Inhomogenität von (2.71) abschätzen. Dazu sei die Quadratur Q_m mit Gewichten $\alpha_{k,j}$ auf dem Teilintervall J_i durch

$$Q_m f(e_{\Delta^m}) = \sum_{k=0}^{m+1} \alpha_{k,j} f(e_{i,k}) \tag{2.72}$$

gegeben [13].
Damit erhalten wir mit Taylorentwicklung auf dem Intervall J_i

$$\begin{aligned}f(e_{i,j}) - Q_m f(e_{\Delta^m}) &= c_{i,j} e_{i,j} - \sum_{k=0}^{m+1} \alpha_{k,j} \underbrace{c_{i,k} e_{i,k}}_{= c_{i,j} e_{i,j} + (c'_{i,j} e_{i,j} + c_{i,j} e'_{i,j}) \Delta_{j,k}(t)} \\ &= c_{i,j} e_{i,j} - c_{i,j} e_{i,j} \sum_{k=0}^{m+1} \alpha_{k,j} - (c'_{i,j} e_{i,j} + c_{i,j} e'_{i,j}) \sum_k \alpha_{k,j} \Delta_{j,k}(t) \\ &= -(c'_{i,j} e_{i,j} + c_{i,j} e'_{i,j}) \sum_k \alpha_{k,j} \Delta_{j,k}(t),\end{aligned}$$

und somit wegen der Approximationsqualität der Kollokation insgesamt für die Inhomogenität in der Maximumsnorm

$$\|.\|_\infty = hK(\underbrace{\|e_{\Delta^m}\|_\infty}_{=O(h^m)} + \underbrace{\|e'_{\Delta^m}\|_\infty}_{=O(h^m)}) = O(h^{m+1}), \tag{2.73}$$

mit einer von h unabhängigen Konstante K.
Wir können nun mit der Stabilität des 5-Punkt Differenzenquotienten Δ^4 sehen, dass der konstruierte Fehlerschätzer E_{Δ^m} zumindest Ordnung $m+1$ besitzt[14].

[13] Es gilt $1 = \int_{-2}^{2} K_4(\zeta) d\zeta = Q_m 1 = \sum_{k=0}^{m+1} \alpha_{k,j}$.

[14] Wie bei Differentialgleichungen der Ordnung 2 zeigen numerische Beispiele die tatsächliche Ordnung $m+2$.

h	esterr	ord	const	collerr	collord
5.0000E-01	1.8533E-08	-	-	1.5696E-05	-
2.5000E-01	2.6354E-10	6.1359	1.08E-06	9.0719E-07	4.1128
1.2500E-01	3.9997E-12	6.0420	1.05E-06	5.5331E-08	4.0352
6.2500E-02	6.2030E-14	6.0109	1.04E-06	3.4365E-09	4.0091
3.1250E-02	9.6703E-16	6.0032	1.03E-06	2.1437E-10	4.0028
1.5625E-02	1.5102E-17	6.0008	1.04E-06	1.3392E-11	4.0006

Tabelle 2.3: Ordnung des Fehlers der Fehlerschätzung für das Problem (2.74), $m = 3$.

2.2.4 Numerische Beispiele

Wir wollen hier zunächst folgendes Beispiel einer DGL 4. Ordnung betrachten

$$\begin{aligned} &Lu = u^{(4)} - x(1-x)u = h(x), \quad x \in [0,1], \\ &u(0) = 0, \quad y'(0) = 1, \\ &y(1) = 0, \quad y'(1) = -e, \end{aligned} \tag{2.74}$$

mit der eindeutigen exakten Lösung $z(x) = x(1-x)e^x$, also $h(x) = Lz$.

Tabelle 2.3 und Abbildung 2.4 zeigen die Fehlerordnung für den Fall eines äquidistanten Gitters in $[0,1]$ und drei Kollokationspunkten in jedem Teilintervall ($m = 3$).
Tabelle 2.4 und Abbildung 2.5 entsprechend für den Fall von vier Kollokationspunkten ($m = 4$), Tabelle 2.5 und Abbildung 2.6 für den Fall $m = 5$.
Wir beobachten für die Ordnung des Fehlers des Fehlerschätzers 'Kollokationsordnung +2'.

Nun wollen wir folgendes Beispiel einer DGL 4. Ordnung betrachten:

$$\begin{aligned} &Lu = u^{(4)} - xu'' - x^2u' - x(1-x)u = h(x), \quad x \in [0,1], \\ &u(0) = 0, \quad y'(0) = 1, \\ &y(1) = 0, \quad y'(1) = -e, \end{aligned} \tag{2.75}$$

mit der eindeutigen exakten Lösung $z(x) = x(1-x)e^x$, also $h(x) = Lz$.

h	esterr	ord	const	collerr	collord
5.0000E-01	1.8531E-09	-	-	3.2753E-06	-
2.5000E-01	2.8798E-11	6.0078	1.18E-07	2.0406E-07	4.0046
1.2500E-01	4.4830E-13	6.0054	1.17E-07	1.2716E-08	4.0042
6.2500E-02	7.0013E-15	6.0008	1.18E-07	7.9422E-10	4.0010
3.1250E-02	1.0936E-16	6.0005	1.17E-07	4.9624E-11	4.0005
1.5625E-02	1.7086E-18	6.0002	1.17E-07	3.1014E-12	4.0002

Tabelle 2.4: Ordnung des Fehlers der Fehlerschätzung für das Problem (2.74), $m = 4$.

h	esterr	ord	const	collerr	collord
5.0000E-01	1.1298E-11	-	-	1.1298E-11	-
2.5000E-01	4.599E-14	8.0853	2.72E-09	4.1599E-14	6.0644
1.2500E-01	1.5932E-16	8.0285	2.67E-09	1.5932E-16	6.0210
6.2500E-02	6.1862E-19	8.0087	2.66E-09	6.1862E-19	6.0071
3.1250E-02	2.4132E-21	8.0019	2.65E-09	2.4132E-21	6.0015
1.5625E-02	9.4231E-24	8.0005	2.65E-09	9.4231E-24	6.0005

Tabelle 2.5: Ordnung des Fehlers der Fehlerschätzung für das Problem (2.74), $m = 5$.

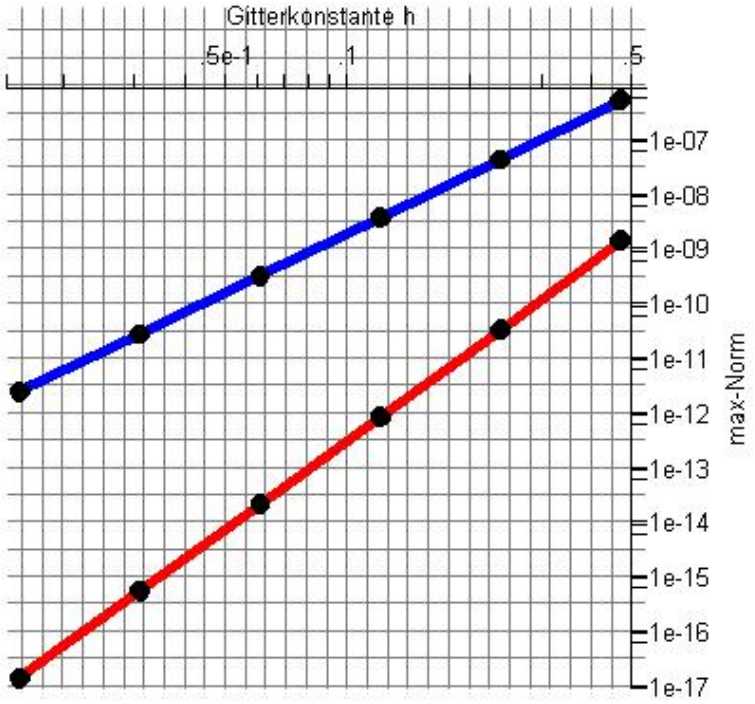

Abbildung 2.4: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot) für das Problem (2.74); $m = 3$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 4, die des Fehlerschätzers ca. 6.

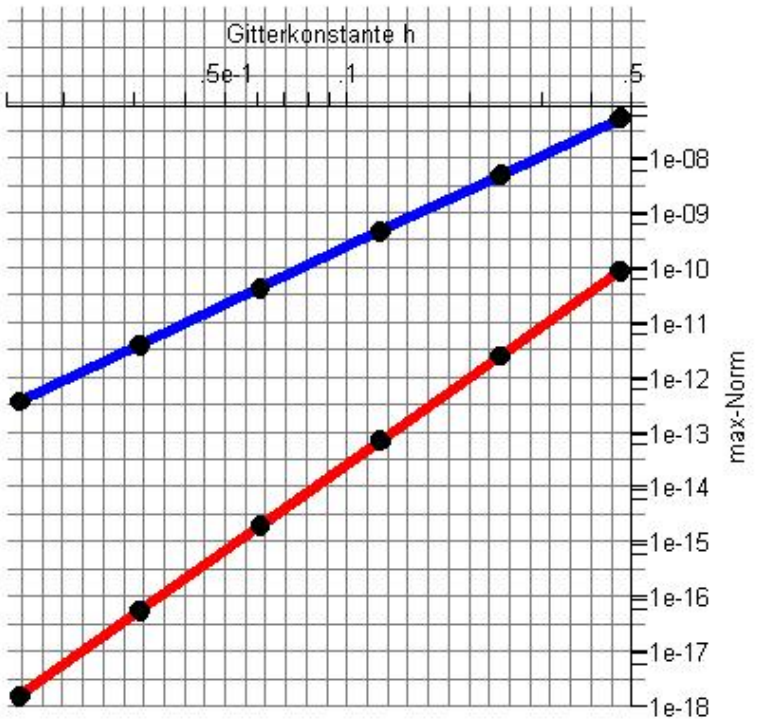

Abbildung 2.5: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot) für das Problem (2.74); $m = 4$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 4, die des Fehlerschätzers ca. 6.

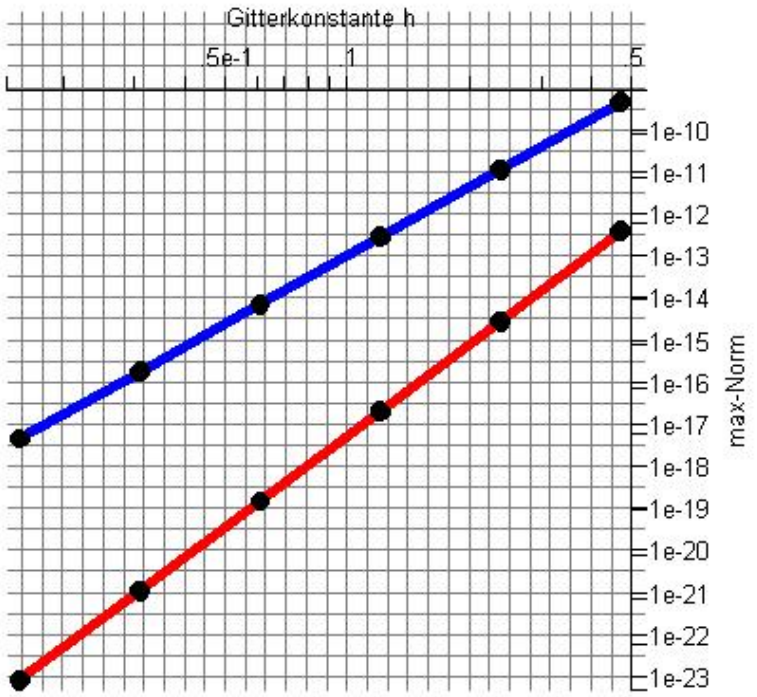

Abbildung 2.6: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot) für das Problem (2.74); $m = 5$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 6, die des Fehlerschätzers ca. 8.

h	esterr	ord	const	collerr	collord
5.0000E-01	4.4540E-08	-	-	1.5929E-05	-
2.5000E-01	6.5303E-10	6.0918	2.68E-06	9.2017E-07	4.1137
1.2500E-01	9.0269E-12	6.1768	2.37E-06	5.6113E-08	4.0355
6.2500E-02	1.2848E-13	6.1346	2.15E-06	3.4849E-09	4.0091
3.1250E-02	1.8929E-15	6.0849	2.02E-06	2.1739E-10	4.0028
1.5625E-02	2.8619E-17	6.0477	1.96E-06	1.3581E-11	4.0006

Tabelle 2.6: Ordnung des Fehlers der Fehlerschätzung für das Problem (2.75), $m = 3$.

Im Differenzenschema für den Fehlerschätzer E_{Δ^m} verwenden wir als Diskretisierung [15], der nun in f vorkommenden, 1. Ableitung den symmetrischen Differenzenquotienten

$$u'(x) = \frac{1}{2h}(u(x+h) - u(x-h)) + O(h^2), \tag{2.76}$$

und für die 2. Ableitung

$$u'' = \Delta_h^2 u(x) + O(h^2). \tag{2.77}$$

Tabelle 2.6 und Abbildung 2.7 zeigen die Fehlerordnung für den Fall eines äquidistanten Gitters in $[0, 1]$ und drei Kollokationspunkten in jedem Teilintervall ($m = 3$).
Tabelle 2.7 und Abbildung 2.8 entsprechend für den Fall von vier Kollokationspunkten ($m = 4$), Tabelle 2.8 und Abbildung 2.9 für den Fall $m = 5$.
Wir beobachten erneut für die Ordnung des Fehlers des Fehlerschätzers 'Kollokationsordnung +2'.

[15] siehe hierzu auch Maple-Codes

h	esterr	ord	const	collerr	collord
5.0000E-01	4.7203E-09	-	-	3.3210E-06	-
2.5000E-01	7.7033E-11	5.9373	2.90E-07	2.0695E-07	4.0042
1.2500E-01	1.1067E-12	6.1212	3.73E-07	1.2895E-08	4.0044
6.2500E-02	1.5927E-14	6.1186	3.72E-07	8.0541E-10	4.0009
3.1250E-02	2.3578E-16	6.0780	3.35E-07	5.0323E-11	4.0005
1.5625E-02	3.5719E-18	6.0446	2.90E-07	3.1451E-12	4.0000

Tabelle 2.7: Ordnung des Fehlers der Fehlerschätzung für das Problem (2.75), $m = 4$.

h	esterr	ord	const	collerr	collord
5.0000E-01	3.5631E-11	-	-	3.5037E-08	-
2.5000E-01	1.2524E-13	8.1523	1.01E-08	5.2337E-10	6.0650
1.2500E-01	4.2704E-16	8.1961	1.08E-08	8.0593E-12	6.0211
6.2500E-02	1.5161E-18	8.1378	9.62E-09	1.2531E-13	6.0071
3.1250E-02	5.5961E-21	8.0817	8.12E-09	1.9559E-15	6.0015
1.5625E-02	2.1196E-23	8.0445	7.04E-09	3.0552E-17	6.0005

Tabelle 2.8: Ordnung des Fehlers der Fehlerschätzung für das Problem (2.75), $m = 5$.

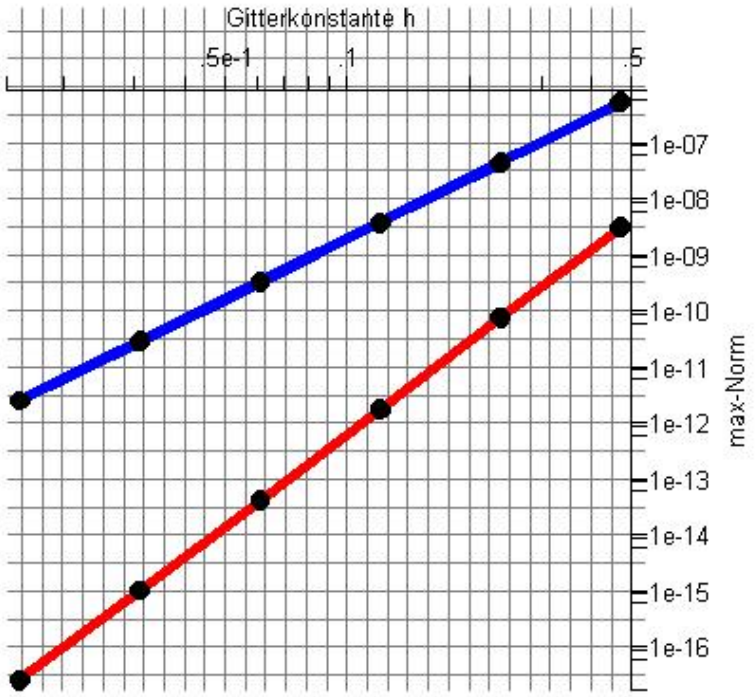

Abbildung 2.7: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot) für das Problem (2.75); $m = 3$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 4, die des Fehlerschätzers ca. 6.

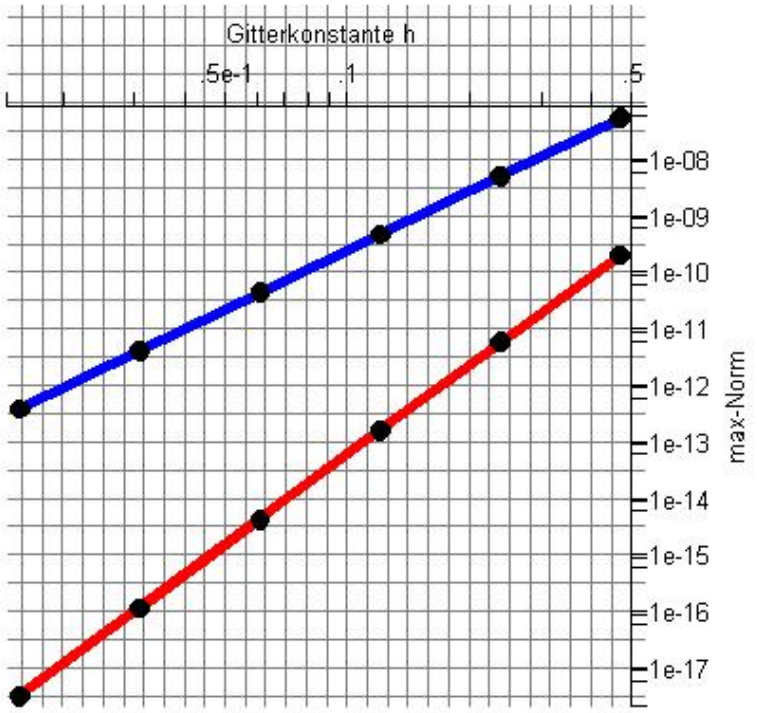

Abbildung 2.8: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot) für das Problem (2.75); $m = 4$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 4, die des Fehlerschätzers ca. 6.

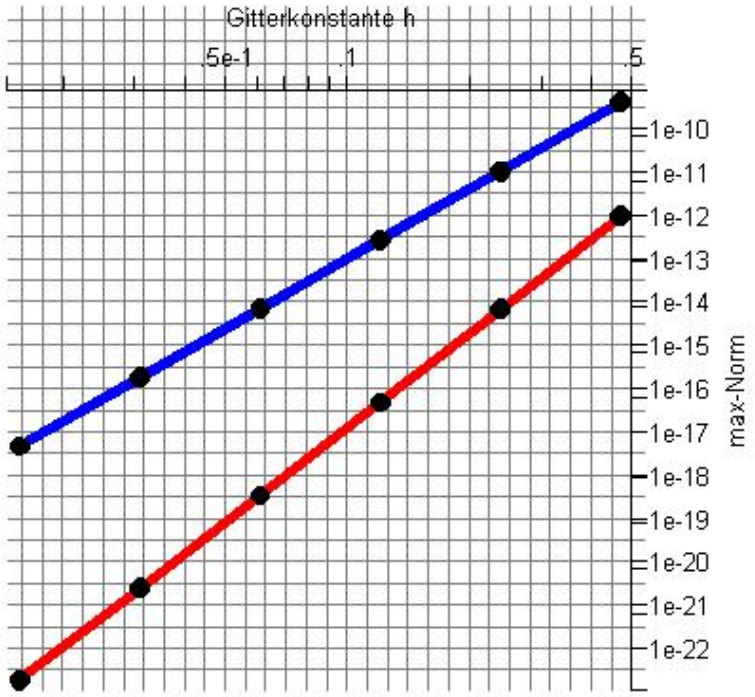

Abbildung 2.9: Doppelt-Logarithmischer-Plot des Kollokationsfehlers (blau) und des Fehlers ϵ (rot) für das Problem (2.75); $m = 5$. Die Steigung des Fehlerverlaufs der Kollokation beträgt ca. 6, die des Fehlerschätzers ca. 8.

2.3 Maple Codes

2.3.1 Code für 2. Ordnung

Maple Code zum Generieren von Konvergenzplots für Probleme zweiter Ordnung (vgl. Abbildung 2.1 bzw. 2.2):

```
> restart;
  #Anzahl der Nachkommastellen bei Gleitpunktarithmetik
  kernelopts(maxdigits);
  Digits := 20;

> #Pakete
  with(LinearAlgebra):
  with(plots):

#DGL und exakte Loesung:#
##########################

> sol:=x->x*(1-x)*exp(-x^2); #exakte Loesung
 lambda:=x->-(1+x); mu:=x->-x; #Koeffizienten
 g:=x->(D@@2)(sol)(x)-mu(x)*(D)(sol)(x)-lambda(x)*sol(x);
 simplify(g(x));
 yl,yr:=sol(0),sol(1); #Randbedingungen
> #Loesbarkeit:
  dsolve([diff(y(x),x$2)-mu(x)*diff(y(x),x)-lambda(x)*y(x)=g(x)
                ,y(0)=yl,y(1)=yr],y(x));
  assign(%);
  ysol:=t->subs(x=t,y(x));
  ysol(t);

> for n from 1 to 6 do #aeusserste Schleife

#PARAMETER:#
############

   N:=2^n:     #Anzahl der Teilintervalle
   m:=2:       #Anzahl der Kollokationspunkte
   d:=m+1:     #Grad der Kollokationspolynome
```

```
   h:=1/N:     #Teilintervalllaenge
   l:=h/(m+1): #Laenge der Kollokationsintervalle

 #GITTER uniform auf [0,1]:#
 ############################

   a:=Array(1..N,1..m+2):
    for i from 1 to N do
      for j from 1 to m+2 do
        a[i,j]:=(i-1)*h+(j-1)*l
      end do:
    end do:

#KOLLOKATION:#
##############

>  #Kollokationspolynom auf dem i-ten Teilintervall:
   p:=(i,x)->add(coe[i,k,n]*x^k,k=0..d):

   #Array fuer die N*(m+2) Bestimmungsgleichungen
   eqsys:=Array(1..N*(m+2)):

   #DGL an Kollokationspunkten
   counter:=0:
   for i from 1 to N do
     for j from 2 to m+1 do
       eq:=diff(p(i,x),x$2)-mu(x)*diff(p(i,x),x)
           -lambda(x)*p(i,x)-g(x):
       counter:=counter+1;
       eqsys[counter]:=evalf(subs(x=a[i,j],eq));
     end do:
   end do:

   #Randbedingungen:
   counter:=counter+1:
   eqsys[counter]:=subs(x=0,p(1,x))=yl:
   counter:=counter+1:
   eqsys[counter]:=subs(x=1,p(N,x))=yr:

   #Stetigkeitsbedingungen:
   for i from 1 to N-1 do
```

```
      counter:=counter+1;
      eqsys[counter]:=p(i,a[i,m+2])-p(i+1,a[i+1,1])=0;
      counter:=counter+1;
      eqsys[counter]:=subs(x=a[i,m+2],diff(p(i,x),x))
                     -subs(x=a[i+1,1],diff(p(i+1,x),x))=0;
   end do:

   #Berechnung der Koeffizienten:
>  unkn:=[seq(seq(coe[i,j,n],i=1..N),j=0..d)]:
   eqsys2:=convert(eqsys,'list'):
   solve(eqsys2,unkn):
   map(assign,%):

   #Globales Kollokationspolynom:
>   p_global:=proc(x)
    local i;
    global a;
    for i from N to 1 by -1 do
      if x>=a[i,1] then return p(i,x) end if;
    end do:
    end proc:

#DEFEKTBERECHNUNG#
##################

>  #Kollokationspolynom in rechter Seite der DGL eingesetzt
   F_p:=(i,x)->mu(x)*subs(x_=x,simplify(diff(p(i,x_),x_)))
               +lambda(x)*p(i,x)+g(x):

   counter:=0:
   #Interpolation von F_p:
   for i from 1 to N do
     intpol_F:=(i,x)->add(icoe_F[i,k,n]*x^k,k=0..m+1):
     solve([seq(F_p(i,a[i,k])=intpol_F(i,a[i,k]),k=1..m+2)],
           [seq(icoe_F[i,k,n],k=0..m+1)]):
     map(assign,%):

   #Defektberechnung:
   for j from 2 to m+1 do
     counter:=counter+1:
     I1:=int((1+t)*intpol_F(i,a[i,j]+t*l),t=-1..0);
```

```
    I2:=int((1-t)*intpol_F(i,a[i,j]+t*l),t=0..1);
    Q[i,j]:=evalf((I1+I2));
    defekt[i,j]:=(1/l^2)*(p(i,a[i,j-1])-2*p(i,a[i,j])
                +p(i,a[i,j+1]))-Q[i,j];
  end do:
  end do:

>  for i from 2 to N do
    counter:=counter+1:
    I1:=int((1+t)*intpol_F(i-1,a[i,1]+t*l),t=-1..0);
    I2:=int((1-t)*intpol_F(i,a[i,1]+t*l),t=0..1);
    Q[i,1]:=evalf((I1+I2));
    defekt[i,1]:=(1/l^2)*(p(i-1,a[i,1]-l)-2*p(i,a[i,1])
                +p(i,a[i,2]))-Q[i,1];
  end do:

#FDS FUER FEHLERSCHAETZER#
##########################

>  eeq:=Array(1..(N*m+2*(N-1))):
  e:=array(1..N,0..m+1):
  e[1,0]:=0: e[N,m+1]:=0:

  #Kollokationspunkte
  counter:=0:
  for i from 1 to N do
    for j from 1 to m do
      counter:=counter+1;
      eeq[counter]:=(e[i,j-1]-2*e[i,j]+e[i,j+1])*1/l^2
                    -mu(a[i,j+1])*(e[i,j+1]
                    -e[i,j-1])*1/(2*l)
                    -lambda(a[i,j+1])*e[i,j]-defekt[i,j+1]=0;
    end do:
  end do:

  #Teilungspunkte:
  for i from 2 to N do
    counter:=counter+1:
    eeq[counter]:=(e[i-1,m]-2*e[i,0]+e[i,1])*1/l^2
                  -mu(a[i,1])*(e[i,1]
                  -e[i-1,m])*1/(2*l)
```

```
                    -lambda(a[i,1])*e[i,0]-defekt[i,1]=0;
   end do:

   #Da doppelt im array e:
   for i from 1 to N-1 do
     counter:=counter+1:
     eeq[counter]:=e[i,m+1]-e[i+1,0];
   end do:

   #Loesung des Gleichungssystems
>  unkn:=[seq(seq(e[i,j],j=0..m+1),i=1..N)]:
   unkn2:=unkn[2..nops(unkn)-1]:
   eeq2:=convert(eeq,'list'):
   solve(eeq2,unkn2):
   map(assign,%):

#FEHLERASYMPTOTIK#
##################

>  c:=Vector(N*(m+2)): #Fehler des Fehlerschaetzers
   kolle:=Vector(N*(m+2)): #Kollokationsfehler

   counter:=0:
   for i from 1 to N do
     for j from 0 to m+1 do
       counter:=counter+1:
       c[counter]:=abs(evalf(p_global((i-1)/N+(j)*l)
                   -ysol((i-1)/N+(j)*l)-e[i,j])):
       kolle[counter]:=abs(evalf[15](p_global((i-1)/N+(j)*l)
                       -ysol((i-1)/N+(j)*l))):
     end do:
   end do:
   l:=(j)->kolle[j]:
   erg[n]:=VectorNorm(Vector(counter,l),infinity):
   m:=(j)->c[j]:
   eerg[n]:=VectorNorm(Vector(counter,m),infinity):

end do: #Ende auesserste Schleife

#LOGLOG-PLOTS#
###############
```

```
> p1:=loglogplot([seq([(1/2^(j)),(erg[j])],j=1..6)],style=line,
      color=blue,thickness=3,gridlines=true,
      labels=["Gitterkonstante h","max-Norm"],
      labeldirections=[horizontal,vertical]):

  p2:=loglogplot([seq([(1/2^(j)),(erg[j])],j=1..6)],style=point,
      color=black,thickness=5,gridlines=true):

  p3:=loglogplot([seq([(1/2^(j)),(eerg[j])],j=1..6)],style=line,
      color=red,thickness=3,gridlines=true):

  p4:=loglogplot([seq([(1/2^(j)),(eerg[j])],j=1..6)],style=point,
      color=black,thickness=5,gridlines=true):

  display(p3,p4,p1,p2);
```

2.3.2 Code für 4. Ordnung

Maple Code zum Generieren von Konvergenzplots für Probleme vierter Ordnung (vgl. Abbildung 2.4 bis 2.9):

```
> restart;
  #Anzahl der Nachkommastellen bei Gleitpunktarithmetik
  kernelopts(maxdigits);
  Digits := 40;

> #Pakete
  with(LinearAlgebra):
  with(plots):

#DGL und exakte Loesung:#
##########################

> sol:=x->x*(1-x)*exp(x); #exakte Loesung
  #Koeffizienten
  mu1:=x->0; mu2:=x->0; mu3:=x->0; lambda:=x->x*(1-x);
  #Randbedingungen:
```

```
  yl,yr,yl2,yr2:=sol(0),sol(1),evalf(subs(x=0,diff(sol(x),x))),
                 evalf(subs(x=1,diff(sol(x),x)));
  g:=x->(D@@4)(sol)(x)-mu1(x)*(D@@3)(sol)(x)
        -mu2(x)*(D@@2)(sol)(x)
        -mu3(x)*(D)(sol)(x)
        -lambda(x)*(sol)(x);
  simplify(g(x));

  #Loesbarkeit:
> ode:= diff(y(x),x$4)-mu1(x)*diff(y(x),x$3)
       -mu2(x)*diff(y(x),x$2)-mu3(x)*diff(y(x),x)
       -lambda(x)*y(x)=g(x):
  bcs:= y(0)=yl,y(1)=yr,D(y)(0)=yl2,D(y)(1)=yr2:
  dsolve({ode,bcs});
  assign(%);

> for n from 1 to 6 do #auesserste Schleife

#PARAMETER:#
#############

   N:=2^n:     #Anzahl der Teilintervalle
   m:=5:       #Anzahl der Kollokationspunkte
   d:=m+3:     #Grad der Kollokationspolynome
   h:=1/N:     #Teilintervalllaenge
   l:=h/(m+1): #Laenge der Kollokationsintervalle

#GITTER uniform auf [0,1]:#
############################

   a:=Array(1..N,1..m+2):
   for i from 1 to N do
     for j from 1 to m+2 do
       a[i,j]:=(i-1)*h+(j-1)*l
     end do:
   end do:

#KOLLOKATION:#
##############
>  #Kollokationspolynom auf dem i-ten Teilintervall
   p:=(i,x)->add(coe[i,k,n]*x^k,k=0..d):
```

```
#Array fuer die N*(m+4) Bestimmungsgleichungen
eqsys:=Array(1..N*(m+4)):

#DGL an Kollokationspunkten
counter:=0:
for i from 1 to N do
  for j from 2 to m+1 do
    eq:=diff(p(i,x),x$4)-mu1(x)*diff(p(i,x),x$3)
        -mu2(x)*diff(p(i,x),x$2)
        -mu3(x)*diff(p(i,x),x)
        -lambda(x)*p(i,x)-g(x):
    counter:=counter+1;
    eqsys[counter]:=evalf(subs(x=a[i,j],eq))=0;
  end do:
end do:

#Randbedingungen:
counter:=counter+1:
eqsys[counter]:=subs(x=0,p(1,x))=yl:
counter:=counter+1:
eqsys[counter]:=subs(x=1,p(N,x))=yr:
counter:=counter+1:
eqsys[counter]:=subs(x=0,diff(p(1,x),x))=yl2:
counter:=counter+1:
eqsys[counter]:=subs(x=1,diff(p(N,x),x))=yr2:

#Stetigkeitsbedingungen:

for i from 1 to N-1 do
  counter:=counter+1;
  eqsys[counter]:=p(i,a[i,m+2])-p(i+1,a[i+1,1])=0;
  counter:=counter+1;
  eqsys[counter]:=subs(x=a[i,m+2],diff(p(i,x),x))
                  -subs(x=a[i+1,1],diff(p(i+1,x),x))=0;
  counter:=counter+1;
  eqsys[counter]:=subs(x=a[i,m+2],diff(p(i,x),x$2))
                  -subs(x=a[i+1,1],diff(p(i+1,x),x$2))=0;
  counter:=counter+1;
  eqsys[counter]:=subs(x=a[i,m+2],diff(p(i,x),x$3))
                  -subs(x=a[i+1,1],diff(p(i+1,x),x$3))=0;
```

```
   end do:

   #Berechnung der Koeffizienten:
>  unkn:=[seq(seq(coe[i,j,n],i=1..N),j=0..d)]:
>  eqsys2:=convert(eqsys,'list'):
>  solve(eqsys2,unkn):
>  map(assign,%):

   #Globales Kollokationspolynom:
>  p_global:=proc(x)
   local i;
   global a;
   for i from N to 1 by -1 do
     if x>=a[i,1] then return p(i,x) end if;
   end do;
   end proc:

#DEFEKTBERECHNUNG#
##################
>  #Kollokationspolynom in rechter Seite der DGL eingesetzt
   F_p:=(i,x)->mu1(x)*subs(x_=x,simplify(diff(p(i,x_),x_$3)))
               +mu2(x)*subs(x_=x,simplify(diff(p(i,x_),x_$2)))
               +mu3(x)*subs(x_=x,simplify(diff(p(i,x_),x_)))
               +lambda(x)*p(i,x)+g(x):

   counter:=0:
   #Interpolation vpn F_p
   for i from 1 to N do
     intpol_F:=(i,x)->add(icoe_F[i,k,n]*x^k,k=0..m+1):
     solve([seq(F_p(i,a[i,k])=intpol_F(i,a[i,k]),k=1..m+2)],
           [seq(icoe_F[i,k,n],k=0..m+1)]):
     map(assign,%):

   #innere Pkte, mind. 2 von TP d. groben Gitters entfernt:
   for j from 3 to m do
     counter:=counter+1:
     I1:=int((1/6*t^3+t^2+2*t+4/3)
         *intpol_F(i,a[i,j]+t*l),t=-2..-1);
     I2:=int((-1/2*t^3-t^2+2/3)
         *intpol_F(i,a[i,j]+t*l),t=-1..0);
     I3:=int((1/2*t^3-t^2+2/3)
```

```
      *intpol_F(i,a[i,j]+t*l),t=0..1);
  I4:=int((-1/6*t^3+t^2-2*t+4/3)
      *intpol_F(i,a[i,j]+t*l),t=1..2);
  Q[i,j]:=evalf((I1+I2+I3+I4));
  defekt[i,j]:=evalf((1/l^4)*(p(i,a[i,j-2])
              -4*p(i,a[i,j-1])+6*p(i,a[i,j])
              -4*p(i,a[i,j+1])+p(i,a[i,j+2]))
              -Q[i,j]);
end do;
end do:

#An den Teilungspunken d. groben Gitters:
for i from 2 to N do
  counter:=counter+1:
  I1:=int((1/6*t^3+t^2+2*t+4/3)
      *intpol_F(i-1,a[i,1]+t*l),t=-2..-1);
  I2:=int((-1/2*t^3-t^2+2/3)
      *intpol_F(i-1,a[i,1]+t*l),t=-1..0);
  I3:=int((1/2*t^3-t^2+2/3)
      *intpol_F(i,a[i,1]+t*l),t=0..1);
  I4:=int((-1/6*t^3+t^2-2*t+4/3)
      *intpol_F(i,a[i,1]+t*l),t=1..2);
  Q[i,1]:=evalf((I1+I2+I3+I4));
  defekt[i,1]:=evalf((1/l^4)*(p(i-1,a[i,1]-2*l)
              -4*p(i-1,a[i,1]-l)+6*p(i,a[i,1])
              -4*p(i,a[i,1]+l)+p(i,a[i,1]+2*l))-Q[i,1]);
#rechts vom TP:
  counter:=counter+1:
  I1:=int((1/6*t^3+t^2+2*t+4/3)
      *intpol_F(i-1,a[i,2]+t*l),t=-2..-1);
  I2:=int((-1/2*t^3-t^2+2/3)
      *intpol_F(i,a[i,2]+t*l),t=-1..0);
  I3:=int((1/2*t^3-t^2+2/3)
      *intpol_F(i,a[i,2]+t*l),t=0..1);
  I4:=int((-1/6*t^3+t^2-2*t+4/3)
      *intpol_F(i,a[i,2]+t*l),t=1..2);
  Q[i,2]:=evalf((I1+I2+I3+I4));
  defekt[i,2]:=evalf((1/l^4)*(p(i-1,a[i,2]-2*l)
               -4*p(i,a[i,2]-l)+6*p(i,a[i,2])
               -4*p(i,a[i,2]+l)+p(i,a[i,2]+2*l))-Q[i,2]);
#links vom TP:
```

```
      counter:=counter+1:
      I1:=int((1/6*t^3+t^2+2*t+4/3)
          *intpol_F(i-1,a[i-1,m+1]+t*l),t=-2..-1);
      I2:=int((-1/2*t^3-t^2+2/3)
          *intpol_F(i-1,a[i-1,m+1]+t*l),t=-1..0);
      I3:=int((1/2*t^3-t^2+2/3)
          *intpol_F(i-1,a[i-1,m+1]+t*l),t=0..1);
      I4:=int((-1/6*t^3+t^2-2*t+4/3)
          *intpol_F(i,a[i-1,m+1]+t*l),t=1..2);
      Q[i-1,m+1]:=evalf((I1+I2+I3+I4));
      defekt[i-1,m+1]:=evalf((1/l^4)*(p(i-1,a[i-1,m+1]-2*l)
                       -4*p(i-1,a[i-1,m+1]-l)
                       +6*p(i-1,a[i-1,m+1])
                       -4*p(i-1,a[i-1,m+1]+l)
                       +p(i,a[i-1,m+1]+2*l))
                       -Q[i-1,m+1]);
   end do:

   #Integralrestglied des EDS fuer u'(0)
>  corr1:=l^3/2*int((1-t)^3*intpol_F(1,l*t),t=0..1)
         -16*l^3/4*int((1-t)^3*intpol_F(1,2*l*t),t=0..1)
         +3^4*l^3/18*int((1-t)^3*intpol_F(1,t*3*l),t=0..1):

   #Integralrestglied des EDS fuer u'(1)
>  corr2:=1/(2*l)*int(((1-l)-t)^3*intpol_F(N,t),t=1-l..1)
         -1/(4*l)*int(((1-2*l)-t)^3*intpol_F(N,t),t=1-2*l..1)
         +1/(18*l)*int(((1-3*l)-t)^3*intpol_F(N,t),t=1-3*l..1):


#FDS FUER FEHLERSCHAETZER#
##########################
>  counter:=0:
   eeq:=Array(1..N*(m+1)+(N-1)-1):
   e:=array(1..N,1..m+2):
   e[1,1]:=0:
   e[N,m+2]:=0:

   #EDS fr linke Ableitungsrb:
   counter:=counter+1:
   eeq[counter]:=1/(6*l)*(-11*e[1,1]+18*e[1,2]
                -9*e[1,3]+2*e[1,4])
```

```
              =1/(6*l)*(-11*p(1,0)+18*p(1,l)
              -9*p(1,2*l)+2*p(1,3*l))
              -yl2-corr1:

#rechte Ableitungsrb:
counter:=counter+1:
eeq[counter]:=1/(6*l)*(11*e[N,m+2]-18*e[N,m+1]
              +9*e[N,m]-2*e[N,m-1])
              =1/(6*l)*(11*p(N,1)-18*p(N,1-l)
              +9*p(N,1-2*l)-2*p(N,1-3*l))
              -yr2-corr2:

#innere Pkte, mind. 2 von TP entfernt:
for i from 1 to N do
  for j from 3 to m do
    counter:=counter+1:
    eeq[counter]:=(1/l^4)*(e[i,j-2]-4*e[i,j-1]+6*e[i,j]
                  -4*e[i,j+1]+e[i,j+2])
                  -mu2(a[i,j])*1/l^2
                  *(e[i,j-1]-2*e[i,j]+e[i,j+1])
                  -mu3(a[i,j])*1/(2*l)
                  *(e[i,j+1]-e[i,j-1])
                  -lambda(a[i,j])*e[i,j]
                  -defekt[i,j]=0:
  end do:
end do:

#An den Teilungspunken d. groben Gitters:
for i from 2 to N do
  counter:=counter+1:
  eeq[counter]:=(1/l^4)*(e[i-1,m]-4*e[i-1,m+1]
                +6*e[i,1]-4*e[i,2]+e[i,3])
                -mu2(a[i,1])*1/l^2*(e[i-1,m+1]
                -2*e[i,1]+e[i,2])
                -mu3(a[i,1])*1/(2*l)*(e[i,2]
                -e[i-1,m+1])
                -lambda(a[i,1])*e[i,1]
                -defekt[i,1]=0:

#rechts vom TP:
counter:=counter+1:
```

```
   eeq[counter]:=(1/l^4)*(e[i-1,m+1]-4*e[i,1]+6*e[i,2]
                 -4*e[i,3]+e[i,4])
                 -mu2(a[i,2])*1/l^2*(e[i,1]-2*e[i,2]+e[i,3])
                 -mu3(a[i,2])*1/(2*l)*(e[i,3]-e[i,1])
                 -lambda(a[i,2])*e[i,2]
                 -defekt[i,2]=0:

   #links vom TP:
   counter:=counter+1:
   eeq[counter]:=(1/l^4)*(e[i-1,m-1]-4*e[i-1,m]+6*e[i-1,m+1]
                 -4*e[i,1]+e[i,2])
                 -mu2(a[i-1,m+1])*1/l^2*(e[i-1,m]
                 -2*e[i-1,m+1]+e[i,1])
                 -mu3(a[i-1,m+1])*1/(2*l)*(e[i,1]-e[i-1,m])
                 -lambda(a[i-1,m+1])*e[i-1,m+1]
                 -defekt[i-1,m+1]=0:
   end do:

   #Da doppelt im array e:
   for i from 1 to N-1 do
     counter:=counter+1:
     eeq[counter]:=e[i,m+2]-e[i+1,1]=0;
   end do:

   #Loesung des Gleichungssystems
>  unkn:=[seq(seq(e[i,j],j=1..m+2),i=1..N)]:
   unkn2:=unkn[2..nops(unkn)-1]:
   eeq2:=convert(eeq,'list'):
   solve(eeq2,unkn2):
   map(assign,%):

#FEHLERASYMPTOTIK#
##################
>  c:=Vector(N*(m+2)): #Fehler des Fehlerschaetzers
   kolle:=Vector(N*(m+2)): #Kollokationsfehler
   counter:=0:
   for i from 1 to N do
     for j from 1 to m+2 do
       counter:=counter+1:
       c[counter]:=abs(evalf(p_global((i-1)/N+(j-1)*l)
                  -sol((i-1)/N+(j-1)*l)-e[i,j])):
```

```
        kolle[counter]:=abs(evalf(p_global((i-1)/N+(j-1)*l)
                        -sol((i-1)/N+(j-1)*l))):
      end do:
   end do:
   l:=(j)->kolle[j]:
   erg[n]:=VectorNorm(Vector(counter,l),infinity):
   m:=(j)->c[j]:
   eerg[n]:=VectorNorm(Vector(counter,m),infinity):

> end do: #Ende aeusserste Schleife

#LOGLOG-PLOTS#
###############
>  p1:=loglogplot({seq([(1/2^(j)),(erg[j])],j=1..6)},
       style=line,color=blue,thickness=3,gridlines=true,
       labels=["Gitterkonstante h","max-Norm"],
       labeldirections=[horizontal,vertical]):

   p2:=loglogplot({seq([(1/2^(j)),(erg[j])],j=1..6)},
       style=point,color=black,thickness=5,gridlines=true):

   p3:=loglogplot({seq([(1/2^(j)),(eerg[j])],j=1..6)},
       style=line,color=red,thickness=3,gridlines=true):

   p4:=loglogplot({seq([(1/2^(j)),(eerg[j])],j=1..6)},
       style=point,color=black,thickness=5,gridlines=true):

   display(p3,p4,p1,p2);
```

Kapitel 3

Anhang

3.1 Fehlerschätzung für FDS-Lösung

Wir wollen hier erneut das folgende lineare Problem 2. Ordnung betrachten:

$$\begin{aligned} &y''(t) = f(t, y(t), y'(t)) = g(t) + \mu(t)y'(t) + \lambda(t)y(t) \qquad t \in [0,1], \\ &y(0) = \beta_1, \quad y(1) = \beta_2, \end{aligned} \tag{3.1}$$

wobei wir wieder die eindeutige Lösbarkeit voraussetzen ($z(t)$ bezeichne die Lösung).
Uns interessiert hier eine a-posteriori Fehlerschätzung der zugehörigen Finite-Differenzen-Lösung η_Δ auf dem äquidistanten Gitter Δ ($h_i \equiv h$)[1]:

$$\Delta_h^2 \eta_\Delta = f(\eta_\Delta), \tag{3.2}$$

mit den Randbedingungen aus (3.1) und die erste Ableitung in f diskretisiert durch den symmetrischen Differenzenquotienten

$$u'(t) \approx \frac{1}{2h}(u(t+h) - u(t-h)). \tag{3.3}$$

Dazu definieren wir den Defekt von η_Δ bezüglich des EDS (2.28) mit 3-Punkt Quadratur:

$$D_\Delta := \Delta_h^2 \eta_\Delta - Q_3 \eta_\Delta, \tag{3.4}$$

[1]Da (3.2) eine $O(h^2)$-Approximation liefert, streben wir (aus Analogiegründen zum letzten Abschnitt) Ordnung 4 für den Fehler der Fehlerschätzung an.

wobei Q_3 eine interpolatorische Quadraturformel von Exaktheitgrad 2 für das Integral

$$\int_{-1}^{1} K_2(\zeta) f(t+\zeta h) d\zeta \tag{3.5}$$

ist.
Interpolieren wir die Punkte $(t-h, a), (t, b)$ und $(t+h, c)$ mit $a, b, c \in \mathbb{R}$ durch das quadratische Polynom $q(t)$ und integrieren (3.5) mit f durch q ersetzt, so ergeben sich die Quadraturgewichte $\frac{1}{12}, \frac{10}{12}$ und $\frac{1}{12}$ für die Quadraturformel Q_3:

$$\int_{-1}^{1} K_2(\zeta) q(t+\zeta h) d\zeta = \frac{1}{12}a + \frac{10}{12}b + \frac{1}{12}c. \tag{3.6}$$

Bemerkung. Verwendet man naheliegenderweise lineare Interpolation (Polygonzug), so sind die zugehörigen Quadraturgewichte $\frac{1}{6}, \frac{4}{6}$ und $\frac{1}{6}$.
Der daraus resultierende Fehlerschätzer, selbst bei der wesentlichen Vereinfachung von (3.1) durch $\mu \equiv 0$, hat jedoch nur Fehlerordnung 2, was ihn unbrauchbar macht.

Zur Vereinfachung betrachten wir zunächst den Fall

$\underline{\mu \equiv 0:}$

Da hier in f kein Ableitungsterm auftritt, brauchen wir uns über eine zusätzlich Glättung der FDS-Lösung durch geeignete Interpolation keine Gedanken machen.
Wir definieren analog zum letzten Abschnitt das Schema für den Fehlerschätzer E_Δ:

$$\Delta_h^2 E_\Delta = f(E_\Delta) + D_\Delta, \tag{3.7}$$

mit homogenen Randbedingungen.

Als Beispiel betrachten wir

$$\begin{aligned} &Lu = u'' + (1+x)u = g(x), \quad x \in [0,1], \\ &u(0) = u(1) = 0, \end{aligned} \tag{3.8}$$

mit der eindeutigen exakten Lösung: $z(x) = x(1-x)e^x$, also $g = Lz$.

h	esterr	ord	const	fdserr	fdsord
5.0000E-01	4.4445E-03	-	-	6.3412E-02	-
2.5000E-01	2.7217E-04	4.0295	7.26E-02	1.5805E-02	2.0043
1.2500E-01	1.7428E-05	3.9651	6.70E-02	3.9472E-03	2.0015
6.2500E-02	1.0899E-06	3.9992	7.14E-02	1.0024E-03	1.9774
3.1250E-02	6.8322E-08	3.9957	7.17E-02	2.5057E-04	2.0002
1.5625E-02	4.2697E-09	4.0002	7.17E-02	6.2641E-05	2.0000

Tabelle 3.1: Ordnung des Fehlers der Fehlerschätzung für das Problem (3.8).

In Tabelle 3.1 beobachten wir Ordnung 4 für den Fehler der Fehlerschätzung ('esterr' bzw. 'ord') bei einer FDS-Ordnung von 2 ('fdsord').

Wir betrachten nun den nächst schwierigen Fall

$\underline{\mu \equiv \text{const} \neq 0 :}$

Das Problem bei Auftreten der ersten Ableitung in der rechten Seite f ist die fehlende Glattheit der FDS-Lösung.
Quadratische Interpolation (über je drei benachbarte Teilungspunkte des Gitters Δ) liefert hier keine vernünftige Fehlerordnung des Fehlerschätzers.

Eine Möglichkeit bietet jedoch Hermite-Interpolation der FDS-Lösung:
Dazu extrapoliert man zuerst die FDS-Lösung am linken Rand durch den Wert η_{li}:

$$\frac{1}{h^2}(\eta_h - 2\eta_0 + \eta_{li}) = \mu(0)\frac{1}{2h}(\eta_h - \eta_{li}) + \lambda(0)\eta_0 + g(0),$$

und analog am rechten Rand durch den Wert η_{re}.
Als Startschritt interpoliert man die Werte η_{li}, η_0 und η_h quadratisch. Anschließend legt man jeweils durch zwei benachbarte Werte ein quadratisches Interpolationspolynom und fordert einen C^1-Übergang an allen Punkten des Gitters Δ.
Dies ergibt eine stückweise quadratische globale C^1-Interpolierende [2].
Im Fall $\mu \equiv const$ kann so wieder Ordnung 4 für den Fehler des Schätzers beobachtet werden (siehe Tabelle 3.2).

[2]Diese ist 'sequentiell' erzeugt, also algorithmisch global. Eine rein lokal ezeugte Variante kann durch eine stückweise kubische C^1-Interpolierende erreicht werden, vgl. [12].

h	esterr	ord	const	fdserr	fdsord
2.5000E-01	7.6610E-03	-	-	4.1294E-02	-
1.2500E-01	3.7627E-04	4.3477	3.19	9.3849E-03	2.1375
6.2500E-02	2.4795E-05	3.9236	1.30	2.2996E-03	2.0290
3.1250E-02	1.5530E-06	3.9970	1.63	5.7549E-04	1.9984
1.5625E-02	9.6802E-08	4.0039	1.63	1.4369E-04	2.0019

Tabelle 3.2: Ordnung des Fehlers der Fehlerschätzung für das Problem (3.9).

Für weiter Details[3], verweisen wir hier auf die Diplomarbeit von G. Kitzler [12].

Als Beispiel dazu betrachten wir

$$\begin{aligned} &Lu = u'' - 4u' + (1+x)u = g(x), \quad x \in [0,1], \\ &u(0) = u(1) = 0, \end{aligned} \tag{3.9}$$

mit der eindeutigen exakten Lösung: $z(x) = x(1-x)e^x$, also $g = Lz$.

3.2 FDS Maple Codes

Maple Code zu dem Fehlerschätzer der FDS-Lösung von Problem (3.8):

```
> restart;
  Digits := 20:
  #Pakete
  with(LinearAlgebra):
  with(plots):

#DGL und Loesung:#
###################

> sol:=x->x*(1-x)*exp(x):
```

[3]Insbesondere Quadratur und nicht-äquidistanter Fall.

```
  lambda:=x->-(1+x):
  g:=x->(D@@2)(sol)(x)-lambda(x)*sol(x):
  yl,yr:=sol(0),sol(1):
  dsolve([diff(y(x),x$2)-lambda(x)*y(x)=g(x),y(0)=yl,y(1)=yr],y(x)):
  assign(%):
  ysol:=t->subs(x=t,y(x)):

> for n from 1 to 6 do  #aeusserste Schleife

#GITTER uniform auf [0,1]:#
############################

  N:=2^(n):
  h:=1/N:
  a:=Array(1..N+1):
  for j from 1 to N+1 do
    a[j]:=(j-1)*h:
  end do:

#FDS-LOESUNG:#
##############
  eqp:=Array(1..N-1):
  p:=array(1..N+1):
  p[1]:=0: p[N+1]:=0:
  counter:=0:

> for j from 2 to N do
    counter:=counter+1:
    eqp[counter]:=1/h^2*(p[j+1]-2*p[j]+p[j-1])
                  -lambda(a[j])*p[j]-g(a[j]):
  end do:

  unkn:=[seq(p[i],i=2..N)]:
  eeqp:=convert(eqp,'list'):
  solve(eeqp,unkn):
  map(assign,%):

#FDS-FEHLER:#
#############

> counter:=0:
```

```
  fehler:=Vector(N+1):
  for j from 1 to N+1 do
    counter:=counter+1:
    fehler[counter]:=evalf(p[j]-ysol(a[j])):
  end do:
  l:=(j)->fehler[j]:
  ergp[n]:=VectorNorm(Vector(counter,l),infinity);

#DEFEKTBERECHNUNG:#
####################

> f1:=j->lambda(a[j])*p[j]+g(a[j]):

  counter:=0:
  for j from 2 to N do
    counter:=counter+1:
    defekt[j]:=evalf(1/h^2*(p[j+1]-2*p[j]+p[j-1])
               -(1/12*f1(j-1)+10/12*f1(j)+1/12*f1(j+1))):
  end do:

#SCHEMA FUER SCHAETZER:#
########################

#> h:=1/N:

> e:=array(1..N+1):
  e[1]:=0: e[N+1]:=0:
  eeq:=Array(1..N-1):
  counter:=0:
  for j from 2 to N do
    counter:=counter+1:
    eeq[counter]:=1/h^2*(e[j+1]-2*e[j]+e[j-1])
                  -lambda(a[j])*e[j]-defekt[j]:
  end do:

> unkn2:=[seq(e[i],i=2..N)]:
  eeq2:=convert(eeq,'list'):
  solve(eeq2,unkn2):
> map(assign,%):

#FEHLER DER FEHLERSCHAETZUNG:#
```

```
##############################

> counter:=0:
  ffehler:=Vector(N+1):
  for j from 1 to N+1 do
    counter:=counter+1:
    ffehler[counter]:=evalf(abs((p[j]-ysol(a[j]))-e[j])):
  end do:

  l3:=(j)->ffehler[j]:
  erg[n]:=VectorNorm(Vector(counter,l3),infinity):

end do: #Ende der aeussersten Schleife


#LOGLOG-PLOTS:#
###############

> p1:=loglogplot({seq([(1/2^(j)),(erg[j])],j=2..6)},
      style=line,color=red,thickness=3,gridlines=true,
      labels=["Gitterkonstante h","max-Norm"],
      labeldirections=[horizontal,vertical]):

> p2:=loglogplot({seq([(1/2^(j)),(erg[j])],j=2..6)},
      style=point,color=black,thickness=5,gridlines=true):

> p3:=loglogplot({seq([(1/2^(j)),(ergp[j])],j=2..6)},
      style=line,color=blue,thickness=3,gridlines=true):

> p4:=loglogplot({seq([(1/2^(j)),(ergp[j])],j=2..6)},
      style=point,color=black,thickness=5,gridlines=true):

> display(p1,p2,p3,p4);
```

Maple Code zu dem Fehlerschätzer der FDS-Lösung von Problem (3.9):

```
> restart;
  Digits := 20:
  with(LinearAlgebra):
```

```
  with(plots):

#DGL und Loesung:#
###################

>  sol:=x->x*(1-x)*exp(x):
   lambda:=x->-(1+x):
   mu:=x->4:
   g:=x->(D@@2)(sol)(x)-mu(x)*(D)(sol)(x)-lambda(x)*sol(x):
   simplify(g(x)):
   yl,yr:=sol(0),sol(1):
   dsolve([diff(y(x),x$2)-mu(x)*diff(y(x),x)
   -lambda(x)*y(x)=g(x),y(0)=yl,y(1)=yr],y(x)):
   assign(%):
   ysol:=t->subs(x=t,y(x)):

> for n from 2 to 6 do #aeusserste Schleife

  N:=2^(n):
  h:=1/N:

#GITTER uniform auf [0,1]:#
############################

>  a:=array(1..N+1):
   for j from 1 to N+1 do
     a[j]:=(j-1)*h:
   end do:

#FDS-LOESUNG:#
###############

>  eqp:=array(1..N-1):
   p:=array(1..N+1):
   p[1]:=0: p[N+1]:=0:
   counter:=0:
   for j from 2 to N do
     counter:=counter+1:
     eqp[counter]:=1/h^2*(p[j+1]-2*p[j]+p[j-1])
     -mu(a[j])*1/(2*h)*(p[j+1]-p[j-1])-lambda(a[j])*p[j]-g(a[j]):
   end do:
```

```
   unkn:=[seq(p[i],i=2..N)]:
   eeqp:=convert(eqp,'list'):
   solve(eeqp,unkn):
   map(assign,%):

#RAND-EXTRAPOLATION:#
######################

> pLINKSAUSSEN:='pLINKSAUSSEN':
  eq:=1/h^2*(p[2]-2*p[1]+pLINKSAUSSEN)
  -mu(a[1])*1/(2*h)*(p[2]-pLINKSAUSSEN)
  -lambda(a[1])*p[1]-g(a[1]):
  pLINKSAUSSEN:=solve(eq,pLINKSAUSSEN):

  pRECHTSAUSSEN:='pRECHTSAUSSEN':
  eq:=1/h^2*(pRECHTSAUSSEN-2*p[N+1]+p[N])
  -mu(a[N+1])*1/(2*h)*(pRECHTSAUSSEN-p[N])
  -lambda(a[N+1])*p[N+1]-g(a[N+1]):
  pRECHTSAUSSEN:=solve(eq,pRECHTSAUSSEN):

#FDS-FEHLER:#
#############

  counter:=0:
  fehler:=vector(N+1):
  for j from 1 to N+1 do
    counter:=counter+1:
    fehler[counter]:=evalf(p[j]-ysol(a[j])):
  end do:
  l:=(j)->fehler[j]:
  ergp[n]:=VectorNorm(Vector(counter,l),infinity);

#INTERPOLATION (HERMITE) VON p:#
################################

> for j from 0 to N+1 do
    intpol_p:=(x,j)->add(icoe_p[k,j,n]*x^k,k=0..2):
  end do:
  solve([intpol_p(-h,0)=pLINKSAUSSEN,
  seq(intpol_p(a[j],0)=p[j],j=1..2)],
```

```
  [seq(icoe_p[k,0,n],k=0..2)]):
  map(assign,%):
  solve([seq(subs(x=a[j],diff(intpol_p(x,j-1),x))=
  subs(x=a[j],diff(intpol_p(x,j),x)),j=1..N+1),
  seq(seq(intpol_p(a[j+k],j)=p[j+k],j=1..N),k=0..1),
  intpol_p(a[N+1],N+1)=p[N+1],
  intpol_p(a[N+1]+h,N+1)=pRECHTSAUSSEN],
  [seq(seq(icoe_p[k,j,n],k=0..2),j=1..N+1)]):
  map(assign,%):

#DEFEKTBERECHNUNG:#
####################

> f1:=j->lambda(a[j])*p[j]+g(a[j]):

  counter:=0:
  for j from 2 to N do
    counter:=counter+1:
    defekt[j]:=evalf(1/h^2*(p[j+1]-2*p[j]+p[j-1])-
    (1/12*f1(j-1)+10/12*f1(j)+1/12*f1(j+1)+
    int(mu((a[j-1]+a[j])/2)*subs(x_=a[j]+
       t*h,simplify(diff(intpol_p(x_,j-1),x_)))
       *(1-abs(t)),t=-1..0)+
    int(mu((a[j]+a[j+1])/2)*subs(x_=a[j]+
       t*h,simplify(diff(intpol_p(x_,j),x_)))
       *(1-abs(t)),t=0..1))):
  end do:

#SCHEMA FUER SCHAETZER:#
########################

>  e:=array(1..N+1):
   e[1]:=0: e[N+1]:=0:
   eeq:=array(1..N-1):
   counter:=0:
   for j from 2 to N do
     counter:=counter+1:
     eeq[counter]:=1/h^2*(e[j+1]-2*e[j]+e[j-1])
     -mu(a[j])*1/(2*h)*(e[j+1]-e[j-1])
     -lambda(a[j])*e[j]-defekt[j]:
   end do:
```

```
   unkn2:=[seq(e[i],i=2..N)]:
   eeq2:=convert(eeq,'list'):
   solve(eeq2,unkn2):
   map(assign,%):

#FEHLER DER FEHLERSCHAETZUNG:#
###############################

>  counter:=0:
   ffehler:=vector(N+1):
   for j from 1 to N+1 do
     counter:=counter+1:
     ffehler[counter]:=evalf(abs((p[j]-ysol(a[j]))-e[j])):
   end do:
   l3:=(j)->ffehler[j]:
   erg[n]:=VectorNorm(Vector(counter,l3),infinity):

end do: #Ende der aeussersten Schleife

#LOGLOG-PLOTS:#
###############

> p1:=loglogplot({seq([(1/2^(j)),(erg[j])],j=2..6)},
      style=line,color=red,thickness=3,gridlines=true,
      labels=["Gitterkonstante h","max-Norm"],
      labeldirections=[horizontal,vertical]):

> p2:=loglogplot({seq([(1/2^(j)),(erg[j])],j=2..6)},
      style=point,color=black,thickness=5,gridlines=true):

> p3:=loglogplot({seq([(1/2^(j)),(ergp[j])],j=2..6)},
      style=line,color=blue,thickness=3,gridlines=true):

> p4:=loglogplot({seq([(1/2^(j)),(ergp[j])],j=2..6)},
      style=point,color=black,thickness=5,gridlines=true):

> display(p1,p2,p3,p4);
```

Literaturverzeichnis

[1] W. Auzinger, O. Koch and E. Weinmüller, *Efficient collocation schemes for singular boundary value problems* (Numer.Algorithms 31, 5-25, 2002).

[2] U. Ascher, R.M.M. Mattheij and R.D. Russell, *Numerical solutions of boundary value problems for ordinary differential equations* (Prentice-Hall, Englewood Cliffs, NJ 1988).

[3] W. Auzinger, O. Koch, S. Lammer, E. Weinmüller: *Variationen der Defektkorrektur zur effizienten numerischen Lösung gewöhnlicher Differentialgleichungen*, (Technical Report ANUM Preprint No.8/02).

[4] W. Auzinger, *Einführung in die Numerik der Differentialgleichungen Teil 1* (Vorlesungsskript TU Wien 2007).

[5] W. Auzinger, D. Praetorius, *Numerische Mathematik Teil 2* (Vorlesungsskript TU Wien 2006).

[6] H. Engels, *Numerical Quadrature and Cubature* (Academic Press Inc., 1980).

[7] W. Auzinger, *An overview on defect-based a posteriori error estimation for ODEs and DAEs* (Vortrag Chernivtsi, 2009).

[8] P.E. Zadunaisky, *On the estimation of errors propagated in the numerical integration of ODEs* (Numer.Math. 27, 21- 39, 1976).

[9] H.J. Stetter, *The defect correction principle and discretization methods* (Numer.Math. 29, 425- 443, 1978).

[10] C. Großman, H.-G. Roos, *Numerik partieller Differentialgleichungen* (Teubner-Verlag, 1992).

[11] A. S. Bagherzadeh, *Dissertation* (TU Wien, in Vorbereitung).

[12] G. Kitzler, *Diplomarbeit* (TU Wien, in Vorbereitung).