

Freie Universität Berlin
Fachbereich Mathematik und Informatik
Bachelorarbeit
im Studiengang
„Mathematik Bachelor of Science“
Thema:
„Explizite s -stufige Peer-Methoden zur
numerischen Lösung von
Anfangswertproblemen“

betreut von
Prof. Dr. Volker John
und
Dr. Alfonso Caiazzo

vorgelegt von
Louis Nicolas Jardine Anderson

Matrikelnummer: 5173212

E-Mail: louisnicoa98@zedat.fu-berlin.de

Berlin, den 22. Juni 2023

Inhaltsverzeichnis

1	Einführung	3
2	Definition - s-stufige explizite Peer-Methode	4
3	Kompakte Darstellung	5
4	Residuum und Konsistenzordnung	6
5	Konvergenz	11
6	Superkonvergenz	16
7	Konstruktionsprinzip	32
8	Numerische Versuche und Fazit	33
9	Ausblick - linear-implizite Peer-Methoden	36
10	Anhang	44

1 Einführung

Im Folgenden möchte ich die sogenannte explizite s -stufige Peer-Methode einführen, welche der Approximation von Anfangswertproblemen für Differentialgleichungen erster Ordnung dient. Dabei werde ich mich grundsätzlich, sofern nicht anderweitig erwähnt, auf Podhaisky et al. [1] berufen. Insofern gilt diese Referenz für die gesamte Arbeit und wird nicht nochmal explizit erwähnt.

Gegeben sei das Anfangswertproblem für ein System einer gewöhnlichen Differentialgleichung erster Ordnung

$$\begin{aligned} y' &= f(t, y), \quad t \in [t_0, t_e], \\ y(t_0) &= y_0. \end{aligned} \tag{1}$$

Um jenes Anfangswertproblem zu lösen, werden wir das explizite Peer-Verfahren einführen. Zuerst betrachten wir die Definition des expliziten Verfahrens. Anschließend studieren wir den Fehler, die Konsistenz, die Konvergenz sowie die Superkonvergenz der expliziten s -stufigen Peer-Methode. Mit jenen Erkenntnissen können wir nämlich beurteilen, inwieweit die Verfahren überhaupt dazu geeignet sind, uns Lösungen des beschriebenen Anfangswertproblems anzunähern.

In der Praxis wenden wir das explizite Peer-Verfahren an, indem wir bestimmte Ausdrücke, die wir für die Approximation benötigen (wie beispielsweise der Wert der Approximation des vorherigen Zeitschrittes), in die unten stehende Summelformel, siehe (2), einsetzen. Dann erhalten wir eine Approximation für unseren erwünschten Zeitschritt. Eine Besonderheit des s -stufigen expliziten Peer-Verfahrens besteht, dass wir pro Zeitschritt auch s Abschätzungen berechnen können. Aber dazu später mehr.

Besagte Summenformel besteht allerdings auch aus Ausdrücken, welche wir nicht aus unserem Anfangswertproblem schließen können. Im Wesentlichen handelt es sich dabei um Einträge aus bestimmten Koeffizientenmatrizen, welche uns jedoch unbekannt sind. Die Frage stellt sich also, wie wir diese Koeffizienten bestimmen, damit die explizite s -stufige Peer-Methode auch ein bestimmtes Maß an Genauigkeit erfüllt. Die Kernaufgabe dieser Arbeit ist es, ein Konstruktionsprinzip zu ermitteln. Dieses wird uns eine Anleitung liefern, wie wir die Koeffizienten bestimmen, unter der Voraussetzung dass die explizite Peer-Methode superkonvergent, also konvergent von Ordnung $p + 1$ ist. Vorweggenommen sei schon mal, dass die Koeffizienten unter anderem von den im Verfahren verwendeten Schrittweiten abhängt. Sobald die Koeffizienten entsprechend bestimmt werden, können wir die Approximation $U_{m,i} \approx f(t_{m,i})$ bestimmen. Mehr dazu siehe später.

Außerdem widmen wir uns der Implementierung des expliziten Verfahrens. Dabei finden wir heraus, dass die explizite Methode für steife Systeme deutlich schlechter geeignet ist als für nichtsteife Systeme. Bei Letzteren erhalten wir, unter einer Einschränkung, zufriedenstellende Ergebnisse.

Nachdem wir uns mit den numerischen Beispielen und den daraus resultierenden Erkenntnissen final auseinandersetzen, geben wir zum Abschluss einen kurzen Ausblick auf die linear-impliziten Peer-Methode. Diese ist, im Gegensatz zum expliziten Verfahren, auch für steife Systeme gut geeignet.

2 Definition - s-stufige explizite Peer-Methode

Definiere im Folgenden eine s -stufige und explizite Peer-Methode. Dabei wird der Zeitschritt von t_m nach $t_{m+1} = t_m + h_m$ offensichtlich durch die Schrittweite h_m vollzogen. Pro Zeitschritt berechnen wir mit der s -stufigen Peer-Methode s Approximationen $U_{m,i} \approx f(t_{m,i})$, da $i = 1, \dots, s$. Die fraglichen Stellen $t_{m,i} = t_m + c_i h_m$ setzen sich hierbei zusammen aus paarweise verschiedenen Knoten c_i . Die Definition lautet

$$U_{m,i} = \sum_{j=1}^s b_{ij} U_{m-1,j} + h_m \sum_{j=1}^s a_{ij} f(t_{m-1,j}, U_{m-1,j}) + h_m \sum_{j=1}^{i-1} r_{ij} f(t_{m,j}, U_{m,j}), \quad (2)$$

$$i = 1, \dots, s.$$

Der Begriff Peer-Methode rührt daher, dass wir diese Methode mit einer verschiedenen Anzahl von s Stufen anwenden können, zum Beispiel mit einer Stufe oder fünf Stufen. Alle Näherungen, welche wir mit dem Verfahren berechnen, sind gleichberechtigt, egal wie viele Stufen unser Verfahren hat.

Wie die Koeffizienten a_{ij} , b_{ij} und r_{ij} ermittelt werden, zeigen wir an anderer Stelle. Grundsätzlich hängen sie vom Schrittweitenverhältnis

$$\sigma_m = h_m / h_{m-1} \quad (3)$$

ab.

Bemerkung 1:

Bei den Peer-Methoden bezeichnen wir das Schrittweitenverhältnis in Anlehnung an die Literatur als σ anstatt ω - im Gegensatz zu linearen Mehrschrittverfahren.

3 Kompakte Darstellung

Im Folgenden möchten wir die Peer-Methoden-Berechnung kompakter darstellen.

Dazu betrachten wir die Notationen

$$\begin{aligned} U_m &= (U_{m,i})_{i=1}^s \in \mathbb{R}^{sn}, \\ F(t_m, U_m) &= f(t_{m,i}, U_{m,i}), \\ A_m &= (a_{ij})_{i,j=1}^s, \\ B_m &= (b_{ij})_{i,j=1}^s, \\ R_m &= (r_{ij})_{i,j=1}^s. \end{aligned}$$

Ebenfalls benötigen wir das Kronecker-Produkt, welches wir erstmal definieren. Betrachte nämlich eine $(m \times n)$ -Matrix A sowie eine $(p \times r)$ -Matrix B . Dann ist das Kronecker-Produkt $A \otimes B = C$ wie folgt definiert:

Es lautet $C = (a_{ij} \cdot B) = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \cdots & \cdots & \cdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$. Explizit ist das Kronecker-Produkt

$$C = (a_{ij} \cdot B) = \begin{pmatrix} a_{11}b_{11} & \cdots & a_{11}b_{1r} & \cdots & \cdots & a_{1n}b_{11} & \cdots & a_{1n}b_{1r} \\ \cdots & \ddots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots \\ a_{11}b_{p1} & \cdots & a_{11}b_{pr} & \cdots & \cdots & a_{1n}b_{p1} & \cdots & a_{1n}b_{pr} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ a_{m1}b_{11} & \cdots & a_{m1}b_{1r} & \cdots & \cdots & a_{mn}b_{11} & \cdots & a_{mn}b_{1r} \\ \cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\ a_{m1}b_{p1} & \cdots & a_{m1}b_{pr} & \cdots & \cdots & a_{mn}b_{p1} & \cdots & a_{mn}b_{pr} \end{pmatrix}.$$

Mit dem Kronecker-Produkt lautet das Peer-Verfahren

$$U_m = (B_m \otimes I)U_{m-1} + h_m(A_m \otimes I)F(t_{m-1}, U_{m-1}) + h_m(R_m \otimes I)F(t_m, U_m). \quad (4)$$

Für alle $j \geq i$ sind die Summanden von $h_m \sum_{j=1}^{i-1} r_{ij} f(t_{m,j}, Y_{m,j})$ gleich 0, da die Summe nur bis $i-1$ geht. Daher muss R_m eine strenge untere Dreiecksmatrix sein.

Sei $R_m = 0$. Dann sind die jeweiligen Stufenwerte von den vorausgegangenen Stufenwerten unabhängig, da die $f(t_{m,j}, U_{m,j})$ beziehungsweise die $F(t_m, U_m)$ entfallen.

Es ermöglicht uns daher, die s Stufenwerte parallel zu ermitteln.

Wir haben nun, eben mit dem Ziel, eine Lösung einer Differentialgleichung erster Ordnung zumindest näherungsweise anzugeben, die Peer-Methode explizit definiert.

Dies führt uns nun zur Frage, wie genau jene Approximation ist. Daher studieren wir im nächsten Abschnitt den Fehler beziehungsweise das Residuum, sowie die Konsistenz der Peer-Methode.

4 Residuum und Konsistenzordnung

Nun möchten wir eine Konsistenzordnung für das Peer-Verfahren ermitteln. Dazu definieren wir das Residuum der i -ten Stufe - hierbei setzen wir die tatsächliche Lösung in unser explizites Verfahren ein und erhalten

$$\Delta_{m,i} := y(t_{m,i}) - \sum_{j=1}^s b_{ij}y(t_{m-1,j}) - h_m \sum_{j=1}^s a_{ij}y'(t_{m-1,j}) - h_m \sum_{j=1}^{i-1} r_{ij}y'(t_{m,j}), \quad (5)$$

$$i = 1, \dots, s.$$

Definition 1:

Die explizite s -stufige Peer-Methode verfügt über die Konsistenzordnung p genau dann, wenn

$$\Delta_{m,i} := O(h_m^{p+1})$$

gilt für

$$i = 1, \dots, s.$$

Betrachte nun die Koeffizienten a_{ij} , b_{ij} und r_{ij} . Ursprünglich hatten wir bei der Bildung der expliziten Peer-Methode noch offengehalten, wie wir die Koeffizienten bestimmen.

Ein Kriterium dafür, wie wir sie wählen, besteht darin, dass das Peer-Verfahren mit ebenjenen Koeffizienten die Konsistenzordnung p haben soll.

Mithilfe der Taylorreihe betrachten wir, wie sich y , also die gesuchte Lösung unseres ursprünglichen Anfangswertproblems aus (1), am Entwicklungspunkt t_m verhält.

Zur Erinnerung: Für ein glattes $f : I \rightarrow \mathbb{R}$ so dass $I \in \mathbb{R}$ und $a \in I$ ist

$$T_{f(x;a)} = \sum_{l=0}^{\infty} \frac{f^{(l)}(a) \cdot (x-a)^l}{l!}.$$

Zur folgenden Taylorapproximation haben wir $\mathcal{O}(h_m^{p+1})$ addiert. Dies machen wir, da wir eben die Koeffizienten so bestimmen wollen, dass für das Residuum $\Delta_{m,i} := \mathcal{O}(h_m^{p+1})$ gelten soll - und die Konsistenzordnung somit p beträgt. Wir erhalten

$$y(t_m + c_i h_m) = \sum_{l=0}^p \frac{c_i^l \cdot h_m^l \cdot y^{(l)}(t_m)}{l!} + \mathcal{O}(h_m^{p+1}). \quad (6)$$

Außerdem gilt wegen $t_{m-1} + h_{m-1} = t_m$ und $h_{m-1} = \frac{\sigma_m}{h_m}$

$$\begin{aligned} y(t_{m-1} + c_i h_{m-1}) &= y(t_m - h_{m-1} + c_i h_{m-1}) \\ &= y(t_m + (c_i - 1)h_{m-1}) \\ &= y(t_m + \frac{(c_i - 1)h_m}{\sigma_m}). \end{aligned} \quad (7)$$

Führe eine Taylorentwicklung für den Ausdruck von (7) durch und erhalte

$$y(t_{m-1} + c_i h_{m-1}) = \sum_{l=0}^p \frac{(c_i - 1)^l \cdot h_m^l \cdot y^{(l)}(t_m)}{l! \cdot \sigma_m^l} + \mathcal{O}(h_m^{p+1}). \quad (8)$$

Nun haben wir also gezeigt, wie sich $y(t_{m-1} + c_i h_{m-1}) = y(t_{m-1,i})$ und $y(t_m + c_i h_m) = y(t_{m,i})$ mittels Taylorentwicklung darstellen lassen. Das Residuum ist, siehe (5), der Ausdruck

$$\Delta_{m,i} = y(t_m + c_i h_m) - \sum_{j=1}^s b_{ij} y(t_{m-1,j}) - h_m \sum_{j=1}^s a_{ij} y'(t_{m-1,j}) - h_m \sum_{j=1}^{i-1} r_{ij} y'(t_{m,j}) + \mathcal{O}(h_m^{p+1}).$$

Im Folgenden werden wir für die Ausdrücke mit y die Summen aus (6) beziehungsweise (8) einsetzen. Ziehe diese Summen, also die Summen um $\sum_{l=0}^p$, jeweils auseinander.

Zuerst notieren wir die Summanden in denen wir $l = 0$ eingesetzt haben - in den darauffolgenden Summanden bilden wir dafür Summen mit Startwert 1 anstatt 0.

Wir erhalten für $l = 0$ nur die ersten beiden folgenden Summanden, da sich nur dort $l = 0$ einsetzen lässt (nicht aber in den Termen mit y').
 Setze außerdem (8) ein, ziehe den Vorfaktor h_m jeweils in die beiden zugehörigen Summen herein und erhalte

$$\begin{aligned} \Delta_{m,i} = & \frac{c_i^0 \cdot h_m^0 \cdot y^{(0)}(t_m)}{0!} - \sum_{j=1}^s b_{ij} \frac{(c_i - 1)^0 \cdot h_m^0 \cdot y^{(0)}(t_m)}{0! \cdot \sigma_m^0} + \sum_{l=1}^p \frac{c_i^l \cdot h_m^l \cdot y^{(l)}(t_m)}{l!} \\ & - \sum_{j=1}^s b_{ij} \sum_{l=1}^p \frac{(c_i - 1)^l \cdot h_m^l \cdot y^{(l)}(t_m)}{l! \cdot \sigma_m^l} - \sum_{j=1}^s a_{ij} \frac{\delta}{\delta t} \sum_{l=1}^p \frac{(c_i - 1)^l \cdot h_m^{l+1} \cdot y^{(l)}(t_m)}{l! \cdot \sigma_m^l} \\ & + \sum_{j=1}^{i-1} r_{ij} \frac{\delta}{\delta t} \sum_{l=1}^p \frac{c_i^l \cdot h_m^{l+1} \cdot y^{(l)}(t_m)}{l!} + \mathcal{O}(h_m^{p+1}). \end{aligned}$$

Kürze und leite ab. Der Ausdruck h_m^l lässt sich deshalb ausklammern, da sich bei den h_m^{l+1} der Exponent durch das Ableiten bereits im Vorhinein um 1 reduziert, und die h_m^l selber Faktor zweier anderer Summanden sind. Dann ist

$$\begin{aligned} \Delta_{m,i} = & (1 - \sum_{j=1}^s b_{ij}) y(t_m) + \sum_{l=1}^p \left\{ c_i^l - \sum_{j=1}^s b_{ij} \frac{(c_j - 1)^l}{\sigma_m^l} - l \cdot \sum_{j=1}^s a_{ij} \frac{(c_j - 1)^{l-1}}{\sigma_m^{l-1}} \right. \\ & \left. - l \cdot \sum_{j=1}^{i-1} r_{ij} c_j^{l-1} \right\} \frac{h_m^l \cdot y^{(l)}(t_m)}{l!} + \mathcal{O}(h_m^{p+1}). \end{aligned} \quad (9)$$

Wie der Fehler $\Delta_{m,i}$ lautet, wurde soeben präzisiert.
 Definiere nun folgende Summe aus unserer Rechnung, siehe oben:

$$AB_i(l) := c_i^l - \sum_{j=1}^s b_{ij} \frac{(c_j - 1)^l}{\sigma_m^l} - l \cdot \sum_{j=1}^s a_{ij} \frac{(c_j - 1)^{l-1}}{\sigma_m^{l-1}} - l \cdot \sum_{j=1}^{i-1} r_{ij} c_j^{l-1}. \quad (10)$$

Satz 1:

Sei

$$B_m \mathbb{1} = \mathbb{1}$$

für

$$\mathbb{1} = (1, 1, \dots, 1)^T.$$

Gelte zudem

$$(AB_i(l))_{i=1}^s = AB(l) = 0 \quad \forall l = 0, 1, \dots, p. \quad (11)$$

Genau dann, wenn beide Bedingungen gelten, verfügt das explizite Peer-Verfahren über Konsistenzordnung p .

Beweis:

Da der Faktor um $AB_i(l) = 0$ ist, gilt laut (9) $\Delta_{m,i} = (1 - \sum_{j=1}^s b_{ij})y(t_m) + O(h_m^{p+1})$ und damit $\sum_{j=1}^s b_{ij}y(t_m) = 1$. Dies merken wir uns erstmal vor.

Formuliere das Kriterium $(AB_i(l))_{i=1}^s = AB(l) = 0 \forall l = 0, 1, \dots, p$ in einer Matrixgleichung - dazu verwenden wir die Bezeichnungen

$$\begin{aligned}\hat{D} &= \text{diag}(0, 1, \dots, s), \\ S_m &= \text{diag}(1, \sigma_m, \dots, \sigma_m^{s-1}), \\ \hat{S}_m &= \text{diag}(1, \sigma_m, \dots, \sigma_m^s)\end{aligned}$$

mit dem Ziel, (in dem Fall) die Konsistenzordnung $p = s$ zu erfüllen - siehe die Spaltenanzahl $s + 1$ in der Matrixgleichung

$$\begin{aligned} & \begin{pmatrix} 1 & c_1 & \dots & c_1^s \\ \dots & \dots & \dots & \dots \\ 1 & c_n & \dots & c_n^s \end{pmatrix} + B_m \begin{pmatrix} 1 & c_1 - 1 & \dots & (c_1 - 1)^s \\ \dots & \dots & \dots & \dots \\ 1 & c_n - 1 & \dots & (c_n - 1)^s \end{pmatrix} \hat{S}_m^{-1} \\ -A_m \begin{pmatrix} 0 & 1 & \dots & (c_1 - 1)^{s-1} \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & (c_n - 1)^{s-1} \end{pmatrix} \hat{D} \begin{pmatrix} 0 & 0 \\ 0 & S_m^{-1} \end{pmatrix} - R_m \begin{pmatrix} 0 & 1 & \dots & c_1^{s-1} \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & c_n^{s-1} \end{pmatrix} \hat{D} = 0. \end{aligned} \tag{12}$$

Betrachte nun die erste Spalte unserer Koeffizientenmatrizen A_m , B_m , und R_m . Welche Kriterien gelten für jene Spalte, damit die Matrixgleichung $= 0$ ist und somit die Konsistenzordnung s erfüllt wird?

Dazu betrachten wir Satz 1 und setzen in (10) $l = 0$ ein, da $l = 0, 1, \dots, p$ unseren $1, \dots, s + 1$ Spalten aus den Matrizen unserer Matrixgleichung entspricht:

$$AB_i(0) = 1 - \sum_{j=1}^s b_{ij} = 0.$$

Wir erhalten nur für die erste Spalte von B_m eine Bedingung - in den Summanden, die jeweils die Koeffizienten a_{ij} und r_{ij} beinhalten, kann $l = 0$ aufgrund des Exponenten $l - 1$ nicht eingesetzt werden.

Daraus folgt, für $\mathbb{1} = (1, 1, \dots, 1)^T$,

$$B_m \mathbb{1} = \mathbb{1}. \quad (13)$$

□

Damit ist eine Bedingung für die erste Spalte unserer Koeffizientenmatrizen geliefert, welche sich wie gesagt nur an B_m richtet. Diese Bedingung dient wie gesagt dazu, dass Definition 1 zutreffen soll, also dass das explizite Verfahren über Konsistenzordnung p verfügt. □

Betrachte nun die Kriterien für die Spalten 2 bis $s + 1$, und springe wieder zu (12). Ergänze zudem die Bezeichnungen

$$\begin{aligned} D &= \text{diag}(1, \dots, s), \\ C &= \text{diag}(1, \dots, c_i), \\ V_0 &= (c_i^{j-1})_{i,j=1}^s, \\ V_1 &= ((c_i - 1)^{j-1})_{i,j=1}^s. \end{aligned}$$

Wir stellen folgende Bedingung an B_m und R_m auf, und zwar

$$A_m = (CV_0D^{-1} - R_mV_0)S_mV_1^{-1} - \frac{1}{\sigma_m}B_m(C - I)V_1D^{-1}V_1^{-1}. \quad (14)$$

Jenes Kriterium kommt durch

$$\begin{aligned} 0 &= c_i^l - \sum_{j=1}^s b_{ij} \frac{(c_j - 1)^l}{\sigma_m^l} + l \cdot \sum_{j=1}^s a_{ij} \frac{(c_j - 1)^{l-1}}{\sigma_m^{l-1}} + l \cdot \sum_{j=1}^{i-1} r_{ij} c_j^{l-1} \\ l \cdot \sum_{j=1}^s a_{ij} \frac{(c_j - 1)^{l-1}}{\sigma_m^{l-1}} &= c_i^l - \sum_{j=1}^s b_{ij} \frac{(c_j - 1)^l}{\sigma_m^l} + l \cdot \sum_{j=1}^{i-1} r_{ij} c_j^{l-1} \\ \sum_{j=1}^s a_{ij} (c_j - 1)^{l-1} &= c_i^l \frac{1}{l} \sigma_m^{l-1} - \frac{1}{l} \frac{\sigma_m^{l-1}}{\sigma_m^l} \sum_{j=1}^s b_{ij} (c_j - 1)^l + \sigma_m^{l-1} \sum_{j=1}^{i-1} r_{ij} c_j^{l-1} \\ \sum_{j=1}^s a_{ij} (c_j - 1)^{l-1} &= (c_i^l \frac{1}{l} - \sum_{j=1}^{i-1} r_{ij} c_j^{l-1}) \sigma_m^{l-1} - \frac{1}{\sigma_m} \sum_{j=1}^s b_{ij} (c_j - 1)^l \frac{1}{l} \end{aligned}$$

zustande.

Bemerkung 2:

Wir haben gezeigt, dass die explizite Peer-Methode über die Konsistenzordnung $p = s$ verfügt, wenn die Bedingungen (13) und (14) zutreffen. Hier liegt jedoch kein Einschrittverfahren vor, da wir nicht nur die Näherungen an den jeweils unmittelbar vorausgegangenen Stellen berechnen.

Daher besitzt das Peer-Verfahren die Konvergenzordnung p , wenn es die Konsistenzordnung p hat und optimal nullstabil ist - analog zu linearen Mehrschrittverfahren. Dies wäre im folgenden Abschnitt zu zeigen. Insbesondere betrachten wir konstante Matrizen B_m , dessen Koeffizienten unabhängig vom Schrittweitenverhältnis (3) sind. Dann werden wir auch überhaupt Nullstabilität definieren.

Das Kriterium (13) heißt Präkonsistenz allgemeiner linearer Methoden, und $\mathbb{1}$ ist der Präkonsistenzvektor.

5 Konvergenz

Betrachte wie angekündigt konstante Matrizen $B_m = B$. Im Folgenden zeigen wir, dass daraus zwingend Nullstabilität folgt.

Nun wählen wir die Matrizen B_m derart, dass sie konstant sind, also $B_m = B$ für alle m . Dies ermöglicht es uns nämlich, folgende Definition der Nullstabilität zu betrachten. Diese ist deshalb von Bedeutung, weil, wie bereits in letztem Abschnitt erwähnt, aus der Konsistenzordnung p sowie der Nullstabilität die Konvergenzordnung p folgt. Dass das explizite Peer-Verfahren Letzteres erfüllen soll, ist nun, wie der Name des Abschnittes schon andeutet, das Ziel.

Definition 2:

Sei U_m die gewünschte Approximation für das Anfangswertproblem (1), welche wir mit der s -stufigen expliziten Peer-Methode berechnen. Der Ausdruck U_{m-1} sei eben der vorherige Wert. B sei die zugehörige Differentialgleichung und für y' die Lösung, also gelte $U_m = BU_{m-1}$.

Ein lineares Mehrschrittverfahren - und somit auch das exponentielle Peer-Verfahren - heißt laut Bartels [4] nullstabil genau dann wenn für alle Eigenwerte

von B λ_i, λ_j

$$|\lambda_i| \leq 1 \tag{15}$$

und

$$\forall \lambda_i, \lambda_j : |\lambda_i| = |\lambda_j| = 1 : \lambda_i = \lambda_j \tag{16}$$

gilt.

Eine andere zulässige Definition der Nullstabilität besagt, siehe Podhaisky et al. [1], dass diese genau dann erfüllt ist, wenn eine Konstante K existiert so dass für alle $m, k \geq 0$

$$\|B_{m+k} \cdots B_{m+1} B_m\| \leq K \tag{17}$$

gilt.

Wir verschärfen die Bedingungen an B weiter. Wir fordern, dass B optimal nullstabil ist, um die Konvergenz sicherzustellen. Im darauffolgenden Satz zeigen wir nämlich, dass die explizite Peer-Methode Konvergenzordnung p hat, wenn sie über die Konsistenzordnung p verfügt und optimal nullstabil ist. Optimal nullstabil ist B dann, wenn für alle Eigenwerte

$$\lambda_1 = 1, \lambda_2 = \dots = \lambda_s = 0 \tag{18}$$

gilt.

Die Frage stellt sich, ob die Matrix B , mit unseren bisherigen Bedingungen, dem ebenfalls gerecht werden kann. Es sei also gezeigt, dass B überhaupt optimal nullstabil sein kann. Aus (13) folgt, dass zwingend ein Eigenwert $\lambda_1 = 1$ zum Eigenvektor $\mathbb{1}$ existiert. Somit ist es also möglich, unsere Matrix B im Einklang mit der zusätzlichen Eigenschaft der optimalen Nullstabilität zu wählen. Schließlich können wir B eben derart wählen, dass sie über den Eigenwert $\lambda_1 = 1$ zum Eigenvektor $\mathbb{1}$ verfügt und dass die verbleibenden Eigenwerte 0 sind. Und auch die zwei Bedingungen für die Nullstabilität können durch unser B erfüllt werden.

Es sei vorweggenommen, dass wir in Lemma 1 die Gestalt der Matrix B genauer einschränken werden, und zwar derart dass B optimal nullstabil ist.

Betrachte nun die Eigenwerte λ_i unserer Matrix B welche betragsmäßig am größten sind. Wir wünschen uns, dass die geometrische Vielfachheit der jeweiligen λ_i gleich 1 ist. Dann existiert laut Bulirsch und Stoer [3] (siehe Satz 6.9.2(b) auf den Seiten 81 - 82) eine Vektornorm, welche $\text{lub}(B) = \rho(B)$ besagt.

Hierbei ist $\rho(B)$ der Spektralradius von B . Die Norm $\text{lub}(B) = \max_{x \neq 0} \frac{\|Bx\|}{\|x\|}$ ist eine Matrixnorm zu einer von Bulirsch und Stoer [3], wieder in Satz 6.9.2(b), eingeführten Vektornorm $\|x\|$ für $x \in \mathbb{C}^2$. Die Matrix B soll wie gesagt optimal nullstabil sein, also über die Eigenwerte in (18) verfügen. Dessen betragsmäßig größter Eigenwert ist also $\lambda_1 = 1$, während alle anderen Eigenwerte gleich 0 sind.

Somit ist die algebraische Vielfachheit des Eigenwertes $\lambda_1 = 1$ gleich 1. Und da die algebraische Vielfachheit stets größer und gleich der geometrischen Vielfachheit ist, ist auch die zugehörige geometrische Vielfachheit gleich 1. Hiermit ist also gezeigt, dass eine Matrixnorm existiert so dass $\|B\| = \rho(B) = 1$ ist, da eben der Spektralradius von B gleich 1 ist. Einführen tun wir die Matrixnorm

$$\|B\| = \rho(B) = 1 \quad (19)$$

wie gesagt deshalb, da wir sie im folgenden Konvergenzsatz aufgreifen werden.

Betrachte aber zunächst den globalen Fehler $\epsilon_m = Y(t_m) - U_m$ mit

$$Y(t_m) = \begin{pmatrix} y(t_m + c_1 h_m) \\ \dots \\ y(t_m + c_s h_m) \end{pmatrix}.$$

Satz 2:

Gegeben seien $h_{max} = \max_m h_m$, $\sigma_m \leq \sigma_{max}$ und $\epsilon_0 = O(h_0^s)$ - letzteres stelle den Fehler der Startwerte dar.

Darüber hinaus sei die explizite Peer-Methode (3) optimal nullstabil und es treffe (13) zu. Für $\sigma_m \leq \sigma_{max}$ seien die Koeffizientenmatrizen R_m gleichmäßig beschränkt. Die Koeffizientenmatrix A_m erfülle die Bedingung (14).

Dann ist das Peer-Verfahren konvergent von der Ordnung $p = s$, da $\epsilon_m = O(h_{max}^s)$ für $h_{max} \rightarrow 0$ gilt.

Beweis:

In unserem Beweis betrachten wir, der Einfachheit unserer Schreibweise halber, den skalaren Fall $n = 1$.

Betrachte zunächst $U_{m,i}$, $i \in 1, \dots, s$ aus (3). Benutze außerdem

$$Y(t_m) = \begin{pmatrix} y(t_m + c_1 h_m) \\ \dots \\ y(t_m + c_s h_m) \end{pmatrix},$$

wie oben bereits eingeführt, sowie

$$U_m = \begin{pmatrix} U_{m,1} \\ \dots \\ U_{m,s} \end{pmatrix}$$

und

$$F(t_m, U_m) = \begin{pmatrix} f(t_{m,1}, U_{m,1}) \\ \dots \\ f(t_{m,s}, U_{m,s}) \end{pmatrix}.$$

Zudem benötigen wir als Vorbereitung den Satz von Picard-Lindelöf. Betrachte, siehe (1),

$$y' = f(t, y), t \in [t_0, t_e]$$

$$y(t_0) = y_0,$$

also ein Anfangswertproblem gewöhnlicher Differentialgleichungen. Dieses Anfangswertproblem kann in einer beliebigen Umgebung von t_0 eindeutig gelöst werden kann, wenn f Lipschitz-stetig ist.

Da wir nur Anfangswertprobleme betrachten möchten, indem dies möglich ist, setzen wir in unserem Beweis die Lipschitz-Stetigkeit von f voraus. Es gilt

$$\epsilon_m = Y(t_m) - U_m$$

und daher

$$\begin{aligned} \epsilon_m &= B\epsilon_{m-1} + h_m A_m (F(t_{m-1}, Y(t_{m-1})) - F(t_{m-1}, U_{m-1})) \\ &+ h_m R_m (F(t_m, Y(t_m)) - F(t_m, U_m)) + \Delta_m. \end{aligned}$$

Die explizite Peer-Methode ist in unserem Fall konsistent von der Ordnung $p = s$, da der zu beweisende Satz die Bedingungen (13) und (14) an B_m beziehungsweise A_m voraussetzt.

Nutze im Folgenden die Matrixnorm aus (19). Sie liefert $\|B\| = 1$, woraus $\|B\epsilon_{m-1}\| \leq \|B\| \|\epsilon_{m-1}\| = \|\epsilon_{m-1}\|$ folgt aufgrund der Homogenität der Norm. Aus der Konsistenz von der Ordnung $p = s$ folgt $\Delta_{m,i} = \mathcal{O}(h_m^{p+1})$ und somit $|\Delta_{m,i}| \leq d_1 h_m^{s+1}$, jeweils für $i = 1, \dots, s$.

Durch die Lipschitzkonstante L schätzen wir f nach oben ab, siehe die Lipschitzstetigkeit von f oben. Ferner nutzen wir $\epsilon_m = Y(t_m) - U_m$ beziehungsweise

$\epsilon_{m-1} = Y(t_{m-1}) - U_{m-1}$ und es folgt

$$\begin{aligned}\|\epsilon_m\| &\leq \|\epsilon_{m-1}\| + h_m \cdot \|A_m\| \cdot \|\epsilon_{m-1}\| + h_m \cdot \|R_m\| \cdot \|\epsilon_m\| + d_1 h m^{s+1} \\ &\leq \|\epsilon_{m-1}\| + h_m \cdot \|A_m\| \cdot \|\epsilon_{m-1}\| + h_m \cdot \|R_m\| \cdot \|\epsilon_m\| + d_1 h m^{s+1}.\end{aligned}$$

Für $\sigma_m \leq \sigma_{max}$ sei R_m nach Voraussetzung unseres Satzes beschränkt. Wir zeigen nun, dass $\sigma_m A_m$ ebenfalls beschränkt sein muss. Laut (14) gilt nämlich

$$\sigma_m A_m = \sigma_m (C V_0 D^{-1} - R_m V_0) S_m V_1^{-1} - B_m (C - I) V_1 D^{-1} V_1^{-1}$$

- siehe (14) für die zugehörigen Ausdrücke.

Zunächst sind σ_m und S_m offensichtlich beschränkt. Was B_m betrifft, hatten wir zu Beginn des fünften Abschnitts festgelegt, dass wir zugunsten der Nullstabilität konstante Matrizen $B_m = B$ betrachten - was heißt, dass B_m nicht unbeschränkt sein kann.

Dass R_m beschränkt ist, haben wir wie gesagt vorausgesetzt. Die übrigen Terme sind fest und daher grundsätzlich beschränkt. Damit ist die Beschränktheit von $\sigma_m A_m$ gezeigt.

Somit gilt, mit (3),

$$\begin{aligned}\|\epsilon_m\| &\leq (1 + h_{m-1} d_2) \|\epsilon_{m-1}\| + h_{m-1} d_3 \|\epsilon_m\| + d_4 h_{m-1}^{s+1} \\ (1 + h_{m-1} d_3) \|\epsilon_m\| &\leq (1 + h_{m-1} d_2) \|\epsilon_{m-1}\| + d_4 h_{m-1}^{s+1}.\end{aligned}$$

Wähle d_5 und d_6 so dass $\frac{1+h_{m-1}d_2}{1+h_{m-1}d_3} = 1 + h_{m-1}d_5$ und $\frac{d_4 h_{m-1}^{s+1}}{1+h_{m-1}d_3} = d_6 h_{m-1}^{s+1}$ gilt - es folgt

$$\begin{aligned}\|\epsilon_m\| &\leq d_6 h_{m-1}^{s+1} + (1 + h_{m-1} d_5) \|\epsilon_{m-1}\| \\ &\leq d_6 h_{m-1}^{s+1} + e^{(h_{m-1} d_5)} \|\epsilon_{m-1}\| \\ &\leq d_6 h_{m-1}^{s+1} + e^{(h_{m-1} d_5)} (d_6 h_{m-2}^{s+1} + e^{(h_{m-2} d_5)} \|\epsilon_{m-2}\|) \\ &\leq d_6 h_{m-1}^{s+1} + d_6 h_{m-2}^{s+1} e^{(h_{m-1} d_5)} + e^{(h_{m-1} + h_{m-2}) d_5} (d_6 h_{m-3}^{s+1} + e^{(h_{m-3} d_5)} \|\epsilon_{m-3}\|).\end{aligned}$$

Wiederholtes Einsetzen liefert dann

$$\|\epsilon_m\| \leq d_6 h_{max}^{s+1} (h_{m-1} + h_{m-2} e^{(h_{m-1} d_5)} + h_{m-3} e^{(h_{m-1} + h_{m-2}) d_5} + \dots + h_0 e^{(h_{m-1} + \dots + h_1) d_5}) + e^{(t_m - t_0) d_5} \|\epsilon_0\|.$$

Für

$$\int_{t_i}^{t_{i+1}} e^{(t_m-t)d_5} dt \geq (t_{i+1} - t_i)e^{(t_m-t_{i+1})d_5} = h_i e^{(t_m-t_{i+1})d_5}$$

und

$$\|e_0\| = 0$$

gilt also

$$\begin{aligned} \|e_m\| &\leq Ch_{max}^p \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} e^{(t_m-t)d_5} dt \\ &= Ch_{max}^p \int_{t_0}^{t_m} e^{(t_m-t)d_5} dt \\ &= \frac{Ch_{max}^p}{d_5} (e^{(t_m-t_0)d_5} - 1). \end{aligned}$$

Insgesamt folgt daher die Abschätzung

$$\|\epsilon_m\| \leq \frac{Ch_{max}^p}{d_5} (e^{(t_m-t_0)d_5} - 1) + e^{(t_m-t_0)d_5} \|\epsilon_0\|.$$

Damit ist gezeigt, dass unter der Voraussetzung an die Startwerte gilt: $\epsilon_m = O(h_{max}^s)$ für $h_{max} \rightarrow 0$. Somit ist die explizite Peer-Methode konvergent, wenn die Bedingung $\epsilon_0 = O(h_0^s)$ erfüllt wird. \square

6 Superkonvergenz

Betrachte das Residuum (5) sowie den globalen Fehler ϵ_m von unserer expliziten Peer-Methode. Das Residuum ist in unserem Fall der lokale Diskretisierungsfehler, im Gegensatz zu dem globalen Fehler welcher den globalen Diskretisierungsfehler darstellt.

Wie bereits ausgeführt, lauten unsere Anforderungen an die explizite Peer-Methode, dass sowohl die Konsistenzordnung (welche sich auf den lokalen Diskretisierungsfehler bezieht) als auch die Konvergenzordnung (globaler Diskretisierungsfehler) jeweils $p = s$ beträgt.

Denn es gilt, siehe Satz 1 beziehungsweise Satz 2: Gilt die Konsistenzordnung

p , so ist das lokale Residuum $\Delta_{m,i} := \mathcal{O}(h_m^{p+1})$ für alle $i = 1, \dots, s$ und gilt die Konvergenzordnung p , so ist der globale Fehler $\epsilon_m = \mathcal{O}(h_{max}^s)$ für $s = p$ und $h_{max} \rightarrow 0$. Demzufolge ist $\Delta_{m,i}$ aber um eine h -Potenz größer als ϵ_m .

In diesem Abschnitt wollen wir nun superkonvergente Verfahren entwickeln. Eine explizite Peer-Methode, welche superkonvergent ist, verfügt über die Konvergenzordnung $p = s + 1$. Dies hat den Vorteil, dass $\epsilon_m = \mathcal{O}(h_{max}^{s+1})$ für $h_{max} \rightarrow 0$ gilt und die Größe der h -Potenz gleich groß wie beim lokalen Residuum ist.

Gebe nun erstmal einen Ausblick auf diesen Abschnitt, inklusive Wiederholung des vorherigen Abschnitts. Lege zunächst Bedingungen für unser $B = B_m$ fest, mit dem Ziel, optimale Nullstabilität und darüber hinaus Superkonvergenz zu sichern. $B = B_m$ gilt in unserem Fall deshalb, da wir mit konstanten Matrizen B dafür sorgen, dass die explizite Peer-Methode nullstabil ist. Aus Nullstabilität und Konsistenzordnung p folgt Konvergenzordnung p .

Außerdem setzen wir, siehe (13), $B\mathbb{1} = \mathbb{1}$ voraus. Dies ermöglicht es, wie wir gezeigt hatten, dass (15), (16) und (18) gelten und somit optimale Nullstabilität erfüllt wird. Wenn zudem A_m , in Abhängigkeit eines beliebigen R_m und eben unseres B_m , derart bestimmt wird dass (14) gilt, dann verfügt das explizite Peer-Verfahren über die Konvergenzordnung $p = s$.

Steige in den Abschnitt ein, indem wir eine Matrix B studieren, welche unsere erwünschten Kriterien erfüllt. Anschließend werden wir beweisen, wie wir andere Koeffizienten wie zum Beispiel von der Matrix R bestimmen müssen, um ein superkonvergentes explizites Peer-Verfahren zu erhalten. Zum Schluss erhalten wir ein Konstruktionsprinzip für dieses Verfahren. Dieses Prinzip besagt, in welchen Schritten wir vorgehen, um eine superkonvergente Peer-Methode zu erhalten, also, wie und in welcher Reihenfolge wir die Koeffizienten bestimmen.

Gelte

$$B = \mathbb{1}v^T + QWQ^{-1} \tag{20}$$

mit

$$v = \begin{pmatrix} \tilde{\sigma} \\ 1 - \tilde{\sigma}\mathbb{1} \end{pmatrix}, \quad W = \begin{pmatrix} \tilde{W} & -\tilde{W}\mathbb{1} \\ 0^T & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} (I - \mathbb{1}\tilde{\sigma}^T)\tilde{Q} & (1 + \tilde{\sigma}^T\tilde{Q}\mathbb{1})\mathbb{1} - \tilde{Q}\mathbb{1} \\ -\tilde{\sigma}^T\tilde{Q} & 1 + \tilde{\sigma}^T\tilde{Q}\mathbb{1} \end{pmatrix}.$$

Hierbei sei \tilde{W} eine streng obere Dreiecksmatrix, \tilde{Q} eine reguläre Matrix und $\tilde{\sigma}$ beliebig.

Lemma 1:

Sei B wie in (20) gegeben. Dann ist die explizite Peer-Methode optimal nullstabil und erfüllt jeweils

$$B\mathbb{1} = \mathbb{1}, \quad B^j = \mathbb{1}v^T \quad (21)$$

für $j \geq s - 1$.

Beweis:

Zunächst zeigen wir, dass mit (20) folgende Eigenschaften gelten: $Q\mathbb{1} = \mathbb{1}$,

$W\mathbb{1} = 0$, $v^T Q = e_s^T$.

Es gilt

$$\begin{aligned} (Q\mathbb{1})_{i \in \{1, 2, \dots, s-1\}} &= \tilde{Q}\mathbb{1} - \mathbb{1}\tilde{\sigma}^T\tilde{Q}\mathbb{1} + \mathbb{1} + \tilde{\sigma}^T\tilde{Q}\mathbb{1}\mathbb{1} - \tilde{Q}\mathbb{1}\mathbb{1} \\ &= \tilde{Q}\mathbb{1} - \tilde{\sigma}^T\tilde{Q}\mathbb{1} + \mathbb{1} + \tilde{\sigma}^T\tilde{Q}\mathbb{1} - \tilde{Q}\mathbb{1} \\ &= \mathbb{1}, \end{aligned}$$

$$\begin{aligned} (Q\mathbb{1})_s &= -\tilde{\sigma}^T\tilde{Q}\mathbb{1} + \mathbb{1} + \tilde{\sigma}^T\tilde{Q}\mathbb{1} \\ &= \mathbb{1}, \end{aligned}$$

$$\begin{aligned} (W\mathbb{1})_{i \in \{1, 2, \dots, s-1\}} &= \sum_{j=1}^{s-1} (\tilde{w}_{ij}\mathbb{1}_j) - \sum_{j=1}^{s-1} (\tilde{w}_{ij}\mathbb{1}_j)\mathbb{1}_s \\ &= \sum_{j=1}^{s-1} (\tilde{w}_{ij}\mathbb{1}_j) - \sum_{j=1}^{s-1} (\tilde{w}_{ij}\mathbb{1}_j) \\ &= 0, \end{aligned}$$

$$\begin{aligned} (W\mathbb{1})_s &= \sum_{i=1}^s (0\mathbb{1}_i) + 0\mathbb{1}_s \\ &= 0, \end{aligned}$$

$$\begin{aligned} (v^T Q)_{j \in \{1, 2, \dots, s-1\}} &= \sum_{j=1}^{s-1} \tilde{\sigma}_j q_{ij} + (1 - \tilde{\sigma}^T \mathbb{1}) q_{is} \\ &= \sum_{j=1}^{s-1} \tilde{\sigma}_j (\tilde{q}_{ij} - \mathbb{1}_i \tilde{\sigma}_i \tilde{q}_{ij}) + (1 - \tilde{\sigma}^T \mathbb{1}) (-\tilde{\sigma}^T \tilde{Q}) \end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^{s-1} \tilde{v}_j(\tilde{q}_{ij} - \mathbb{1}_i \tilde{v}_i \tilde{q}_{ij}) + \tilde{v}^T \mathbb{1} \tilde{v}^T \tilde{Q} - \tilde{v}^T \tilde{Q} \\
&= \tilde{v}^T \tilde{Q} - \tilde{v}^T \mathbb{1} \tilde{v}^T \tilde{Q} + \tilde{v}^T \mathbb{1} \tilde{v}^T \tilde{Q} - \tilde{v}^T \tilde{Q} \\
&= 0,
\end{aligned}$$

$$\begin{aligned}
(v^T Q)_s &= (v^T)_{i \in \{1, 2, \dots, s-1\}} ((1 + \tilde{v}^T \tilde{Q} \mathbb{1}) \mathbb{1} - \tilde{Q} \mathbb{1}) + (v_s^T) (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) \\
&= \tilde{v}^T ((1 + \tilde{v}^T \tilde{Q} \mathbb{1}) \mathbb{1} - \tilde{Q} \mathbb{1}) + (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) - \tilde{v}^T \mathbb{1} (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) \\
&= \tilde{v}^T \mathbb{1} (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) - \tilde{v}^T \tilde{Q} \mathbb{1} + (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) - \tilde{v}^T \mathbb{1} (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) \\
&= -\tilde{v}^T \tilde{Q} \mathbb{1} + (1 + \tilde{v}^T \tilde{Q} \mathbb{1}) \\
&= 1.
\end{aligned}$$

Damit haben wir nachgewiesen, dass jeweils $Q\mathbb{1} = \mathbb{1}$, $W\mathbb{1} = 0$ und $v^T Q = e_s^T$ gelten.

Wir können also (20) umformulieren als

$$\begin{aligned}
B &= \mathbb{1}v^T + QWQ^{-1} \\
&= Q\mathbb{1}e_s^T Q^{-1} + QWQ^{-1} \\
&= Q(\mathbb{1}e_s^T + W)Q^{-1}.
\end{aligned}$$

Die Matrix Q ist laut Voraussetzung regulär. Es existiert also die reguläre Matrix Q so dass $B = Q(\mathbb{1}e_s^T + W)Q^{-1}$ gilt. Damit sind B und $\mathbb{1}e_s^T + W$ zueinander ähnlich, und haben somit die gleichen Eigenwerte.

In der Matrix $\mathbb{1}e_s^T$ sind alle Einträge der s -ten Spalte, und somit auch der unterste Diagonaleintrag, gleich 1. Alle anderen Elemente, und somit auch alle verbleibenden Diagonaleinträge, sind 0. Darum handelt es sich auch um eine obere Dreiecksmatrix. $\mathbb{1}e_s^T + W$ ist ebenfalls eine obere Dreiecksmatrix, da wir die obere Dreiecksmatrix $\mathbb{1}e_s^T$ mit der laut Voraussetzung streng oberen Dreiecksmatrix W addiert haben. Diese Voraussetzung liefert auch, dass alle Diagonalelemente von W gleich 0 sind. Daraus resultiert, dass $\mathbb{1}e_s^T + W$ wiederum einen Diagonaleintrag gleich 1 hat, während die restlichen Diagonalelemente gleich 0 sind.

Da in einer Dreiecksmatrix die Elemente der Hauptdiagonale die Eigenwerte darstellen, ist $\mathbb{1}e_s^T + W$ optimal nullstabil, siehe (18). Oben begründeten wir bereits, dass aufgrund der Ähnlichkeit folgt, dass B dieselben Eigenwerte hat - somit ist B ebenfalls optimal nullstabil. Ein Teil des Lemmas ist also bereits bewiesen.

Es gilt zudem

$$\begin{aligned}
B\mathbb{1} &= Q(\mathbb{1}e_s^T + W)Q^{-1}\mathbb{1} \\
&= Q(\mathbb{1}e_s^T + W)\mathbb{1} \\
&= Q(\mathbb{1}e_s^T\mathbb{1} + W\mathbb{1}) \\
&= Q(\mathbb{1} + 0) \\
&= \mathbb{1}.
\end{aligned}$$

Zeige schließlich, dass $B^j = \mathbb{1}v^T$ für $j \geq s - 1$ gilt. Dazu verwenden wir zwei neue Eigenschaften. Wir haben bereits $\mathbb{1} = Q\mathbb{1}$ nachgewiesen. Selbstredend folgt daraus $Q^{-1}\mathbb{1} = \mathbb{1}$ und damit die Eigenschaft $WQ^{-1}\mathbb{1} = W\mathbb{1} = 0$.

Dass $v^T Q = e_s^T$ gilt, nutzen wir, um die weitere Eigenschaft $v^T QW = 0^T$ nachzuweisen. Zunächst gilt offensichtlich $v^T QW = e_s^T W$. Da die letzte Zeile von W , siehe (20), eine Nullzeile ist und nur der letzte Eintrag von e_s^T ungleich 0 ist, folgt $v^T QW = e_s^T W = 0^T$.

Es gilt, siehe oben,

$$\begin{aligned}
B &= Q(\mathbb{1}e_s^T + W)Q^{-1} \\
&= Q\mathbb{1}e_s^T Q^{-1} + QWQ^{-1} \\
&= \mathbb{1}e_s^T Q^{-1} + QWQ^{-1} \\
&= \mathbb{1}v^T Q Q^{-1} + QWQ^{-1} \\
&= \mathbb{1}v^T + QWQ^{-1}.
\end{aligned}$$

Mit $v^T QW = 0^T$ ist

$$\begin{aligned}
B^j &= (\mathbb{1}v^T)^j + (QWQ^{-1})^j \\
&= \mathbb{1}(v^T \mathbb{1})^{j-1} v^T + QW(Q^{-1}QW)^{j-1} Q^{-1} \\
&= \mathbb{1}(\tilde{v}^T \mathbb{1}_{s-1} + 1 - \tilde{v}^T \mathbb{1}_{s-1})^{j-1} v^T + QW^j Q^{-1} \\
&= \mathbb{1}v^T,
\end{aligned}$$

denn da W eine streng obere Dreiecksmatrix ist, muss W nilpotent sein. Damit wäre also $W^j = 0$ für $j \geq s$ nachgewiesen, da W eine $(s \times s)$ -Matrix ist. Warum ist jedoch auch $W^{s-1} = 0$?

Dies liegt daran, dass auch \tilde{W} eine streng untere Dreiecksmatrix ist. Und da somit die letzte Zeile von $\tilde{W} = 0$ ist, so ist auch der letzte Eintrag des Vektors $\tilde{W}\mathbb{1} = 0$. Somit sind also die letzten beiden Zeilen der Matrix W gleich 0, weshalb bereits $W^{s-1} = 0$ sein muss. Damit ist Lemma 1 bewiesen. \square

Damit haben wir die Gestalt einer Matrix B bestimmt, welche, wie wir auch im Folgenden zeigen werden, für unser Peer-Verfahren in Frage kommt. Selbstredend müssen wir auch die möglichen Einträge von \tilde{v} , \tilde{W} und \tilde{Q} bestimmen, um überhaupt ein festes B zu erhalten. Wir werden im nächsten Satz unter anderem eine weitere Bedingung aufstellen, welche B erfüllen soll. Für den Beweis des Satzes benötigen wir jedoch folgende Bemerkung:

Bemerkung 3:

Sei $y' = f(t, y)$, für $f(t, y)$ derart dass $f : I \times \Omega \rightarrow \mathbb{R}^n$ ($I \in \mathbb{R}$ abgeschlossen, offen oder halboffen) ist. f sei in einer offenen Menge $D \supset G$ stetig differenzierbar bezüglich y . Dann besagt der Mittelwertsatz für Vektorfunktionen

$$f(t, u) - f(t, v) = \int_0^1 f_y(t, v + \theta(u - v))(u - v) d\theta, \quad (t, u), (t, v) \in G.$$

Diesen Mittelwertsatz nutzen wir im folgenden Satz, um den globalen Fehler ϵ_m abzuschätzen.

Satz 3:

Seien $h = \max_m h_m$, $\max_m \sigma_m \leq \sigma_{max}$ und $\epsilon_0 = \mathcal{O}(h_0^{s+1})$. Seien außerdem die streng unteren Dreiecksmatrizen R_m gleichmäßig beschränkt für $\sigma_m \leq \sigma_{max}$. A_m sei durch (14) sowie B durch (20) bestimmt. Zudem gelte die Bedingung

$$v^T AB(s + 1) = 0. \tag{22}$$

Dann gilt, dass die explizite Peer-Methode superkonvergent von der Ordnung $p = s + 1$ ist, also, dass $\epsilon_m = \mathcal{O}(h^{s+1})$ ist.

Beweis:

Im Beweis betrachten wir autonome Differentialgleichungen um uns in unserer Schreibweise kürzer zu fassen. Zudem betrachten wir den skalaren Fall - analog zum Beweis des Satzes 2. Überdies beschränken wir uns auf konstante Schrittweiten $h = h_m$. Somit sind auch die Matrizen $A_m = A$ und $R_m = R$ konstant.

Zunächst gilt

$$\epsilon_m = Y(t_m) - U_m = B\epsilon_{m-1} + hA(F(Y(t_{m-1})) - F(U_{m-1})) + hR(F(Y(t_m)) - F(U_m)) + \Delta_m.$$

Die Voraussetzungen aus Satz 3 garantieren uns, dass die Konvergenzordnung gleich p , und $\|\epsilon_m\| \leq Ch^s$ ist, siehe Satz 2 (es sei noch einmal erwähnt, dass das B von Form (20) die optimale Nullstabilität garantiert).

Verwende nun den Mittelwertsatz aus Bemerkung 3 - er liefert für $i = 1, \dots, s$

$$\begin{aligned} f(y(t_{m-1,i})) - f(U_{m-1,i}) &= J_{m-1,i}(U_{m-1,i} - y(t_{m-1,i})) \\ &= J_{m-1,i}\epsilon_{m-1,j} \end{aligned}$$

und

$$J_{m-1,i} = \int_0^1 f_y(y(t_{m-1,i}) + \theta(U_{m-1,i} - y(t_{m-1,i}))) d\theta.$$

Dann gilt

$$\begin{aligned} A \cdot (F(Y(t_{m-1})) - F(U_{m-1})) &= A \begin{pmatrix} J_{m-1,1}\epsilon_{m-1,1} \\ \dots \\ J_{m-1,s}\epsilon_{m-1,s} \end{pmatrix} \\ &= A \cdot \text{diag}(J_{m-1,i}) \cdot \epsilon_{m-1} \\ &= G_m \epsilon_{m-1} \end{aligned}$$

für $A \cdot \text{diag}(J_{m-1,i}) = G_m$.

Analog gilt

$$R \cdot (F(Y(t_m)) - F(U_m)) = H_m \epsilon_m$$

für $R \cdot \text{diag}(J_{m,i}) = H_m$.

Setze oben ein und erhalte die Rekursion

$$\epsilon_m = B\epsilon_{m-1} + hG_m\epsilon_{m-1} + hH_m\epsilon_m + \Delta_m.$$

Laut Voraussetzung sind die Matrizen R_m streng untere Dreiecksmatrizen. Daher können wir die Komponenten von ϵ_m im Term $hH_m\epsilon_m$ nacheinander von unten nach oben berechnen - obwohl die Darstellung formal implizit aussieht.

Nun wollen wir die Rekursion auf ϵ_0 zurückzuführen. Wir können uns damit die Anfangsvoraussetzung $\epsilon_0 = \mathcal{O}(h_0^{s+1})$ zunutze machen. Setze also zunächst

$$\epsilon_{m-1} = B\epsilon_{m-2} + hG_{m-1}\epsilon_{m-2} + hH_{m-1}\epsilon_{m-1} + \Delta_{m-1}$$

ein, es folgt

$$\epsilon_m = B(B\epsilon_{m-2} + hG_{m-1}\epsilon_{m-2} + hH_{m-1}\epsilon_{m-1} + \Delta_{m-1}) + hG_m\epsilon_{m-1} + hH_m\epsilon_m + \Delta_m$$

$$= B^2 \epsilon_{m-2} + hBG_m \epsilon_{m-2} + hG_m \epsilon_{m-1} + hBH_{m-1} \epsilon_{m-1} + hH_m \epsilon_m + \Delta_m + B\Delta_{m-1}.$$

Sukzessives Einsetzen der ϵ_{m-i} durch ϵ_{m-i-1} für $i \in \{0, \dots, m\}$ liefert

$$\epsilon_m = B^m \epsilon_0 + h \sum_{j=0}^{m-1} B^j G_{m-j} \epsilon_{m-j-1} + h \sum_{j=0}^{m-1} B^j H_{m-j} \epsilon_{m-j} + \sum_{j=0}^{m-1} B^j \Delta_{m-j}. \quad (23)$$

Anschließend spalten wir die letzte Summe $\sum_{j=0}^{m-1} B^j \Delta_{m-j}$ auf in den beiden Summen

$$\sum_{j=0}^{m-1} B^j \Delta_{m-j} = \sum_{j=s-1}^{m-1} B^j \Delta_{m-j} + \sum_{j=0}^{s-2} B^j \Delta_{m-j}. \quad (24)$$

Voraussetzung dieses Satzes ist es, dass die Matrix B durch (20) festgelegt ist. Und wenn dies der Fall ist, so folgt aus Lemma 1, dass, für $j \geq s-1$ wohl-gemerkt, $B^j = \mathbb{1}v^T$ ist. Somit gilt für die Summanden der ersten Summe aus (24)

$$\begin{aligned} B^j \Delta_{m-j} &= \mathbb{1}v^T \Delta_{m-j} \\ &= \mathbb{1}v^T \begin{pmatrix} \Delta_{m-j,1} \\ \dots \\ \Delta_{m-j,s} \end{pmatrix} \\ &= \mathbb{1}v^T \left\{ \begin{pmatrix} (1 - \sum_{j=1}^s b_{1j})y(t_m) \\ \dots \\ (1 - \sum_{j=1}^s b_{sj})y(t_m) \end{pmatrix} + \sum_{l=1}^{s+1} AB(l) \frac{h_m^l \cdot y^{(l)}(t_m)}{l!} \right\} + \mathcal{O}(h_m^{s+2}) \\ &= \mathbb{1}v^T \mathbb{1}y(t_m) - \mathbb{1}v^T \left\{ \begin{pmatrix} \sum_{j=1}^s b_{1j} \\ \dots \\ \sum_{j=1}^s b_{sj} \end{pmatrix} y(t_m) - \sum_{l=1}^{s+1} AB(l) \frac{h_m^l \cdot y^{(l)}(t_m)}{l!} \right\} + \mathcal{O}(h_m^{s+2}), \end{aligned}$$

in dem Fall aber für $l = 0, 1, \dots, s+1$, da wir im Gegensatz zu Satz 1 die Konvergenzordnung $p = s+1$ nachweisen wollen.

Um fortzufahren, benötigen wir nun vier Aussagen. Erstens soll $\mathbb{1}v^T \mathbb{1} = \mathbb{1}$ gelten. Zweitens soll, für alle $i \in 1, \dots, s$, die Summe $\sum_{j=1}^s b_{ij} = 1$ sein. Und drittens sollen die Voraussetzungen dieses zu beweisenden Satzes ausreichend dafür sein, dass die explizite Peer-Methode konvergent von der Ordnung $p = s$ ist (nicht zu verwechseln mit der Superkonvergenz von der Ordnung $p = s+1$, welche wir ja hier erst beweisen wollen).

Zeige diese drei Behauptungen nun. Die erste Aussage weisen wir mithilfe

der Eigenschaften, die wir in Lemma 1 gezeigt haben, nach. Es gilt

$$\begin{aligned}\mathbb{1}v^T\mathbb{1} &= \mathbb{1}v^TQ\mathbb{1} \\ &= \mathbb{1}v^Te_s^T\mathbb{1} \\ &= \mathbb{1} \cdot \mathbb{1},\end{aligned}$$

womit die erste Behauptung gezeigt wäre.

Um nun $\sum_{j=1}^s b_{ij} = 1$ für alle $i \in 1, \dots, s$ nachzuweisen, nutzen wir Lemma 1. Schließlich fordern wir in unserem Satz, dass B wie in (20) gegeben ist, und Lemma 1 liefert, dass in dem Fall $B\mathbb{1} = \mathbb{1}$ gilt. Und wenn diese Gleichung gilt, dann ist für alle $i \in 1, \dots, s$

$$\begin{aligned}\sum_{i=1}^s b_{ij}\mathbb{1}_i &= \mathbb{1}_j \\ &= 1.\end{aligned}$$

Bleibt also noch zu zeigen, dass die Bedingungen unseres Satzes garantieren, dass die explizite Peer-Methode konvergent von der Ordnung $p = s$ ist. Dazu werfen wir einen Blick auf den bereits bewiesenen Satz 2, welcher hinreichende Bedingungen dafür liefert, dass unser Verfahren eben von der Ordnung $p = s$ konvergent ist.

Dass wir in Satz 3, also in unserem Satz, $\epsilon_0 = O(h_0^{s+1})$ fordern, ist eine stärkere Aussage, als es die Bedingung $\epsilon_0 = O(h_0^s)$ aus Satz 2 ist. Und wenn B , wie in Satz 3 gefordert, durch (20) gegeben ist, so gilt laut Lemma 1 $B\mathbb{1} = \mathbb{1}$, was einer weiteren Bedingung aus Satz 2 entspricht. Alle anderen Voraussetzungen des Satzes 2 sind auch in Satz 3 ebensolche.

Diese drei Bedingungen werden wir nun verwenden, um den Ausdruck $B^j\Delta_{m-j}$ weiter umzuformulieren. Insbesondere kommt mit der dritten Aussage Satz 1 zum Tragen. Damit können in der Summe mit den $AB(l)$ die ersten s Summanden entfallen, da wir eben annehmen können, dass die Konvergenzordnung $p = s$ gilt.

Es gilt also

$$\begin{aligned}B^j\Delta_{m-j} &= \mathbb{1}v^T\mathbb{1}y(t_m) - \mathbb{1}v^T\left\{\begin{pmatrix} \sum_{j=1}^s b_{1j} \\ \dots \\ \sum_{j=1}^s b_{sj} \end{pmatrix}y(t_m) - AB(s+1)\frac{h_m^{s+1} \cdot y^{(s+1)}(t_m)}{(s+1)!}\right\} + O(h_m^{s+2}) \\ &= \mathbb{1}v^T\mathbb{1}y(t_m) - \mathbb{1}v^T\left\{\mathbb{1}y(t_m) - AB(s+1)\frac{h_m^{s+1} \cdot y^{(s+1)}(t_m)}{(s+1)!}\right\} + O(h_m^{s+2})\end{aligned}$$

$$\begin{aligned}
&= \mathbb{1}v^T AB(s+1) \frac{h_m^{s+1} \cdot y^{(s+1)}(t_m)}{(s+1)!} + \mathcal{O}(h_m^{s+2}\hat{\epsilon}) \\
&= \mathcal{O}(h_m^{s+2}),
\end{aligned}$$

denn im letzten Schritt wenden wir einfach (22) an.
Daraus folgt definitionsgemäß

$$B^j \Delta_{m-j} \leq Ch^{s+2}$$

und die erste Summe aus (24) können wir dann durch

$$\begin{aligned}
\left\| \sum_{j=s-1}^{m-1} B^j \Delta_{m-j} \right\| &\leq C(m-1-(s-2))h^{s+2} \\
&= C(m-s+1)h^{s+2} \\
&= C(t_e - t_0)h^{s+1}
\end{aligned}$$

abschätzen - schließlich ist t_0 die Anfangs- und t_e die Endzeit, während h die Schrittweite ist.

Betrachte nun die zweite Summe aus (24). Im Gegensatz zur ersten Summe können wir hier nicht B^j durch $\mathbb{1}v^T$ ersetzen - daher kommt auch nicht (22) zum Tragen. Aufgrund von Satz 1, und weil laut Annahme unser Verfahren die Konvergenzordnung $p = s$ hat, können wir aber $\Delta_{m,i} := \mathcal{O}(h_m^{p+1})$ setzen.

Zudem nutzen wir (17) aus der Definition der Nullstabilität. Dies ist erlaubt, denn B ist laut Annahme durch (20) gegeben und dies wiederum bedeutet laut Lemma 1, dass B optimal nullstabil und damit auch nullstabil ist. Mit (17) können wir $\|B^j\| \leq K$ für eine beliebige Konstante K annehmen - beachte hierbei, dass wir konstante Matrizen B betrachten und deshalb B^j notieren.

Insgesamt folgt also, da diese zweite Summe über $s-1$ Summanden verfügt,

$$\sum_{j=0}^{s-2} B^j \Delta_{m-j} \leq CK(s-1)h^{s+1},$$

und für eine Konstante C_1 mit $C_1 \geq C(Ks - K + t_e - t_0)$ gilt somit

$$\sum_{j=0}^{m-1} B^j \Delta_{m-j} = \sum_{j=s-1}^{m-1} B^j \Delta_{m-j} + \sum_{j=0}^{s-2} B^j \Delta_{m-j} \leq C_1 h^{s+1}.$$

Die Matrizen $A_m = A$ und $R_m = R$ sind konstant und daher beschränkt. Zudem sind, siehe (17), die Potenzen der Matrix B gleichmäßig beschränkt. Daher existiert eine Konstante C_2 so dass

$$C_2 \geq \|B^j \text{Adiag}(J_{m-j-1,i})\| = \|B^j G_{m-j}\|$$

und

$$C_2 \geq \|R^j \text{Adiag}(J_{m-1,i})\| = \|B^j H_{m-j}\|$$

gilt.

Springen wir nun zu (23) zurück und setze unsere Abschätzungen ein. Dann ist

$$\begin{aligned} \|\epsilon_m\| &\leq \|B^m \epsilon_0\| + C_2 h \sum_{j=0}^{m-1} \|\epsilon_j\| + C_2 h \sum_{j=0}^{m-1} \|\epsilon_j\| + C_1 h^{s+1} \\ &= \|B^m \epsilon_0\| + 2C_2 h \sum_{j=0}^{m-1} \|\epsilon_j\| + C_2 h \|\epsilon_m\| + C_1 h^{s+1}. \end{aligned}$$

Laut Annahme des Satzes ist $\epsilon_0 = \mathcal{O}(h_0^{s+1}) = \mathcal{O}(h^{s+1})$ (aufgrund konstanter Schrittweite $h_m = h$). Außerdem ist unsere explizite Peer-Methode in dem Satz bereits laut Voraussetzung konvergent von Ordnung $p = s$. Es gilt also auch $\epsilon_m = \mathcal{O}(h^s)$. Zudem greifen wir auch $\|B\| = 1$ auf. Daraus folgt die Abschätzung

$$\begin{aligned} \|B^m \epsilon_0\| + C_2 h \|\epsilon_m\| + C_1 h^{s+1} &\leq \|B^m\| \|\epsilon_0\| + C_2 h C_3 h^s + C_1 h^{s+1} \\ &= \|\epsilon_0\| + (C_2 C_3 + C_1) h^{s+1} \\ &\leq C_4 h^{s+1} + (C_2 C_3 + C_1) h^{s+1} \\ &= (C_4 + C_2 C_3 + C_1) h^{s+1} \\ &= C_5 h^{s+1}. \end{aligned}$$

Dann gilt, siehe oben,

$$\|\epsilon_m\| \leq 2C_2 h \sum_{j=0}^{m-1} \|\epsilon_j\| + C_5 h^{s+1}. \quad (25)$$

Zeige nun per Induktion, dass daraus wiederum

$$\|\epsilon_m\| \leq (1 + 2C_2 h)^{m-1} C_5 h^{s+1} \quad (26)$$

folgt.

Den Induktionsanfang führen wir erfolgreich durch, indem wir $m = 1$ setzen -

sowohl in (25) als auch in (26) erhalten wir $\|\epsilon_m\| \leq C_5 h^{s+1}$. Die Induktionsannahme besteht somit darin, dass, für $k \in \{1, \dots, m-1\}$,

$$\|\epsilon_k\| \leq (1 + 2C_2 h)^{k-1} C_5 h^{s+1}$$

gilt.

Führe unter dieser Induktionsannahme den Induktionsschritt durch, indem wir k durch $k+1$ ersetzen - damit erhalten wir für $k \in \{1, \dots, m-1\}$

$$\|\epsilon_{k+1}\| \leq 2C_2 h \sum_{j=1}^k (1 + 2C_2 h)^j C_5 h^{s+1} + C_5 h^{s+1}$$

und damit

$$\begin{aligned} \|\epsilon_m\| &\leq 2C_2 h \sum_{j=1}^{m-1} (1 + 2C_2 h)^{j-1} C_5 h^{s+1} + C_5 h^{s+1} \\ &= (1 + 2C_2 h) \sum_{j=1}^{m-1} (1 + 2C_2 h)^{j-1} C_5 h^{s+1}. \end{aligned}$$

Mit der Partialsumme der geometrischen Reihe erhalten wir

$$\begin{aligned} \sum_{j=1}^{m-1} (1 + 2C_2 h)^{j-1} &= \sum_{j=0}^{m-2} (1 + 2C_2 h)^j \\ &= \frac{(1 + 2C_2 h)^{m-1} - 1}{(1 + 2C_2 h) - 1} \end{aligned}$$

und es folgt

$$\begin{aligned} \|\epsilon_m\| &\leq (1 + 2C_2 h) \frac{(1 + 2C_2 h)^{m-1} - 1}{(1 + 2C_2 h) - 1} C_5 h^{s+1} \\ &= (1 + (1 + 2C_2 h)^{m-1} - 1) C_5 h^{s+1} \\ &= (1 + 2C_2 h)^{m-1} C_5 h^{s+1} \\ &\leq e^{2C_2 h(m-1)} C_5 h^{s+1} \\ &\leq e^{2C_2(t_e - t_0)} C_5 h^{s+1}. \end{aligned}$$

Damit haben wir bewiesen, dass $\|\epsilon_m\| \leq O(h^{s+1})$ ist, wenn die Voraussetzungen des Satzes 3 erfüllt sind. \square

Satz 4:

Sei R eine konstante Matrix, $c_s = 1$ und $v = e_s$. Die Matrix B sei durch (20) gegeben und es gelte

$$\sum_{i=1}^{s-1} r_{si} c_i^{l-1} = \frac{1}{l}, \quad l = 2, \dots, s+1. \quad (27)$$

Dann ist, unter den Voraussetzungen von Satz 3, die explizite Peer-Methode superkonvergent von der Ordnung $p = s+1$.

Beweis:

Zunächst zeigen wir, dass, wenn $c_s = 1$, $v = e_s$ sowie (28) zutreffen, die Bedingung (22) erfüllt ist. Mit (22) lässt sich, siehe Satz 3, nämlich zeigen dass die explizite Peer-Methode superkonvergent ist.

Formuliere (22) um. Nutze die Bezeichnung $c^l = (c_1^l, \dots, c_s^l)$ sowie, für $l = s+1$, (12), und erhalte

$$e_s^T [c^{s+1} - \frac{1}{\sigma^{s+1}} B(c - \mathbb{1})^{s+1} - \frac{s+1}{\sigma^s} A(c - \mathbb{1})^s - (s+1)Rc^s] = 0.$$

Der Summand mit B entfällt, wie wir im Folgenden zeigen werden - nicht zuletzt gelten $e_s^T Q = e_s^T$ und $c_s = 1$. Es gilt

$$\begin{aligned} e_s^T \frac{1}{\sigma^{s+1}} B(c - \mathbb{1})^{s+1} &= \frac{1}{\sigma^{s+1}} e_s^T (\mathbb{1}v^T + QWQ^{-1})(c - \mathbb{1})^{s+1} \\ &= \left(\frac{1}{\sigma^{s+1}} e_s^T \mathbb{1}v^T + \frac{1}{\sigma^{s+1}} e_s^T QWQ^{-1} \right) (c - \mathbb{1})^{s+1} \\ &= \left(\frac{1}{\sigma^{s+1}} e_s^T + \frac{1}{\sigma^{s+1}} e_s^T WQ^{-1} \right) (c - \mathbb{1})^{s+1}. \end{aligned}$$

Schließlich ist e_s^T der s -te Einheitsvektor. $\mathbb{1}v^T$ ist, siehe Beweis von Lemma 1, eine Matrix, dessen s -ter Diagonaleintrag 1 ist, im Gegensatz zu den anderen Koeffizienten welche gleich 0 sind. $e_s^T \cdot \mathbb{1}v^T$ ergibt daher e_s^T .

Der s -te Eintrag von $(c - \mathbb{1})^{s+1}$ ist gleich 0, da $c_s - \mathbb{1}_s = 1 - 1 = 0$. Und da die ersten $s-1$ Einträge von e_s^T gleich 0 sind, ist $e_s^T \cdot (c - \mathbb{1})^{s+1}$ gleich 0. W ist, siehe Lemma 1, eine Matrix dessen letzte Zeile nur über Nulleinträge verfügt. Daher gilt $e_s^T W = 0$, mit der selben Begründung wie oben.

Womit gezeigt wäre, dass der Ausdruck $e_s^T \frac{1}{\sigma^{s+1}} B(c - \mathbb{1})^{s+1}$ entfällt. Es gilt, siehe oben,

$$e_s^T \frac{s+1}{\sigma^s} A(c - \mathbb{1})^s = e_s^T c^{s+1} + e_s^T (s+1)Rc^s.$$

Die Bezeichnungen aus (12) liefern

$$\begin{aligned}
e_s^T A &= [e_s^T c^{s+1} - e_s^T \frac{1}{\sigma^{s+1}} B(c - \mathbb{1})^{s+1} - e_s^T R c^s (s + 1)] S_m D^{-1} V_1^{-1} \\
&= [e_s^T C V_0 - e_s^T R V_0 D] V_1^{-1} D^{-1} S_m \\
&= e_s^T (C V_0 D^{-1} - R V_0) S_m V_1^{-1}.
\end{aligned}$$

Für $c_s = 1$, $v = e_s$ und $e_s^T A = e_s^T (C V_0 D^{-1} - R V_0) S_m V_1^{-1}$ lautet Bedingung (22) also

$$1 - (s + 1) e_s^T R c^s - \frac{s + 1}{\sigma^s} e_s^T (C V_0 D^{-1} - R V_0) S_m V_1^{-1} (c - \mathbb{1})^s = 0. \quad (28)$$

Dafür, dass (28) gilt, sind offensichtlich die Bedingungen

$$1 - (s + 1) e_s^T R c^s = 0 \quad (29)$$

und

$$e_s^T (C V_0 D^{-1} - R V_0) = 0 \quad (30)$$

hinreichend.

Aufgrund von $c_s = 1$ gilt $e_s^T C V_0 = \mathbb{1}^T$. Somit ist (30) äquivalent dazu, dass

$$\mathbb{1}^T D^{-1} - e_s^T R V_0 = 0$$

gilt.

Diese Bedingung wiederum lässt sich, für $l = 1, \dots, s + 1$, umformulieren, da $V_0 = (c_i^{j-1})_{i,j=1}^s$ gilt - es folgt

$$\mathbb{1}^T D^{-1} - e_s^T R V_0 = 0 \quad (31)$$

$$\mathbb{1}_l^T e_l \frac{1}{l} = e_s^T R V_0 \quad (32)$$

$$\frac{1}{l} = \sum_{i=1}^{s-1} r_{si} c_i^{l-1}, \quad \forall l = 1, \dots, s + 1. \quad (33)$$

Mit unseren Voraussetzungen gilt (33) also genau dann, wenn (30) gilt. Wenn wir nun zeigen könnten, dass aus (33) (29) folgt, wäre klar, dass (33) hinreichend dafür ist, dass (28) gilt. Schließlich sind (29) und (30) die beiden ursprünglichen hinreichenden Bedingungen dafür.

Zeige also, dass aus (33) $1 - (s + 1) e_s^T R c^s = 0$ folgt. Wir werden zeigen, dass

$\frac{1}{l} = \sum_{i=1}^{s-1} r_{si}c_i^{l-1}$ sogar nur für $l = s + 1$ gelten muss, damit (29) folgt.

Gelte also $\frac{1}{s+1} = \sum_{i=1}^{s-1} r_{si}c_i^s$. Dann gilt

$$\begin{aligned} 1 &= (s+1) \sum_{i=1}^{s-1} r_{si}c_i^s \\ 1 &= (s+1)e_s^T R c^s \\ 1 - (s+1)e_s^T R c^s &= 0. \end{aligned}$$

Dies zeigt, dass

$$\frac{1}{l} = \sum_{i=1}^{s-1} r_{si}c_i^{l-1}, \quad l = 1, \dots, s+1$$

eine hinreichende Bedingung dafür ist, dass (29), und damit (22) gilt.

Ursprünglich wollten wir jedoch beweisen, dass diese hinreichende Bedingung für $l = 2, \dots, s+1$ gilt - es bleibt also noch zu zeigen dass die Bedingung (33) für $l = 1$ überflüssig ist.

Das würde bedeuten, dass die Bedingungen $1 - (s+1)e_s^T R c^s = 0$ und $e_s^T (CV_0 D^{-1} - RV_0) = 0$ nicht zwingend für $l = 1$ gelten müssen. Das Ersteres stimmt, haben wir bereits gezeigt, dafür genügt nämlich dass es nur für $l = s+1$ gilt.

Und Letzteres ist der Fall, wenn die erste Komponente des Vektors $S_m V_1^{-1} (c - \mathbb{1})^s$ gleich 0 ist. Denn in (28) wird jeweils die erste Komponente von $(CV_0 D^{-1} - RV_0)$ mit der ersten Komponente von $S_m V_1^{-1} (c - \mathbb{1})^s$ multipliziert. Ist Letztere für $s = 1$ gleich 0, so muss für $s = 1$ natürlich nicht unbedingt $e_s^T (CV_0 D^{-1} - RV_0) = 0$ gelten.

Dass überhaupt

$$w_1 = e_1^T S_m V_1^{-1} (c - \mathbb{1})^s = 0$$

gilt, wollen wir also nun zeigen.

Bilde zunächst das Polynom

$$\Phi(t) = \sum_{i=1}^s (t - c_i + 1) = t^s + \sum_{k=0}^{s-1} \psi_k t^k.$$

Dann ist

$$\Phi(c_i - 1) = \prod_{i=1}^s (c_i - 1 - c_i + 1) = 0.$$

Daraus folgt ebenfalls

$$(c_i - 1) + \sum_{k=0}^{s-1} \psi_k (c_i - 1)^k = 0.$$

Bilde ein lineares Gleichungssystem und erhalte

$$(c - \mathbb{1})^s + (\mathbb{1}, c - \mathbb{1}, \dots, (c - \mathbb{1})^{s-1})\psi = 0.$$

In Abschnitt 4 haben wir die Matrix $V_1 = ((c_i - 1)^{j-1})_{i,j=1}^s$ definiert. Also notieren wir

$$(c - \mathbb{1})^s + V_1\psi = 0. \quad (34)$$

In unserem gesamten Beweis (nicht zu verwechseln mit dem bloßen Beweis von $w_1 = e_1^T S_m V_1^{-1} (c - \mathbb{1})^s = 0$) setzen wir wie gesagt voraus, dass $c_s = 1$ gilt. Dann ist

$$\Phi(c_s = 1) = \Phi(0) = 0^s + \psi_0 0^0 = \psi_0.$$

Gleichzeitig gilt auch

$$\Phi(c_i - 1) = \prod_{i=1}^{s-1} (c_s - c_i)(c_s - c_s) = 0$$

weswegen $\psi_0 = 0$ ist.

Um zu zeigen, dass die Bedingung (34) für $l = 1$ entbehrlich ist, nutzen wir $\psi_0 = 0$ um zu zeigen dass $w_1 = 0$ ist.

Aufgrund von (35) wissen wir dass $(c - \mathbb{1})^s + V_1\psi = 0$ gilt. Zudem ist der erste Eintrag der Diagonalmatrix S_m gleich 1, siehe Beweis von Satz 1. Daraus folgt

$$\begin{aligned} -(c - \mathbb{1})^s &= V_1\psi \\ -V_1^{-1}(c - \mathbb{1})^s &= \psi \\ -e_1^T V_1^{-1}(c - \mathbb{1})^s &= \psi_0 \\ -e_1^T S_m V_1^{-1}(c - \mathbb{1})^s &= \psi_0 \end{aligned}$$

und da wir bereits gezeigt haben dass $\psi_0 = 0$ ist haben wir

$$-e_1^T S_m V_1^{-1}(c - \mathbb{1})^s = -w_1 = 0$$

gezeigt. \square

Somit ist gezeigt, dass die Bedingung (34) nicht zwingend für $l = 1$ gelten

muss. Wenn (34) gilt, folgt daraus dass (22) beziehungsweise mit unseren Voraussetzungen (29) gilt. Und wenn (22) zutrifft, sowie die anderen Bedingungen aus Satz 3, dann ist die explizite Peer-Methode superkonvergent von der Ordnung $p = s + 1$.

Nun wissen wir, wie wir eine superkonvergente s -stufige explizite Peer-Methode konstruieren. Im folgenden, kurzen Abschnitt werden wir, nun aber der Reihenfolge nach, erläutern wie wir das superkonvergente Verfahren anwenden.

7 Konstruktionsprinzip

Wie gehen wir nun, für ein gegebenes Anfangswertproblem (1), vor, wenn wir die Lösung dessen mittels der expliziten Peer-Methode approximieren wollen? Bestimmte Koeffizienten können wir frei wählen, andere sind eindeutig bestimmt - und wiederum andere hängen selber von verschiedenen Koeffizienten ab. Folgende fünf Schritte führen wir durch:

1. Beliebig wählen dürfen wir im ersten Schritt die reguläre Matrix \tilde{Q} sowie die streng obere Dreiecksmatrix \tilde{W} (so dass beide Matrizen jeweils $s - 1$ Zeilen und Spalten haben). Wir müssen aber $v = e_s$ sowie $c_s = 1$ setzen. Die verbleibenden Koeffizienten des Vektors c benötigen wir noch nicht, wir bestimmen sie später. Genau genommen besteht der erste Schritt darin, dass wir B nach (20) bestimmen, mithilfe der soeben genannten Koeffizienten. Demzufolge muss also $\tilde{v} = 0$ sein.
2. Nun wählen wir die Koeffizienten c_1, \dots, c_{s-2} . Somit ist einzig die Komponente c_{s-1} des Vektors c unbekannt. Ebenfalls wählen wir eine streng untere Dreiecksmatrix R beliebig, mit jeweils s Zeilen und Spalten - aber die letzte Zeile lassen wir offen.
3. Die fehlenden Koeffizienten $c_{s-1}, r_{s1}, \dots, r_{s,s-1}$ berechnen wir mit der Gleichung in (33). Es genügt aber, dass diese Gleichung für $l = 2, \dots, s + 1$ gilt. Wir haben also sämtliche b_{ij} und r_{ij} beisammen, es fehlen nur die Koeffizienten von A_m .
4. Mit der Gleichung in (14) können wir A_m bestimmen - dank der bereits gefundenen B_m und R_m .

Somit wissen wir, wie wir die Koeffizienten a_{ij}, b_{ij} und r_{ij} ermitteln. Wir können sie also in (2) einsetzen. Dann können wir ein beliebiges Anfangswertproblem (1) für ein System von Differentialgleichungen erster Ordnung approximieren, und zwar derart dass die explizite Peer-Methode superkonvergent ist - eben

weil wir die Koeffizienten a_{ij} , b_{ij} und r_{ij} nach diesem Kriterium bestimmt haben.

8 Numerische Versuche und Fazit

In dem anliegenden Python-Code, siehe Anhang 1, habe ich das explizite s -stufige Peer-Verfahren implementiert und darüber hinaus Beispiele für steife beziehungsweise nichtsteife Systeme gezeigt. Dazu habe ich das Modellproblem für lineare Stabilität

$$\begin{aligned}y' &= \lambda y, \\ y(0) &= 1\end{aligned}$$

verwendet, für $\lambda < 0$ und auf dem Intervall $[0, 1]$. Ist λ sehr klein, so ist das Problem steif, ansonsten nichtsteif.

Ich habe, für jeweils unterschiedliche λ , das System viermal implementiert und die Ergebnisse jeweils in Plots abbilden lassen, siehe Abbildungen 1 - 4. In den vier Plots ist jeweils ein Graph eingezeichnet, welcher die eigentliche Lösung des Anfangswertproblems (also von dem Modellproblem für lineare Stabilität) repräsentiert. Zusätzlich enthalten die vier Plots noch jeweils drei andere Graphen. Drei Graphen sind es deshalb, weil ich die dreistufige explizite Peer-Methode in meinen Beispielen genutzt habe. Schließlich werden in einem s -stufigen Verfahren ja s Näherungswerte berechnet, welche zueinander gleichberechtigt sind.

Angemerkt sei, dass ich, der Einfachheit halber, jeweils gleiche Schrittweiten verwendet habe, mit $h_m = \frac{1}{3}$. Das heißt, dass A_m auch fest ist. Zu zeigen ist allerdings noch, wie die tatsächliche Lösung des Anfangswertproblems zustande kommt. Berechnet wird sie durch Trennung der Variablen. Wir erhalten

$$\begin{aligned}y' &= \lambda y \\ \frac{dy}{dx} &= \lambda y \\ dy &= \lambda y dx \\ \frac{dy}{y} &= \lambda dx\end{aligned}$$

und wenn wir auf beiden Seiten integrieren, folgt

$$\int \frac{1}{y} dy = \int \lambda dx.$$

Integration liefert

$$\begin{aligned} \ln(y) &= C + \lambda x \\ y &= e^{C+\lambda x}. \end{aligned}$$

Und weil $y(0) = 1$ gilt, folgt daraus

$$\begin{aligned} y(0) &= e^C. \\ c &= 0 \\ y &= e^{\lambda x}. \end{aligned}$$

Damit wäre y berechnet.

Durch Ausprobieren habe ich nach geeigneten Werten λ gesucht, um gute Beispiele für steife beziehungsweise nichtsteife Systeme vorweisen zu können. In dem Code aus Anhang 1 befinden sich wie gesagt, neben der grundsätzlichen Implementierung des Peer-Verfahrens, vier Beispiele für die Approximation der Lösung des Modellproblems für lineare Stabilität - für jeweils $\lambda = -5$ (Abbildung 1), $\lambda = -50$ (Abbildung 2), $\lambda = -0,01$ (Abbildung 3) und $\lambda = -0,00001$ (Abbildung 4).

Nimmt λ einer der beiden erstgenannten Werte an, so handelt es sich um ein steifes, ansonsten um ein nichtsteifes System. Es sei darauf hingewiesen, dass der Startwert bei den steifen Systemen selbstverständlich auch $y = 1$ beträgt. Warum es auf den Grafiken zu den steifen Systemen nach $y = 0$ aussieht, liegt an der Skalierung der vertikalen Achse.

Unschwer erkennbar ist, dass bei den nichtsteifen Systemen die tatsächlichen Werte von y deutlich präziser approximiert werden als bei den steifen Systemen. Die Abweichung von der tatsächlichen Lösung ist bei den nichtsteifen Systemen (siehe Abbildungen 3 und 4) jeweils deutlich geringer als bei den steifen Systemen (siehe Abbildungen 1 und 2). Dies deckt sich mit den Erkenntnissen von Podhaisky et al. [2].

Beachten muss man hierbei, dass in Beispiel 2, neben der vertikalen Achse, $1e11$ vermerkt ist. Der Fehler ist in Beispiel 2 (ein Beispiel, welches sich auf ein steifes System bezieht) also nochmal deutlich größer, als auf dem ersten Blick sichtbar. Der Befund bestätigt die aufgestellte Hypothese, dass das explizite s -stufige Peer-Verfahren für nichtsteife Systeme deutlich besser geeignet ist als für steife Systeme.

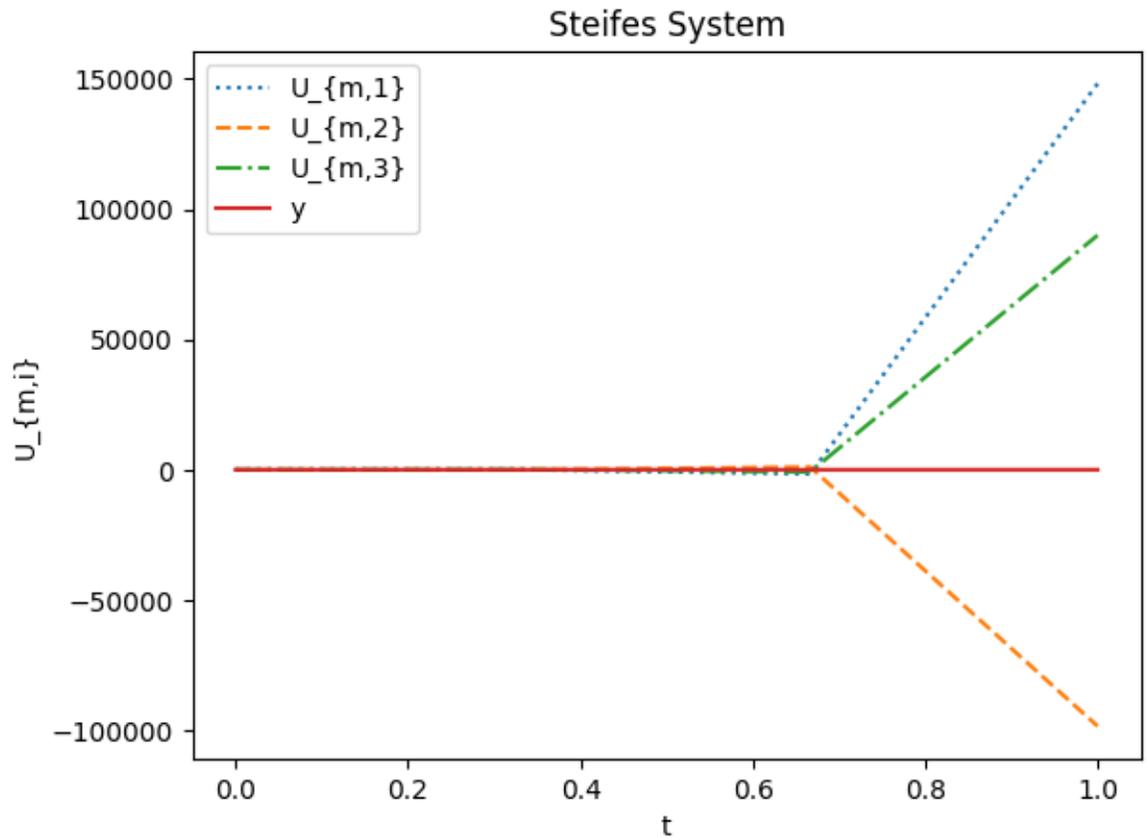


Abbildung 1: Steifes System

Was die nichtsteifen Systeme angeht, muss jedoch einschränkend festgehalten werden, dass die Werte $U_{m,1}$ relativ ungenau sind. Dies ist insbesondere verglichen mit den Werten $U_{m,2}$ und $U_{m,3}$, denen eine gelungene Approximation gelingt, der Fall. Ein naheliegender Grund ist, dass bei der Berechnung der $U_{m,1}$, im Gegensatz zu anderen i , die Summe mit r_{ij} entfällt. Schließlich wird die Summe nur bis $i - 1$ gebildet. Womöglich ist das Ergebnis von $U_{m,1}$ ohne die Summanden mit den Koeffizienten von R_m zu unpräzise. Das soll hier aber nicht Thema sein.

Festhalten lässt sich grundsätzlich, dass die s -stufige explizite Peer-Methode bei nichtsteifen Systemen zuverlässige Ergebnisse liefert - zumindest für $U_{m,i}$ mit $i > 1$. Die unbefriedigenden Ergebnisse bei den steifen Systemen motivieren uns, im nächsten Abschnitt einen Ausblick auf linear-implizite Peer-Methoden

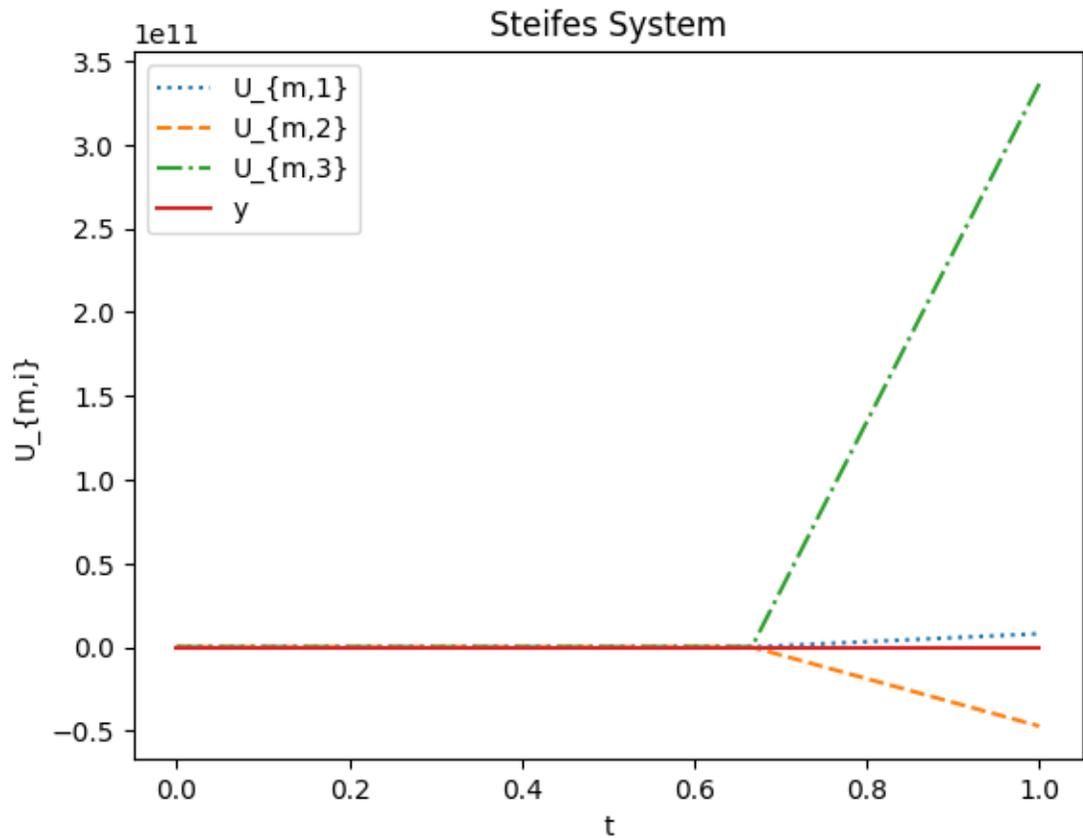


Abbildung 2: Steifes System

zu geben. Diese sind, siehe Podhaisky et al. [2], für steife Verfahren besser geeignet.

9 Ausblick - linear-implizite Peer-Methoden

Betrachte nun, wie anfangs angekündigt, die linear-implizite s -stufige Peer-Methode. Sie ist, laut Podhaisky et al. [2], für das Approximieren steifer Systeme sehr gut geeignet. Dies ist insofern interessant, weil das explizite Verfahren in unseren numerischen Versuchen Schwächen in dem Bereich offenbarte (siehe vorheriges Kapitel).

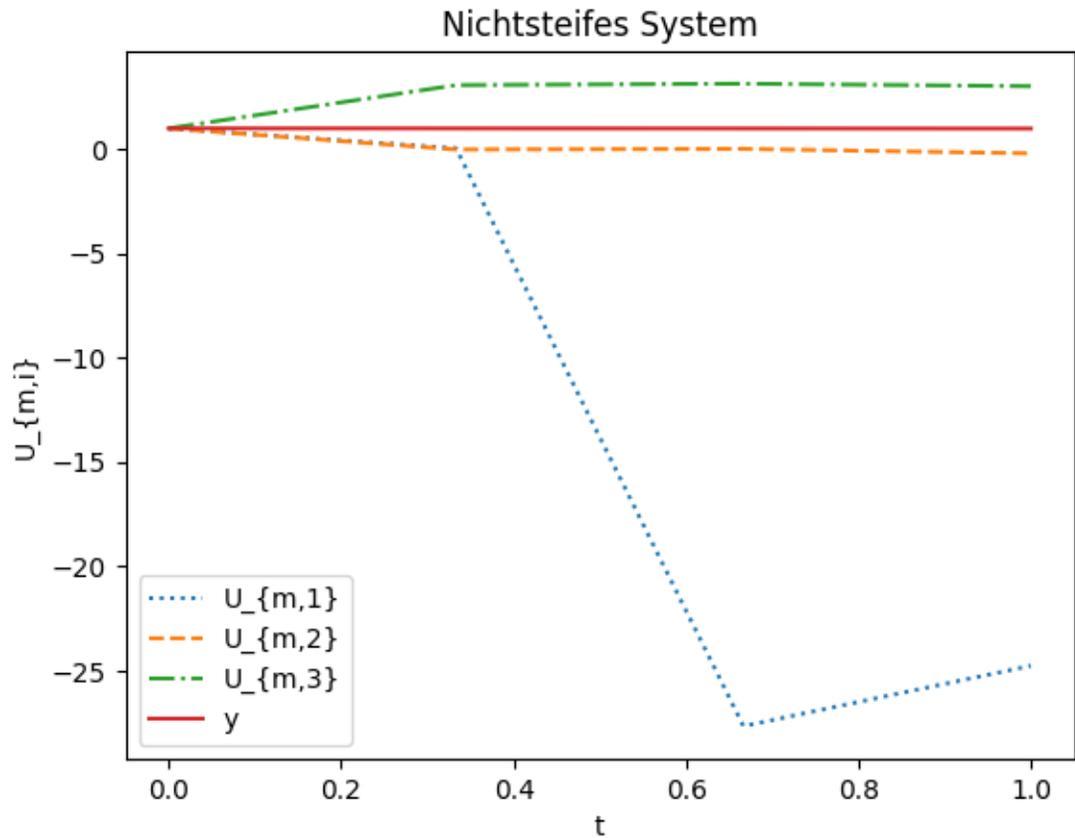


Abbildung 3: Nichtsteifes System

Betrachte also (1), das Anfangswertproblem für Differentialgleichungen erster Ordnung. Um dies zu lösen, definieren wir nun die s -stufige linear-implizite Peer-Methode.

Analog zum expliziten Verfahren berechnen wir s Näherungen $Y_{m,i} \approx y(t_{mi})$, für $y(t_{mi}) = y(t_m + c_i h_m)$. Auch hier stellen die h_m die Schrittweiten sowie die c_i die paarweise verschiedenen Knoten dar.

Zudem setzen wir, um uns kürzer zu fassen, $F_{m,j} = f(t_m + c_j h_m, Y_{m,i})$. Beliebig gewählt werden kann die Matrix T_m - wir bestimmen T_m als eine Approximation an eine Jacobi-Matrix.

Die Definition der s -stufigen linear-impliziten Peer-Methode lautet

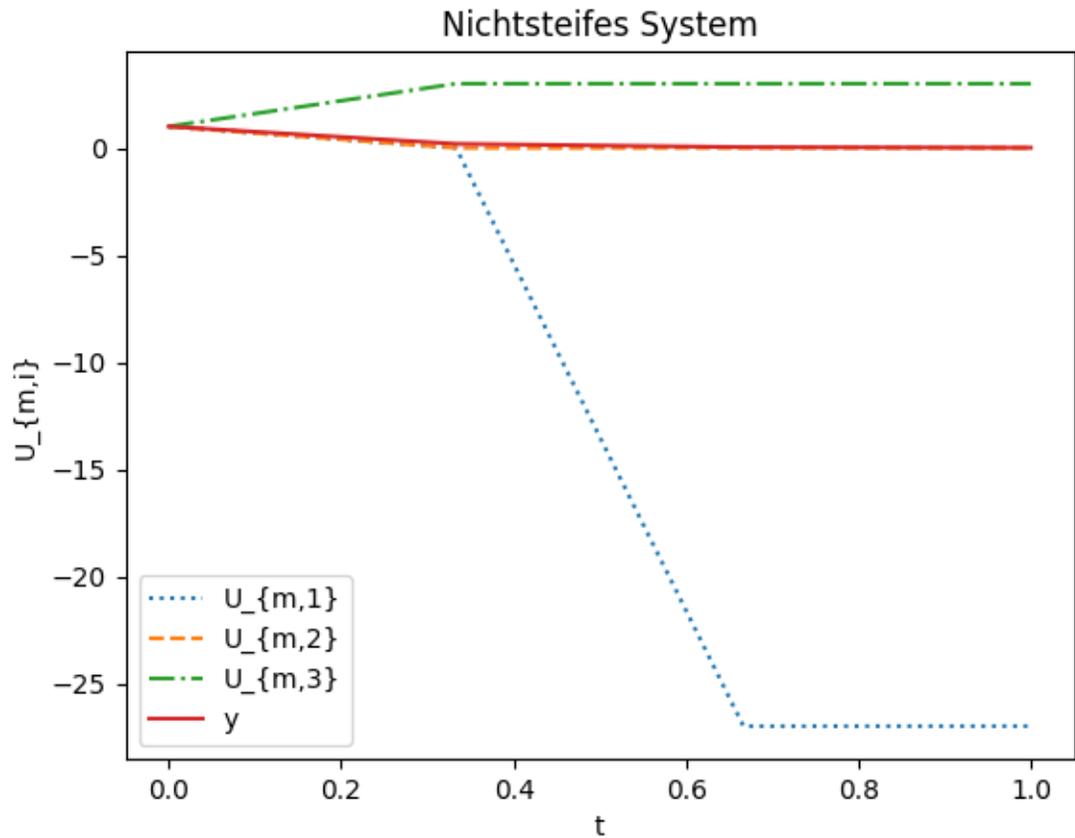


Abbildung 4: Nichtsteifes System

$$(I - h_m \gamma T_m) Y_{m,i} = \sum_{j=1}^s b_{ij} Y_{m-1,j} + h_m \sum_{j=1}^s a_{ij} (F_{m-1,j} - T_m Y_{m-1,j}) + h_m T_m \sum_{j=1}^{i-1} g_{ij} Y_{m,j},$$

$$i = 1, \dots, s.$$

Auch die Definition des impliziten Verfahrens wollen wir kompakter formulieren. Notiere dazu die s Komponenten c_i als einen einzelnen Vektor c , sowie die Koeffizienten a_{ij} , b_{ij} und g_{ij} als (s, s) -Matrizen A_m , B_m und G_0 . Hierbei ist G_0 eine streng untere Dreiecksmatrix, für die $G = G_0 + \gamma I$ gilt, für $\gamma > 0$. Ebenso wie bei der expliziten Methode können die Koeffizienten abhängig von dem Verhältnis der Schrittweiten $\sigma_m = \frac{h_m}{h_{m-1}}$ sein.

Wir betrachten Fälle, in denen $G_0 = 0$ gilt. Dann ist die untere Dreiecksmatrix G konstant wegen $G = G_0 + \gamma I$. Zudem beschäftigen wir uns, wie wir es auch bei dem expliziten Verfahren getan haben, nur mit autonomen Differentialgleichungen.

Dann ist die s -stufige linear-implizite Peer-Methode, weil die Koeffizienten g_{ij} entfallen und weil wir γT_m durch GT_m ersetzen, gegeben durch

$$(I - h_m GT_m)Y_m = B_m Y_{m-1} + h_m A_m (F_{m-1} - T_m Y_{m-1}).$$

Bemerkung 3:

Sei $G_0 = 0$. Dann können wir die s Stufenwerte parallel berechnen, da sie von den jeweils vorausgegangenen Stufenwerten unabhängig sind - siehe den Fall $R_m = 0$ beim expliziten Peer-Verfahren.

Kommen wir nun zu den Konsistenzaussagen.

Und diese treffen zu können, müssen wir, analog zum expliziten Verfahren, das Residuum bilden durch

$$\Delta = Y(t_m) - h_m GT_m Y(t_m) - B_m Y(t_{m-1}) - h_m A_m (Y'(t_{m-1}) - T_m Y(t_{m-1}))$$

so dass

$$Y(t_m) = \begin{pmatrix} y(t_m + c_1 h_m) \\ \dots \\ y(t_m + c_s h_m) \end{pmatrix}$$

und

$$Y(t_{m-1}) = \begin{pmatrix} y(t_{m-1} + c_1 h_{m-1}) \\ \dots \\ y(t_{m-1} + c_s h_{m-1}) \end{pmatrix}.$$

Definition 3:

Die Peer-Methode verfügt über die Konsistenzordnung p genau dann, wenn

$$\Delta_m := O(h_m^{p+1})$$

für

$$h_m \rightarrow 0$$

gilt.

Das Residuum formulieren wir mittels einer Taylorentwicklung um, wobei wir genau so vorgehen wie in (9) - wieder nutzen wir die Taylorentwicklungen (6) und (8). Wir erhalten

$$U_{m,i} = \sum_{j=1}^s b_{ij} U_{m-1,j} + h_m \sum_{j=1}^s a_{ij} f(t_{m-1,j}, U_{m-1,j}) + h_m \sum_{j=1}^{i-1} r_{ij} f(t_{m,j}, U_{m,j}), i = 1, \dots, s$$

und

$$\begin{aligned}\Delta_m &= \sum_{l=0}^p \left(c^l - \frac{1}{\sigma_m^l} B_m (c - \mathbb{1})^l - \frac{1}{\sigma_m^{l-1}} A_m (c - \mathbb{1})^{l-1} \right) \frac{h_m^l}{l!} y^{(l)}(t_m) \\ &\quad - \sum_{l=0}^{p-1} \left(Gc^l - \frac{1}{\sigma_m^l} A_m (c - \mathbb{1})^l \right) \frac{h_m^{l+1}}{l!} T_m y^{(l)}(t_m) + O(h_m^{p+1}).\end{aligned}$$

Wir setzen nun die Koeffizienten $\frac{h_m^l}{l!} y^{(l)}(t_m)$ und $\frac{h_m^{l+1}}{l!} T_m y^{(l)}(t_m)$ gleich 0. Dies machen wir, um Ordnungsbedingungen dafür zu erfahren, wie wir die Koeffizienten a_{ij} , b_{ij} etc. wählen. Nutze dabei die Bezeichnungen

$$AB(l) := c^l - \frac{1}{\sigma_m^l} B_m (c - \mathbb{1})^l - \frac{1}{\sigma_m^{l-1}} A_m (c - \mathbb{1})^{l-1} \quad (35)$$

und

$$G(l) := Gc^l - \frac{1}{\sigma_m^l} A_m (c - \mathbb{1})^l. \quad (36)$$

Satz 5:

Gelte $AB(l) = 0$ für $l = 0, \dots, p$ sowie $G(l) = 0$ für $l = 0, \dots, p - 1$. Dann verfügt die linear-implizite s -stufige Peer-Methode, für beliebige Matrizen T_m , über die Konsistenzordnung $p = s - 1$.

Beweis:

Setze $AB(l) = c^l - \frac{1}{\sigma_m^l} B_m (c - \mathbb{1})^l - \frac{1}{\sigma_m^{l-1}} A_m (c - \mathbb{1})^{l-1} = 0$ und $G(l) = Gc^l - \frac{1}{\sigma_m^l} A_m (c - \mathbb{1})^l = 0$ in das Residuum ein - dann trifft Definition 3 zu. \square

Wie beim expliziten Verfahren wollen wir A_m , B_m , G und c derart wählen, dass die linear-implizite s -stufige Peer-Methode über die Konsistenzordnung p verfügt. Dies würde durch die Bedingungen aus Satz 5 garantiert werden. Deshalb stellen wir wiederum Bedingungen an A_m und B_m auf, und zwar in Form von Matrixgleichungen. Dort lassen wir eben die Forderungen aus Satz 5 mit einfließen.

Einerseits soll $AB(l) = 0$ für $l = 0, \dots, s - 1$ gelten. Andererseits muss $G(l) = 0$ sein - jedoch nicht nur, wie es eigentlich genügen würde, für $l = 0, \dots, s - 2$, sondern zusätzlich auch noch für $l = s - 1$. Dies dient der Vereinfachung unserer Darstellung - denn schließlich gilt das Kriterium für $AB(l)$ auch für $l = s - 1$. Und in unseren Matrixgleichungen entspricht ein einzelnes l einer Spalte.

Im Beweis von Satz 1 hatten wir die Bezeichnung $S_m = \text{diag}(1, \sigma_m, \dots, \sigma_m^{s-1})$ eingeführt. Damit folgt aus $AB(l) = 0$ für $l = 0, \dots, s-1$, siehe (31),

$$\begin{aligned} (\mathbb{1}, c, \dots, c^{s-1}) &= B_m(\mathbb{1}, c - \mathbb{1}, \dots, (c - \mathbb{1})^{s-1})S_m^{-1} \\ &+ A_m(0, \mathbb{1}, c - \mathbb{1}, \dots, (c - \mathbb{1})^{s-2})\text{diag}(0, 1, \frac{2}{\sigma_m}, \dots, \frac{s-1}{\sigma_m^{s-2}}). \end{aligned} \quad (37)$$

Analog liefert, siehe (37), $G(l) = 0$ für $l = 0, \dots, s-1$

$$G(\mathbb{1}, c, \dots, c^{s-1}) = A_m(\mathbb{1}, c - \mathbb{1}, \dots, (c - \mathbb{1})^{s-1})S_m^{-1}.$$

Wir wollen nun ermitteln, wie wir A_m und B_m in Abhängigkeit von G und c bestimmen. Dazu verwenden wir ebenfalls die Bezeichnungen $V_0 = (c_i^{j-1})_{i,j=1}^s$ und $V_1 = ((c_i - 1)^{j-1})_{i,j=1}^s$, die wir im Beweis von Satz 1 eingeführt hatten. Dazu definieren wir die Matrix

$$F_0 = [e_2, e_3, \dots, e_s, 0] = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

Es gilt

$$\begin{aligned} V_0 &= B_m V_1 S_m^{-1} + A_m V_1 S_m^{-1} D F_0^T \\ B_m V_1 S_m^{-1} &= V_0 - A_m V_1 S_m^{-1} D F_0^T \\ B_m &= (V_0 - A_m V_1 S_m^{-1} D F_0^T) S_m V_1^{-1}, \end{aligned} \quad (38)$$

denn dank der Gestalt von F_0 wurde mit F_0^T - bis auf die erste Spalte die 0 ist - der untere Teil der Matrix $S_m^{-1} D F_0^T$ durch den streng unteren Teil von S_m^{-1} dargestellt.

Die andere Matrixgleichung lautet

$$\begin{aligned} G V_0 &= A_m V_1 S_m^{-1} \\ A_m &= G V_0 S_m V_1^{-1}. \end{aligned} \quad (39)$$

Also haben wir unser Ziel erreicht, dass wir A_m und B_m - und zwar einzig in Abhängigkeit der Koeffizienten G und c - bestimmen können. Die G und c

wollen wir wiederum derart bestimmen, dass das linear-implizite Verfahren $L(\alpha)$ -stabil ist, mit möglichst großem Winkel α .

Wir möchten die Stabilitätsfunktion des linear-impliziten Peer-Verfahrens berechnen. Dazu benötigen wir zunächst die Dahlqvist'sche Testgleichung der A-Stabilität

$$y' = \lambda(y). \quad (40)$$

Dazu nehmen wir $T_m = \lambda$ sowie $z = h_m \lambda$ an. Dann gilt

$$\begin{aligned} (I - h_m G T_m) Y_m &= B_m Y_{m-1} + h_m A_m (F_{m-1} - T_m Y_{m-1}) \\ (I - h_m \lambda G) Y_m &= B_m Y_{m-1} + h_m A_m (\lambda Y_{m-1} - \lambda Y_{m-1}) \\ (I - h_m \lambda G) Y_m &= B_m Y_{m-1} \\ Y_m &= (I - zG)^{-1} B_m Y_{m-1}. \end{aligned}$$

Also erhalten wir mit der Stabilitätsfunktion $M(z)$

$$Y_m = M(z) Y_{m-1}, \quad M(z) = (I - zG)^{-1} B_m. \quad (41)$$

Jedoch muss $M(z)$ von der Gestalt her offensichtlich eine Matrix sein, deshalb bezeichnen wir sie im Folgenden auch als Stabilitätsmatrix.

Ebenso wie bei der expliziten Methode wollen wir auch, dass das linear-implizite Verfahren nullstabil ist. Schließlich hat das Verfahren, wenn es die Konsistenzordnung p hat und nullstabil ist, zwingend die Konvergenzordnung p .

Mittels der Stabilitätsmatrix lässt sich zeigen, dass das linear-implizite Verfahren über ein größeres Stabilitätsgebiet verfügt als die explizite Methode. Somit ist diese für steife Systeme besser geeignet.

So weit der Ausblick auf die linear-implizite Peer-Methode.

Literatur

- [1] Helmut Podhaisky, Karl Strehmel, Rüdiger Weiner (1995), Numerik gewöhnlicher Differentialgleichungen, Springer, Halle a. d. S., 2. Auflage
- [2] Helmut Podhaisky, Bernhard Schmitt, Rüdiger Weiner (2005), Multi-Implicit Peer Two-Step W-Methods For Parallel Time Integration, BIT Numerical Mathematics, Ausgabe 45 (2005), Seiten 197-217, Springer
- [3] Roland Bulirsch, Josef Stoer (1990), Numerische Mathematik 2, Springer, München, Würzburg, 3. Auflage
- [4] Sören Bartels (2016), Numerik 3x9, Springer, Freiburg

10 Anhang

Anhang 1:

```
import numpy as np
import random
import matplotlib.pyplot as plt
from math import e

#Zunächst  $s$  wählen, je nach dem welche Stufe unser  $s$ -stufiges
#explizites Peer-Verfahren haben soll.

s = 4

v = np.array([0] * (s-1) + [1])
#Der Vektor  $v = e_{s}^T$ . In Abhängigkeit von unserem gewählten  $s$ 
def vvektor(s):
    v = np.array([0] * (s-1) + [1])
    return "v"

v = np.array([0] * (s-1) + [1])

#Der Eins-Vektor  $\mathbb{1}$  mit  $s$  Komponenten
n1 = np.array([1] * s)

#Definiere die streng obere Dreiecksmatrix  $\hat{W}$ .
W_hat = np.array([[random.randint(1,10)]*(s-1)]*(s-1))
#Anzahl der Zeilen bzw Spalten
rows = len(W_hat);
cols = len(W_hat[0]);

if(rows != cols):
    print("Matrix muss quadratisch sein.");
else:
    #For-Schleife um Matrix in streng obere Dreiecksmatrix umzuwandeln
    print("Streng obere Dreiecksmatrix: ");
    for i in range(0, rows):
        for j in range(0, cols):
            if(j < i+1):
                W_hat[i][j] = 0
                print("0"),
            else:
                print(W_hat[i][j]),
```

```

print(" ");

# Wähle, der Einfachheit halber und da  $\hat{Q}$  sowieso
# regulär sein soll, die Matrix  $\hat{Q}$  als Einheitsmatrix.
Q_hat = np.identity(s-1)

# Die  $s$ -te Zeile von  $W$ , bis auf den  $s$ -ten Eintrag:
zerosarray = np.array([0] * (s-1))

# Die  $s$ -te Spalte von  $W$ :
n2 = np.array([1] * (s-1))
w2 = np.matmul(-W_hat, n2)
w3 = np.append(w2, 0)

W1 = np.vstack([W_hat, zerosarray])

# Die finale Matrix  $W$ :
W = np.c_[ W1, w3 ]

#Eigentlich ist  $B = \mathbb{1}v^T + QWQ^{-1}$ .
#Da aber  $\hat{Q}$  als Einheitsmatrix wurde und, siehe (20) aus dem Manuskript,
#  $\tilde{v} = 0$  ist, ist  $Q$  ebenfalls die Einheitsmatrix.
#Daraus folgt  $B = \mathbb{1}v^T + W$ .
B = np.outer(v, n1) + W

print(B)

#Den Vektor  $\sigma_m$  in Abhängigkeit der
#Schrittweiten  $h_{m-1}$  und  $h_m$  bestimmen:
def sigma(hm, hmminus1):
    sigmam = hm/hmminus1
    return sigmam

hm = 0.2
hmminus1 = 0.2

```

```

sigmam = hm/hmminus1
sigmaarraym = np.array([sigmam,]*(s))
sigmaarrayshortm = np.array([1] + [sigmam,]*(s-1))

#Die Diagonalmatrizen  $S_{\{m\}}$  und  $\hat{S}_{\{m\}}$  definieren.
#Hierfür benötigen wir  $\sigma_{\{m\}}$ .
Sm = np.diag([x ** i for i, x in enumerate(sigmaarrayshortm)])
print(Sm)

Sm_hat = np.diag([x ** i for i, x in enumerate(sigmaarraym)])
print(Sm_hat)

#Gleichung (27) aus dem Manuskript
s = 4
equation2 = True

#Definiere die streng untere Dreiecksmatrix  $R$ :
def equation1():
    R = np.random.rand(s,s)

    #Zeilen- und Spaltenanzahl berechnen
    rows = len(R);
    cols = len(R[0]);

    if(rows != cols):
        print("Matrix muss quadratisch sein.");
    else:
        #For-Schleife um Matrix in streng untere Dreiecksmatrix umzuwandeln
        print("Streng untere Dreiecksmatrix: ");
        for i in range(0, rows):
            for j in range(0, cols):
                if(j+1 > i):
                    R[i][j] = 0
                    print("0"),
                else:
                    print(R[i][j]),

        print(" ");

cvectorlist = random.sample(range(2,20),s-1)

```

```

cvector = np.array(cvectorlist)

return R,cvector

print(R)
print(cvector)

while equation2:
    R,cvector = equation1()
    for l in range(2,s+1):
        if sum ([R[-1:,i]*cvector[i]**l != 1/(l+1) for i in range(1,s-1)]):
            equation2 = True
            break
    equation2 = False
print(cvector[2])
print(R[-1:,0])
print(R[-1:,1])

#Noch einmal die Matrix $R$ sowie die ersten $s-1$ Komponenten des Vektors $c$
print(cvector)
print(R)

#$c_{s} = 1$
c = np.concatenate((cvector, [1]), axis=None)

#Die Matrizen $C$, $C - I$, $V_{0}$, $V_{1}$ sowie seine Inverse definieren.
#Hierfür benötigen wir den Vektor $c$.
C = np.identity(s)*c
cminus1 = c - np.array([1]*s)
Cminus1 = np.identity(s)*cminus1

V_0 = np.vander(c, increasing=True)
V_1 = np.vander(cminus1, increasing=True)
invV_1 = np.linalg.inv(V_1)

#Die Diagonalmatrizen $D$, seine Inverse sowie $\hat{D}$ definieren.

```

```

u = np.arange(s+1)
z = u[1:s+1]

D_hat = np.identity(s+1)*u
D = np.identity(s)*z

invD = np.linalg.inv(D)

#Im Folgenden die Ausdrücke die bei der Definition von  $A_m$  miteinander
#multipliziert werden müssen.
#Soll ich es anders darstellen? Sehen diese Multiplikationen
#übersichtlich genug aus?

#Ausdrücke aus dem ersten Summanden von  $A_m$ 
CV_0 = np.dot(C,V_0)
invDSm = np.dot(invD,Sm)
CV_0invDSm = np.dot(CV_0,invDSm)
CV_0invDSminV_1 = np.dot(CV_0invDSm,invV_1)
SminV_1 = np.dot(Sm,invV_1)

#Ausdrücke aus dem zweiten Summanden von  $A_m$ 
RV_0 = np.dot(R,V_0)
SminV_1 = np.dot(Sm,invV_1)
RV_0SminV_1 = np.dot(RV_0,SminV_1)

#Ausdrücke aus dem dritten Summanden von  $A_m$ 
BCminus1 = np.dot(B,Cminus1)
V_1invD = np.dot(V_1,invD)
BCminus1V_1invD = np.dot(BCminus1,V_1invD)
BCminus1V_1invDinvV_1 = np.dot(BCminus1V_1invD,invV_1)

#Die Matrix  $A_m$  definieren, mit den oben stehenden Ausdrücken
Am = CV_0invDSminV_1 - RV_0SminV_1 - (1/sigmam)* BCminus1V_1invDinvV_1

#Beachte im Folgenden aber, dass  $A_m$  fest ist, da ich die Schrittweite
# $h_m$  äquidistant wähle.

#Für die Funktionsargumente möchte ich im Folgenden Ausdrücke
#für die numerischen Versuche einsetzen.

```

```

# Sei f_tmminus1Umminus1[j] im Code $f(tm1,j, Um1,j)$
# und f_tmUm[j] im Code $f(tm,j, Um,j)$
# und $hm = \frac{1}{3}$.
# Die grundsätzliche Definition der expliziten s-stufigen Peer-Methode
# lautet dann:

def exppeer(i, U_mminus1, f_tmminus1Umminus1, f_tmU_m):
    U_mi = (sum([B[i][j]*U_mminus1[j] for j in range(1,s) ])
            + sum([Am[i][j]* f_tmminus1Umminus1[j] for j in range(1,s) ]) / 3
            + sum([R[i][j]* f_tmU_m[j] for j in range(1,i-1) ]) / 3)

    return U_mi

# Grundsätzlich muss in obenstehender Definition i, U_mminus1,
# f_tmminus1Umminus1 und f_tmU_m noch bestimmt werden.

# Beginne nun unser numerisches Experiment.
# Betrachte dabei das Modellproblem für lineare Stabilität
# $y' = \lambda y$, $y(0) = 1$, im Intervall $[0,1]$, $\lambda < 0$.
# Wende darauf unsere Definition vom Peer-Verfahren an.
# Ist $\lambda$ sehr klein, so ist das System steif.
# Experimentiere nun jeweils mit einem $\lambda$ das sehr klein ist, und mit
# einem $\lambda$ das es nicht ist.

# Beginne mit $\lambda = - 5$.

U_01 = 1
U_02 = 1
U_03 = 1

f_t_01U_01 = -5
f_t_02U_02 = -5
f_t_03U_03 = -5

U_11 = B[1][1]*U_01 + B[1][2]*U_02 + B[1][3]*U_03 + ((Am[1][1]* f_t_01U_01
+ Am[1][2]* f_t_02U_02
+ Am[1][3]* f_t_03U_03) / 3)

```

$$f_t_11U_11 = -5 * U_11$$

$$U_12 = B[2][1]*U_01 + B[2][2]*U_02 + B[2][3]*U_03 + ((Am[2][1]* f_t_01U_01 + Am[2][2]* f_t_02U_02 + Am[2][3]* f_t_03U_03) / 3 + (R[2][1]* f_t_11U_11) / 3)$$

$$f_t_12U_12 = -5 * U_12$$

$$U_13 = B[3][1]*U_01 + B[3][2]*U_02 + B[3][3]*U_03 + ((Am[3][1]* f_t_01U_01 + Am[3][2]* f_t_02U_02 + Am[3][3]* f_t_03U_03) / 3 + R[3][1]* f_t_11U_11 / 3 + R[3][2]* f_t_12U_12 / 3)$$

$$f_t_13U_13 = -5 * U_13$$

$$U_21 = B[1][1]*U_11 + B[1][2]*U_12 + B[1][3]*U_13 + ((Am[1][1]* f_t_11U_11 + Am[1][2]* f_t_12U_12 + Am[1][3]* f_t_13U_13) / 3)$$

$$f_t_21U_21 = -5 * U_21$$

$$U_22 = B[2][1]*U_11 + B[2][2]*U_12 + B[2][3]*U_13 + ((Am[2][1]* f_t_11U_11 + Am[2][2]* f_t_12U_12 + Am[2][3]* f_t_13U_13) / 3 + (R[2][1]* f_t_21U_21) / 3)$$

$$f_t_22U_22 = -5 * U_22$$

$$U_23 = B[3][1]*U_11 + B[3][2]*U_12 + B[3][3]*U_13 + ((Am[3][1]* f_t_11U_11 + Am[3][2]* f_t_12U_12 + Am[3][3]* f_t_13U_13) / 3 + R[3][1]* f_t_21U_21 / 3 + R[3][2]* f_t_22U_22 / 3)$$

$$f_t_23U_23 = -5 * U_23$$

$$U_31 = B[1][1]*U_21 + B[1][2]*U_22 + B[1][3]*U_23 + ((Am[1][1]* f_t_21U_21$$

```
+ Am[1][2]* f_t_22U_22
+ Am[1][3]* f_t_23U_23) / 3)
```

```
f_t_31U_31 = -5 * U_31
```

```
U_32 = B[2][1]*U_21 + B[2][2]*U_22 + B[2][3]*U_23 + ((Am[2][1]* f_t_21U_21
+ Am[2][2]* f_t_22U_22
+ Am[2][3]* f_t_23U_23) / 3
+ (R[2][1]* f_t_31U_31) / 3)
```

```
f_t_32U_32 = -5 * U_32
```

```
U_33 = B[3][1]*U_01 + B[3][2]*U_02 + B[3][3]*U_03 + ((Am[3][1]* f_t_21U_21
+ Am[3][2]* f_t_22U_22
+ Am[3][3]* f_t_23U_23) / 3
+ R[3][1]* f_t_31U_31 / 3
+ R[3][2]* f_t_32U_32 / 3)
```

```
x1 = [0, 1/3, 2/3, 1]
y1 = [U_01, U_11, U_21, U_31]
plt.plot(x1, y1, ':', label = "U_{m,1}")
```

```
x2 = [0, 1/3, 2/3, 1]
y2 = [U_02, U_12, U_22, U_32]
plt.plot(x2, y2, '--', label = "U_{m,2}")
```

```
x3 = [0, 1/3, 2/3, 1]
y3 = [U_03, U_13, U_23, U_33]
plt.plot(x3, y3, '-.', label = "U_{m,3}")
```

```
x4 = [0, 1/3, 2/3, 1]
y4 = [1, e**(-5/3), e**(-10/3), e**(-5)]
plt.plot(x4, y4, '-', label = "y")
```

```
plt.xlabel('t')
plt.ylabel('U_{m,i}')
```

```
plt.title('Steifes System')
```

```
plt.legend()
```

```
plt.show()
```

```
# Betrachte nun ein größeres  $\lambda$  - mit  $\lambda = -0.01$ .
```

```
u_01 = 1
```

```
u_02 = 1
```

```
u_03 = 1
```

```
F_t_01u_01 = -0.01
```

```
F_t_02u_02 = -0.01
```

```
F_t_03u_03 = -0.01
```

```
u_11 = B[1][1]*u_01 + B[1][2]*u_02 + B[1][3]*u_03 + (Am[1][1]* F_t_01u_01  
+ Am[1][2]* F_t_02u_02  
+ Am[1][3]* F_t_03u_03) / 3
```

```
F_t_11u_11 = -0.01 * u_11
```

```
u_12 = B[2][1]*u_01 + B[2][2]*u_02 + B[2][3]*u_03 + ((Am[2][1]* F_t_01u_01  
+ Am[2][2]* F_t_02u_02  
+ Am[2][3]* F_t_03u_03) / 3  
+ (R[2][1]* F_t_11u_11) / 3)
```

```
F_t_12u_12 = -0.01 * u_12
```

```
u_13 = B[3][1]*u_01 + B[3][2]*u_02 + B[3][3]*u_03 + ((Am[3][1]* F_t_01u_01  
+ Am[3][2]* F_t_02u_02  
+ Am[3][3]* F_t_03u_03) / 3  
+ R[3][1]* F_t_11u_11 / 3  
+ R[3][2]* F_t_12u_12 / 3)
```

```
F_t_13u_13 = -0.01 * u_13
```

```
u_21 = B[1][1]*u_11 + B[1][2]*u_12 + B[1][3]*u_13 + ((Am[1][1]* F_t_11u_11  
+ Am[1][2]* F_t_12u_12  
+ Am[1][3]* F_t_13u_13) / 3)
```

$$F_{t_21u_21} = -0.01 * u_{21}$$

$$u_{22} = B[2][1]*u_{11} + B[2][2]*u_{12} + B[2][3]*u_{13} + ((Am[2][1]* F_{t_11u_11} + Am[2][2]* F_{t_12u_12} + Am[2][3]* F_{t_13u_13}) / 3 + (R[2][1]* F_{t_21u_21}) / 3)$$

$$F_{t_22u_22} = -0.01 * u_{22}$$

$$u_{23} = B[3][1]*u_{11} + B[3][2]*u_{12} + B[3][3]*u_{13} + ((Am[3][1]* F_{t_11u_11} + Am[3][2]* F_{t_12u_12} + Am[3][3]* F_{t_13u_13}) / 3 + R[3][1]* F_{t_21u_21} / 3 + R[3][2]* F_{t_22u_22} / 3)$$

$$F_{t_23u_23} = -0.01 * u_{23}$$

$$u_{31} = B[1][1]*u_{21} + B[1][2]*u_{22} + B[1][3]*u_{23} + ((Am[1][1]* F_{t_21u_21} + Am[1][2]* F_{t_22u_22} + Am[1][3]* F_{t_23u_23}) / 3)$$

$$F_{t_31u_31} = -0.01 * u_{31}$$

$$u_{32} = B[2][1]*u_{21} + B[2][2]*u_{22} + B[2][3]*u_{23} + ((Am[2][1]* F_{t_21u_21} + Am[2][2]* F_{t_22u_22} + Am[2][3]* F_{t_23u_23}) / 3 + (R[2][1]* F_{t_31u_31}) / 3)$$

$$F_{t_32u_32} = -0.01 * u_{32}$$

$$u_{33} = B[3][1]*u_{01} + B[3][2]*u_{02} + B[3][3]*u_{03} + ((Am[3][1]* F_{t_21u_21} + Am[3][2]* F_{t_22u_22} + Am[3][3]* F_{t_23u_23}) / 3 + R[3][1]* F_{t_31u_31} / 3 + R[3][2]* F_{t_32u_32} / 3)$$

$$x1 = [0, 1/3, 2/3, 1]$$

$$y1 = [u_{01}, u_{11}, u_{21}, u_{31}]$$

```

plt.plot(x1, y1, ':', label = "U_{m,1}")

x2 = [0, 1/3, 2/3, 1]
y2 = [u_02, u_12, u_22, u_32]
plt.plot(x2, y2, '--', label = "U_{m,2}")

x3 = [0, 1/3, 2/3, 1]
y3 = [u_03, u_13, u_23, u_33]
plt.plot(x3, y3, '-.', label = "U_{m,3}")

x4 = [0, 1/3, 2/3, 1]
y4 = [1, e**(-1/300), e**(-2/300), e**(-0.01)]
plt.plot(x4, y4, '-', label = "y")

plt.xlabel('t')

plt.ylabel('U_{m,i}')

plt.title('Nichtsteifes System')

plt.legend()

plt.show()

# Beginne mit einem relativ großen  $\lambda = -0.00001$ .

U_01 = 1
U_02 = 1
U_03 = 1

f_t_01U_01 = -0.00001
f_t_02U_02 = -0.00001
f_t_03U_03 = -0.00001

U_11 = B[1][1]*U_01 + B[1][2]*U_02 + B[1][3]*U_03 + ((Am[1][1]* f_t_01U_01
+ Am[1][2]* f_t_02U_02
+ Am[1][3]* f_t_03U_03) / 3)

f_t_11U_11 = -0.00001 * U_11

```

$$U_{12} = B[2][1]*U_{01} + B[2][2]*U_{02} + B[2][3]*U_{03} + ((Am[2][1]* f_{t_01}U_{01} + Am[2][2]* f_{t_02}U_{02} + Am[2][3]* f_{t_03}U_{03}) / 3 + (R[2][1]* f_{t_11}U_{11}) / 3)$$

$$f_{t_12}U_{12} = -0.00001 * U_{12}$$

$$U_{13} = B[3][1]*U_{01} + B[3][2]*U_{02} + B[3][3]*U_{03} + ((Am[3][1]* f_{t_01}U_{01} + Am[3][2]* f_{t_02}U_{02} + Am[3][3]* f_{t_03}U_{03}) / 3 + R[3][1]* f_{t_11}U_{11} / 3 + R[3][2]* f_{t_12}U_{12} / 3)$$

$$f_{t_13}U_{13} = -0.00001 * U_{13}$$

$$U_{21} = B[1][1]*U_{11} + B[1][2]*U_{12} + B[1][3]*U_{13} + ((Am[1][1]* f_{t_11}U_{11} + Am[1][2]* f_{t_12}U_{12} + Am[1][3]* f_{t_13}U_{13}) / 3)$$

$$f_{t_21}U_{21} = -0.00001 * U_{21}$$

$$U_{22} = B[2][1]*U_{11} + B[2][2]*U_{12} + B[2][3]*U_{13} + ((Am[2][1]* f_{t_11}U_{11} + Am[2][2]* f_{t_12}U_{12} + Am[2][3]* f_{t_13}U_{13}) / 3 + (R[2][1]* f_{t_21}U_{21}) / 3)$$

$$f_{t_22}U_{22} = -0.00001 * U_{22}$$

$$U_{23} = B[3][1]*U_{11} + B[3][2]*U_{12} + B[3][3]*U_{13} + ((Am[3][1]* f_{t_11}U_{11} + Am[3][2]* f_{t_12}U_{12} + Am[3][3]* f_{t_13}U_{13}) / 3 + R[3][1]* f_{t_21}U_{21} / 3 + R[3][2]* f_{t_22}U_{22} / 3)$$

$$f_{t_23}U_{23} = -0.00001 * U_{23}$$

$$U_{31} = B[1][1]*U_{21} + B[1][2]*U_{22} + B[1][3]*U_{23} + ((Am[1][1]* f_{t_21}U_{21} + Am[1][2]* f_{t_22}U_{22}$$

```

+ Am[1][3]* f_t_23U_23) / 3)

f_t_31U_31 = -0.00001 * U_31

U_32 = B[2][1]*U_21 + B[2][2]*U_22 + B[2][3]*U_23 + ((Am[2][1]* f_t_21U_21
+ Am[2][2]* f_t_22U_22
+ Am[2][3]* f_t_23U_23) / 3
+ (R[2][1]* f_t_31U_31) / 3)

f_t_32U_32 = -0.00001 * U_32

U_33 = B[3][1]*U_01 + B[3][2]*U_02 + B[3][3]*U_03 + ((Am[3][1]* f_t_21U_21
+ Am[3][2]* f_t_22U_22
+ Am[3][3]* f_t_23U_23) / 3
+ R[3][1]* f_t_31U_31 / 3
+ R[3][2]* f_t_32U_32 / 3)

x1 = [0, 1/3, 2/3, 1]

y1 = [U_01, U_11, U_21, U_31]
plt.plot(x1, y1, ':', label = "U_{m,1}")

x2 = [0, 1/3, 2/3, 1]
y2 = [U_02, U_12, U_22, U_32]
plt.plot(x2, y2, '--', label = "U_{m,2}")

x3 = [0, 1/3, 2/3, 1]
y3 = [U_03, U_13, U_23, U_33]
plt.plot(x3, y3, '-.', label = "U_{m,3}")

x4 = [0, 1/3, 2/3, 1]
y4 = [1, e**(-5/3), e**(-10/3), e**(-5)]
plt.plot(x4, y4, '-', label = "y")

plt.xlabel('t')

plt.ylabel('U_{m,i}')

plt.title('Nichtsteifes System')

```

```
plt.legend()
```

```
plt.show()
```

```
# Betrachte ein sehr kleines  $\lambda$  - mit  $\lambda = -50$ .
```

```
u_01 = 1
```

```
u_02 = 1
```

```
u_03 = 1
```

```
F_t_01u_01 = -50
```

```
F_t_02u_02 = -50
```

```
F_t_03u_03 = -50
```

```
u_11 = B[1][1]*u_01 + B[1][2]*u_02 + B[1][3]*u_03 + ((Am[1][1]* F_t_01u_01  
+ Am[1][2]* F_t_02u_02  
+ Am[1][3]* F_t_03u_03) / 3)
```

```
F_t_11u_11 = -50 * u_11
```

```
u_12 = B[2][1]*u_01 + B[2][2]*u_02 + B[2][3]*u_03 + ((Am[2][1]* F_t_01u_01  
+ Am[2][2]* F_t_02u_02  
+ Am[2][3]* F_t_03u_03) / 3  
+ (R[2][1]* F_t_11u_11) / 3)
```

```
F_t_12u_12 = -50 * u_12
```

```
u_13 = B[3][1]*u_01 + B[3][2]*u_02 + B[3][3]*u_03 + ((Am[3][1]* F_t_01u_01  
+ Am[3][2]* F_t_02u_02  
+ Am[3][3]* F_t_03u_03) / 3  
+ R[3][1]* F_t_11u_11 / 3  
+ R[3][2]* F_t_12u_12 / 3)
```

```
F_t_13u_13 = -50 * u_13
```

```
u_21 = B[1][1]*u_11 + B[1][2]*u_12 + B[1][3]*u_13 + ((Am[1][1]* F_t_11u_11  
+ Am[1][2]* F_t_12u_12
```

$$+ Am[1][3] * F_t_13u_13) / 3)$$

$$F_t_21u_21 = -50 * u_21$$

$$u_22 = B[2][1]*u_11 + B[2][2]*u_12 + B[2][3]*u_13 + ((Am[2][1]* F_t_11u_11 + Am[2][2]* F_t_12u_12 + Am[2][3]* F_t_13u_13) / 3 + (R[2][1]* F_t_21u_21) / 3)$$

$$F_t_22u_22 = -50 * u_22$$

$$u_23 = B[3][1]*u_11 + B[3][2]*u_12 + B[3][3]*u_13 + ((Am[3][1]* F_t_11u_11 + Am[3][2]* F_t_12u_12 + Am[3][3]* F_t_13u_13) / 3 + R[3][1]* F_t_21u_21 / 3 + R[3][2]* F_t_22u_22 / 3)$$

$$F_t_23u_23 = -50 * u_23$$

$$u_31 = B[1][1]*u_21 + B[1][2]*u_22 + B[1][3]*u_23 + ((Am[1][1]* F_t_21u_21 + Am[1][2]* F_t_22u_22 + Am[1][3]* F_t_23u_23) / 3)$$

$$F_t_31u_31 = -50 * u_31$$

$$u_32 = B[2][1]*u_21 + B[2][2]*u_22 + B[2][3]*u_23 + ((Am[2][1]* F_t_21u_21 + Am[2][2]* F_t_22u_22 + Am[2][3]* F_t_23u_23) / 3 + (R[2][1]* F_t_31u_31) / 3)$$

$$F_t_32u_32 = -50 * u_32$$

$$u_33 = B[3][1]*u_01 + B[3][2]*u_02 + B[3][3]*u_03 + ((Am[3][1]* F_t_21u_21 + Am[3][2]* F_t_22u_22 + Am[3][3]* F_t_23u_23) / 3 + R[3][1]* F_t_31u_31 / 3 + R[3][2]* F_t_32u_32 / 3)$$

$$x1 = [0, 1/3, 2/3, 1]$$

```

y1 = [u_01, u_11, u_21, u_31]
plt.plot(x1, y1, ':', label = "U_{m,1}")

x2 = [0, 1/3, 2/3, 1]
y2 = [u_02, u_12, u_22, u_32]
plt.plot(x2, y2, '--', label = "U_{m,2}")

x3 = [0, 1/3, 2/3, 1]
y3 = [u_03, u_13, u_23, u_33]
plt.plot(x3, y3, '-.', label = "U_{m,3}")

x4 = [0, 1/3, 2/3, 1]
y4 = [1, e**(-1/300), e**(-2/300), e**(-0.01)]
plt.plot(x4, y4, '-', label = "y")

plt.xlabel('t')
plt.ylabel('U_{m,i}')

plt.title('Steifes System')

plt.legend()

plt.show()

```

Fachbereich Mathematik, Informatik und Physik

SELBSTSTÄNDIGKEITSERKLÄRUNG

Name: ANDERSON	(BITTE nur Block- oder Maschinenschrift verwenden.)
Vorname(n): LOUIS NICOLAS JARDINE	
Studiengang: MATHEMATIK, BACHELOR	
Matr. Nr.: 5173212	

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Bachelorarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keiner anderen Universität als Prüfungsleistung eingereicht.

Datum: 22/6/23

Unterschrift: N. Anderson