Eine flexible Klasse von local time stepping Verfahren

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT) genehmigte

DISSERTATION VON

M. Sc. ABDULLAH DEMIREL

aus Neuss

Tag der mündlichen Prüfung: 16.04.2014
Referentin: Prof. Dr. Marlis Hochbruck
Korreferent: Prof. Dr. Tobias Jahnke

Abstract

In this work we consider numerical time integrators for partial differential equations over a domain, where we need a locally refined spatial discretization. We will discuss the problems and challenges resulting from such problems. We will also consider the construction, the analysis and the implementation of suitable numerical integrators. The application of these integrators to problems of nanophotonics will also be discussed in this work.

The integrators which are presented in this thesis are based on exponential multistep methods. These methods were introduced by Certaine [9]. Hochbruck and Ostermann [33] studied also these integrators, they explained the construction of these methods in a detailed way and performed the related error analysis in an abstract framework of semigroups. Their work is fundamental for this thesis.

Zusammenfassung

In dieser Dissertation werden wir numerische Zeitintegratoren für partielle Differentialgleichungen behandeln. Speziell untersuchen wir Integratoren für partielle Differentialgleichungen auf Gebieten, deren räumliche Diskretisierung eine lokale Verfeinerung erfordert. Wir werden die hiermit verbundenen Schwierigkeiten diskutieren und die Konstruktion, die Analyse und die Implementierung geeigneter Integratoren untersuchen sowie auch deren effiziente Implementierung für Anwendungen aus der Nanophotonik betrachten.

Die in dieser Dissertation behandelten Integratoren basieren auf exponentiellen Mehrschrittverfahren. Diese Verfahren wurden zuerst von Certaine [9] vorgestellt. Hochbruck und Ostermann haben in [33] neben der Konstruktion dieser Integratoren auch deren Fehleranalyse vorgenommen. Deren Arbeit bildet zugleich auch die Grundlage zu dieser Dissertation.

Danksagung

Ich möchte an dieser Stelle die Gelegenheit nutzen, um meinen Dank in meiner letzten wissenschaftlichen Arbeit auszusprechen. An erster Stelle bin ich Allah unendlich dankbar dafür, dass mein Leben stets mit allem Guten versehen wurde. Dann möchte ich mich natürlich sehr herzlich bei meinen Eltern Mehmet und Hatice Demirel sowie bei meinen Geschwistern Yasin und Rıdvan Demirel bedanken, die mich zu jeder Zeit in meinem Leben unterstützt, mir Ihre Liebe geschenkt und für mein Wohl gebetet haben. Vergessen möchte ich auch nicht, meiner Nichte Şevval und meinen Neffen Yunus Emre und Yavuz Selim sowie deren Mutter Yasemin dafür zu danken, dass sie uns mit ihrer Liebe bereichert haben.

Meinen Freunden Cengiz, Imène, Khadija, Kübra, Resmiye, Sadet, Tuğba und Züleyha möchte ich auch einen herzlichen Dank dafür aussprechen, dass sie mich stets motiviert und mein Leben mit Ihrer Freundschaft verschönert haben. Ich verdanke meiner Familie und meinen Freunden viele Momente der Freude, die ich in meinen Erinnerungen so gut wie möglich festhalte, um mich stets erneut darüber glücklich zu schätzen. Diese Menschen stellen einen wesentlichen Teil meines Reichtums dar und ich denke, dass ich diesen Menschen nicht genug dafür danken kann.

Einen weiteren großen Dank möchte ich meiner Referentin Prof. Dr. Marlis Hochbruck aussprechen und das nicht nur wegen der hervorragenden Betreuung ihrerseits, sondern auch für ihre Bemühungen in jeder Hinsicht. Insbesondere die Arbeitsgruppe sowie auch das schöne Arbeitsklima am KIT haben dafür gesorgt, dass ich mich in Karlsruhe von Beginn an wohlgefühlt habe. Hierzu haben vor allem mein Seelenverwandter Bilal Haddou-Temsamani und meine Arbeitskollegen Christian Knieling und Tomislav Pažur beigetragen und dafür möchte ich mich besonders bei ihnen bedanken. Sehr herzlich danke ich auch Prof. Dr. Kurt Busch und Dr. Jens Niegemann für die interdisziplinäre Zusammenarbeit. Darüber hinaus möchte ich mich bei Prof. Dr. Tobias Jahnke für die Übernahme des Korreferats bedanken. Insbesondere möchte ich die Gelegenheit nutzen und Prof. Dr. Fritz Grunewald, Prof. Dr. Marlis Hochbruck, Prof. Dr. Florian Jarre, Prof. Dr. Elena Klimenko und Prof. Dr. Klaus Steffen meinen Dank aussprechen, da ich den Großteil meines mathematischen Wissens und mein Interesse an der Mathematik Ihnen zu verdanken habe. Letztlich möchte ich natürlich allen Mitgliedern meiner Arbeitsgruppe und den Mitarbeitern der Mathematischen Fakultät meinen Dank aussprechen.

Die Dissertation ist im Rahmen eines Projekts des Graduiertenkollegs 1294 „Analysis, Simulation und Design nanotechnologischer Prozesse“ entstanden, das von der Deutschen Forschungsgemeinschaft (DFG) gefördert wird. Als Mitglied des Graduiertenkollegs möchte ich mich für diese Förderung bedanken. Möglicherweise habe ich vergessen, mich bei einigen zu bedanken, hiermit entschuldige ich mich dafür und bedanke mich auch bei diesen Personen. Meine Danksagung möchte ich mit einigen Worten in meiner Muttersprache abschließen, um meinen Dank auch in der von mir gewünschten Art formulieren zu können.

Şükür ve Teşekkür

Allah'ım şükürler olsun bana sunduğun onlarca nimetlere,
beni senin rızan için yürüdüğüm bu yolda ulaştırdın hedeflere.
Ben yalnız senin rızan için sığınırım dualara ve dileklere,
ve ancak senin izninle katılırım senin rızana erişenlere.

Annemin ve babamın hakkı gerçekten bir ömur ödenmez,
ne kadar teşekkür etsem, onlar için ne etsem yetmez.
Bilirim ki onların bana olan sevgisi iki cihanda bitmez
ve dilim dönduğü sürece onlara olan dualarım dinmez.

Abilerim Yasin ve Rıdvan'ın yardım ve duaları hatrımda,
karşıma çıkan her zorlukta onları bulurum yanı başımda.
Sizin sevginiz dolaşır benim kanımda ve damarlarımda,
bu yüzden sizin isimlerinizi zikrederim bütün dualarımda.

Dostlarım sizin desteklerinizi de hiçbir zaman unutmadım,
sözlerinizi, fikrinizi ve de düşüncelerinizi daima umursadım.
Esirgemediğiniz dualarınız ile yanımda oldunuz adım adım
ve sizin gibi bende bu sevgiyi en güzel dualarımda sakladım.

Bir konuda Allah'a tevekkül etmeden önce bağlamalısın deveni,
ben elimden geleni yaptım artık kabul edeceğim Allah'tan geleni.
Allah'a şükür sevenlerimin tamamı karşılıksız sever beni,
Allah'ım ben bilirim senden gelen hayrın elbet vardır bir nedeni.

Inhaltsverzeichnis

1 Einleitung

Die Konstruktion und die Analyse von Zeitintegratoren stellt in der numerischen Mathematik ein bedeutendes Forschungsgebiet dar. Viele Probleme der Ingenieur- und Naturwissenschaften lassen sich mathematisch anhand von zeitabhängigen Differentialgleichungen beschreiben und können mithilfe dieser Modellierungen simuliert werden. Aufgrund der Fortschritte in der Technik und der Forschung besteht ein immer größeres Interesse an aufwendigeren Simulationen, die auch aus numerischer Sicht neue Herausforderungen darstellen. Dies hat zur Folge, dass stets eine Nachfrage nach geeigneten numerischen Verfahren besteht, die eine gute Simulation ermöglichen.

In dieser Dissertation werden wir uns auf die Konstruktion und Analyse numerischer Zeitintegratoren für partielle Differentialgleichungen konzentrieren. Hierbei stehen Probleme, wo die räumliche Diskretisierung geometrisch bedingt eine lokale Verfeinerung aufweist, im Mittelpunkt. Wir sprechen von einer lokalen Verfeinerung, falls nur in einem kleinen Teil der Diskretisierung deutlich kleinere Elemente (z.B. Intervalle, Dreiecke, Tetraeder) auftreten.

Der Aufbau dieser Arbeit ist wie folgt. Im einleitenden Kapitel werden wir die Schwierigkeiten ansprechen, die bei dieser Art von Problemen in Erscheinung treten. Darüber hinaus werden wir auch mit Hilfe eines Beispiels die Konstruktion geeigneter Zeitintegratoren motivieren. Im zweiten Kapitel werden wir bekannte und relevante Resultate zu den klassischen und exponentiellen Mehrschrittverfahren zusammentragen. Zudem stellen wir eine Möglichkeit der Konstruktion klassischer bzw. exponentieller Mehrschrittverfahren vor, die bezüglich der Stabilität optimale Eigenschaften vorweisen. Im darauffolgenden Kapitel werden wir die verschiedenen Umsetzungsmöglichkeiten dieser Verfahren diskutieren.

Die Implementierung sowie die Pseudo-Codes zu den vorgestellten Mehrschrittverfahren werden im vierten Kapitel vorgestellt. Unter Verwendung der Codes präsentieren wir zudem unsere Ergebnisse anhand von einigen Beispielen. Im abschließenden Teil dieser Arbeit befassen wir uns mit nichtlinearen Problemen und werden versuchen die Anwendung und die Fehleranalyse vom semilinearen Fall auf den nichtlinearen Fall zu übertragen.

Für die angesprochenen Problemstellungen werden bisher diverse Verfahren angewendet. Hierzu zählen beispielsweise exponentielle Mehrschrittverfahren [33] oder low-storage

Runge-Kutta-Verfahren [41]. Darüber hinaus gibt es sogenannte local time stepping Verfahren (kurz: LTS-Verfahren) [11, 21, 22], die ganz speziell für solche Problemstellungen konstruiert worden sind.

In der vorliegenden Arbeit werden wir eine flexible Verfahrensklasse von exponentiellen Mehrschrittverfahren und local time stepping Verfahren vorstellen, die auf den klassischen Adams-Verfahren beruhen. Abhängig von einem gegebenen Problem werden wir damit in der Lage sein, den optimalen Vertreter dieser Verfahrensklasse bzgl. der Stabilität zu ermitteln. Wir werden sehen, dass die Wahl des optimalen Vertreters sich wegen des geringeren Rechenaufwands für praktische Umsetzungen lohnt. Aus der mathematischen Perspektive werden wir zugleich feststellen, dass die Fehleranalysen der verschiedenen Verfahrensklassen sehr eng miteinander verbunden sind.

1.1 Motivation

Zur Motivation betrachten wir das folgende Beispiel (siehe auch [33]) der gedämpften Wellengleichung in einer Raumdimension

$$\begin{aligned} & \frac{\partial^2 U}{\partial t^2}(x,t) + \sigma \frac{\partial U}{\partial t}(x,t) = \frac{\partial^2 U}{\partial x^2}(x,t), \\ \text{mit} \quad & U(0,t) = U(6,t), \quad x \in \Omega = [0,6], \quad t \geq 0, \\ \text{und} \quad & U(x,0) = 0, \quad \frac{\partial U}{\partial t}(x,0) = \sin(x\pi). \end{aligned} \tag{1.1}$$

Die exakte Lösung dieser Wellengleichung entspricht dem nachstehenden Ausdruck

$$U(x,t) = \frac{2e^{-\frac{1}{2}\sigma t}}{\sqrt{4\pi^2 - \sigma^2}} \sin(\pi x) \sin\left(\frac{t\sqrt{4\pi^2 - \sigma^2}}{2}\right). \tag{1.2}$$

Das Problem (1.1) lässt sich äquivalent in ein System erster Ordnung

$$\begin{bmatrix} \frac{\partial U}{\partial t}(x,t) \\ \frac{\partial V}{\partial t}(x,t) \end{bmatrix} = \begin{bmatrix} 0 & I \\ \frac{\partial^2}{\partial x^2} & -\sigma I \end{bmatrix} \begin{bmatrix} U(x,t) \\ V(x,t) \end{bmatrix}, \quad V(x,t) := \frac{\partial U}{\partial t}(x,t) \tag{1.3}$$

umschreiben. Für die räumliche Diskretisierung von (1.3) verwenden wir ein Finite-Differenzen-Schema der Ordnung vier, wobei wir die Gitterweite im Teilintervall $[2,4]$ durch einen Faktor $r_s = 1, 2, 4, 8$ verkleinern, um eine lokale Verfeinerung zu erzeugen.

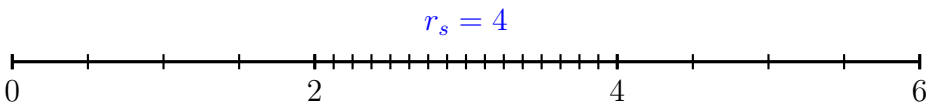

Abbildung 1.1: Räumliche Diskretisierung von Ω mit Verfeinerungsfaktor $r_s = 4$.

Die räumliche Diskretisierung von (1.3) führt auf ein lineares Anfangswertproblem

$$y' = Ay =: f(t, y) \quad \text{mit} \quad y(t_0) = y_0,$$

das wir mit Hilfe des klassischen Adams-Verfahrens der Ordnung vier

$$y_{n+1} = y_n + \frac{\tau}{24}\big(55f(t_n, y_n) - 59f(t_{n-1}, y_{n-1}) + 37f(t_{n-2}, y_{n-2}) - 9f(t_{n-3}, y_{n-3})\big)$$

auf dem Zeitintervall tspan $= [0, 1]$ numerisch lösen, wobei $t_j = j\tau$ für $j \geq 0$ gilt. In Abbildung 1.2 können wir erkennen, dass die numerische Lösung für hinreichend kleine Zeitschrittweiten τ gegen die exakte Lösung konvergiert und wir sehen, dass das numerische Verfahren die Konvergenzordnung vier besitzt.

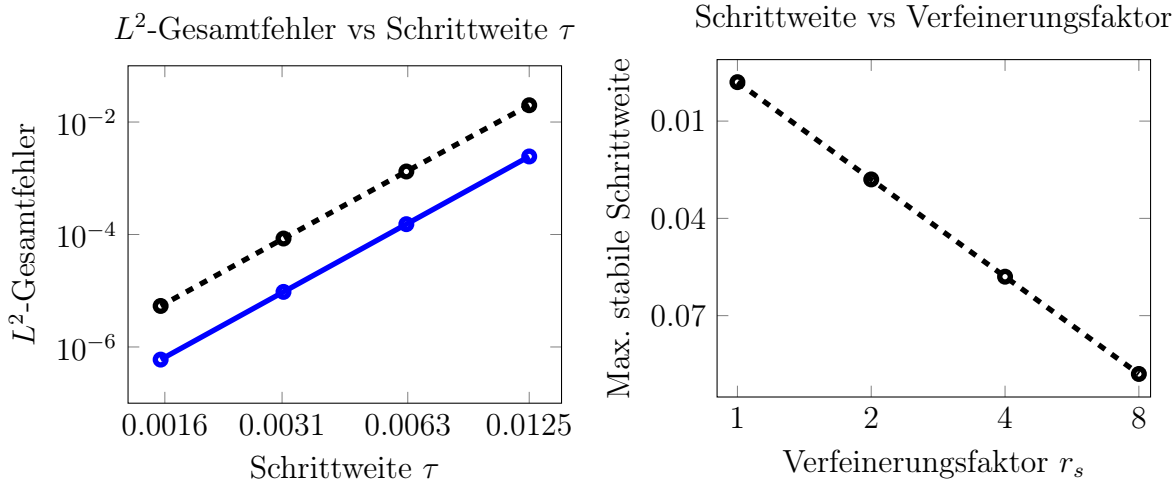

Abbildung 1.2: Links: Gesamtfehler (zeitliche und räumliche Diskretisierung) der numerischen Approximation (schwarz) zur exakten Lösung für $p = 4$ in der L^2-Norm. Rechts: Maximal stabile Schrittweiten in Abhängigkeit des Verfeinerungsfaktor r_s.

Wir beobachten auch, dass die Erhöhung des Verfeinerungsfaktors r_s eine Schrittweiteneinschränkung zur Folge hat. Dieser Effekt ist insbesondere in den Fällen von Nachteil, wo die räumliche Diskretisierung des Gebiets Ω einen sehr kleinen Bereich mit einer lokalen Verfeinerung aufweist, da dies zu einer Schrittweitenreduktion des ganzen Systems führt.

An diesem Beispiel erkennen wir, dass es durchaus sinnvoll ist, das Problem zu zerlegen und die Anteile mit einem geeigneten Verfahren unterschiedlich zu behandeln. Diese Idee ist keineswegs neu und sogar grundlegend für diverse Integratoren wie z.B. LTS-Verfahren [11, 21, 22], IMEX-Verfahren [36] oder Multirate-Verfahren [19, 48]. In dieser Arbeit werden wir uns hauptsächlich auf LTS-Verfahren konzentrieren und eine flexible Klasse von Verfahren vorstellen, die eine Alternative zu den bisher verwendeten Integratoren darstellen. Bevor wir uns der Konstruktion dieser Verfahren widmen, sollten wir vorerst die vorliegenden Schwierigkeiten bzw. Herausforderungen ansprechen.

1.2 Herausforderungen

Wie im motivierenden Beispiel, werden wir uns mit Anfangswertproblemen

$$y' = f(t, y), \quad y(t_0) = y_0 \tag{1.4}$$

befassen, die aus der räumlichen Diskretisierung einer partiellen Differentialgleichung über einem Gebiet Ω resultieren und setzen stets voraus, dass die Lösung y von (1.4) hinreichend glatt ist. Systeme dieser Art haben oft eine sehr hohe Dimension und dies stellt ein Problem für die Anwendung impliziter Verfahren dar, da im Allgemeinen in jedem Zeitschritt ein nichtlineares Gleichungssystem zu lösen ist. Der damit verbundene Rechenaufwand ist oft zu hoch.

Hinzu kommt, dass die lokale Verfeinerung eine Steifigkeit des Problems induziert. Wie wir im motivierenden Beispiel gesehen haben, führt dies aus Stabilitätsgründen zu einer Schrittweitenreduktion und dementsprechend ist auch die Anwendung expliziter Integratoren nicht sehr geeignet. Wir möchten deswegen Zeitintegratoren konstruieren, die eine Art „Kompromiss“ zwischen impliziten und expliziten Verfahren bilden sollen. Aus diesem Grund ist es sinnvoll das Anfangswertproblem (1.4) umzuformulieren in

$$y' = f_{\text{fine}}(t, y) + f_{\text{coarse}}(t, y), \quad y(t_0) = y_0, \tag{1.5}$$

mit einem sogenannten steifen (feinen) Anteil f_{fine} und einem nicht steifen (groben) Anteil f_{coarse}, die vom Integrator jeweils unterschiedlich behandelt werden. Die rechte Seite des Anfangswertproblems wird damit bezüglich der Geometrie in zwei Teile zerlegt.

Es ist wichtig zu veranschaulichen, was bei einer solchen Art von Splitting zu beachten ist. Betrachten wir, wie in Abbildung 1.1, ein Gitter der Form: Die Knoten in den

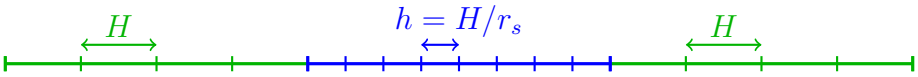

Abbildung 1.3: Räumliche Diskretisierung von $\Omega = [0, 8H + 8h]$ mit $h = \frac{H}{r_s}$ mit Verfeinerungsfaktor $r_s = 2$.

grünen Bereichen werden dem groben Teil und die vom blauen Bereich dem feinen Teil zugeordnet. Die räumliche Diskretisierung des Laplace-Operators über diesem Gitter mit Finite-Differenzen-Schemen der Ordnung zwei, vier und sechs liefert jeweils eine Matrix A, deren Koeffizienten in Abhängigkeit von H und r_s stehen. Fixieren wir beispielsweise $H = 1$ und betrachten die Koeffizienten für $r_s \to \infty$, so können wir erkennen, welche Einträge der Matrix A für $r_s \to \infty$ nicht beschränkt sind.

Die Berechnung der Koeffizienten der Matrix A ist nicht sonderlich schwer, jedoch aufgrund der Fallunterscheidungen recht umfangreich aufzuschreiben. Aus dem Grund präsentieren wir an dieser Stelle das Resultat anhand von Graphiken. In Abbildung 1.4 werden die beschränkten und unbeschränkten Koeffizienten, die aus der Diskretisierung des Laplace-Operators für $r_s \to \infty$ mit dem jeweiligen Finite-Differenzen-Schema resultieren, jeweils in einer Graphik visualisiert. Ein Splitting wie in Abbildung 1.3 würde die Blockmatrix S, die in Abbildung 1.4 in blau markiert ist, als den feinen Anteil beschreiben. Wir sehen aus der ersten Graphik der Abbildung 1.4, dass S bei der Diskretisierung mit einem Finiten-Differenzen-Schema der zweiten Ordnung alle unbeschränkten Koeffizienten enthält. Aus den zwei übrigen Graphiken können wir allerdings erkennen, dass abhängig von der Diskretisierung der Einfluss vom feinen Teil zum groben Teil ebenfalls zu unbeschränkten Koeffizienten führen kann, die bei einem solchen Splitting vernachlässigt werden. Insbesondere ist dies von Bedeutung, wenn beispielsweise der beschränkte Anteil vom unbeschränkten Anteil getrennt werden soll, um dann von dieser Eigenschaft zu profitieren.

Wir haben gesehen, dass der Einfluss des feinen Teils auf den groben Teil insbesondere von der Diskretisierung abhängt. Bei der Verwendung von Finite-Differenzen-Schemen tritt dieser Einfluss abhängig von der Ordnung der räumlichen Diskretisierung auf. Denn die Länge des zu verwendenden Differenzensterns ist abhängig von der Ordnung. Zugleich ist die Länge des Differenzensterns dafür verantwortlich, wie weit sich der Einfluss

vom feinen zum groben Teil erstreckt. Bei einer Diskretisierung mit Finiten-Elementen erstreckt sich dieser Einfluss lediglich auf die benachbarten Elemente. Die Basisfunktion eines jeden Knotens besitzt nämlich nur einen kompakten Träger, dass höchstens so groß ist wie der Teilbereich der benachbarten Elemente, die diesen Knoten enthalten. Dies ist auch bei der Diskretisierung mit Discontinuous-Galerkin-Methoden [27] der Fall. In diesem Fall tritt der Einfluss durch die Verwendung des numerischen Flusses auf, da der numerische Fluss oft eine Beziehung zwischen den Nachbarelementen darstellt.

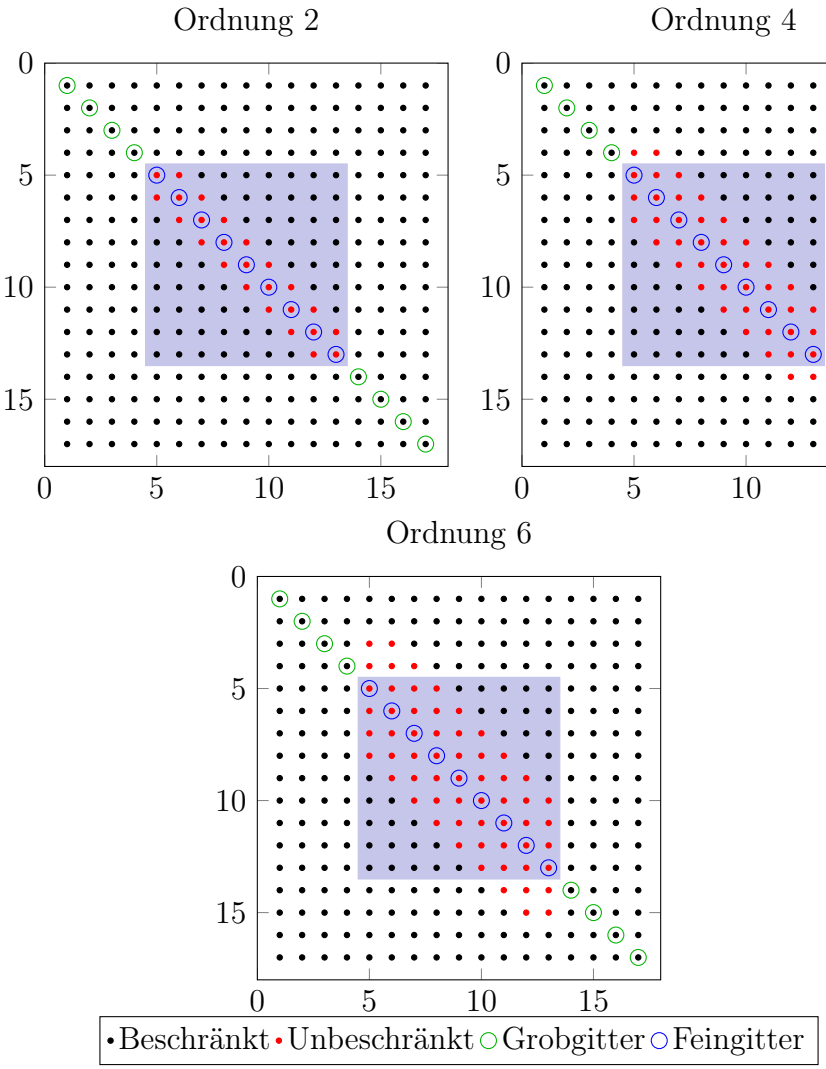

Abbildung 1.4: Beschränkte und unbeschränkte Einträge der Matrix $A \approx \Delta$, die aus einer Finite-Differenzen-Diskretisierung (Ordnungen: 2, 4, 6) resultieren. Der blaue Bereich entspricht der Blockmatrix S, die den feinen Anteil darstellt.

2 Mehrschrittverfahren

In diesem Kapitel fassen wir zuerst einige bekannte Resultate für klassische und exponentielle Mehrschrittverfahren aus der Literatur zusammen. Diesbezüglich konzentrieren wir uns vorerst auf sogenannte semilineare Probleme

$$y' = f(t,y) = \underbrace{Ay}_{=f_{\text{fine}}(t,y)} + \underbrace{g(t,y)}_{=f_{\text{coarse}}(t,y)}, \quad y(t_0) = y_0, \tag{2.1}$$

wobei wir stets voraussetzen, dass die Lösung von (2.1) hinreichend glatt ist. Bevor wir uns den exponentiellen Mehrschrittverfahren widmen, möchten wir vorher die Konstruktion und die Idee klassischer Mehrschrittverfahren anhand der Quellen [24, 25, 31] wiedergeben. Diese Informationen werden uns im weiteren Verlauf helfen, Vergleiche zu erstellen und Rückschlüsse zu ziehen.

2.1 Klassische Mehrschrittverfahren

Zu den klassischen Mehrschrittverfahren zählen die expliziten und impliziten Adams-Verfahren, die Predictor-Corrector-Verfahren sowie die BDF-Verfahren. Wir beschränken uns dabei auf die Zeitintegratoren, die auf Adams-Verfahren basieren.

2.1.1 Explizite Adams-Verfahren

Die exakte Lösung des Anfangswertproblems (2.1) können wir nach dem Hauptsatz der Differential- und Integralrechnung durch

$$y(t_n + \tau) = y(t_n) + \int_{t_n}^{t_n+\tau} f\big(t, y(t)\big)\mathrm{d}t \tag{2.2}$$

beschreiben. Die Idee expliziter Adams-Verfahren besteht darin, die Funktion f durch ein Interpolationspolynom zu approximieren, um eine Näherung an $y(t_n + \tau)$ berechnen zu können. Zur Bestimmung des Interpolationspolynoms wird ein Datensatz von Punkten

$$(t_{n-k+1}, f_{n-k+1}), \ldots, (t_n, f_n) \quad \text{mit} \quad f_j := f\big(t_j, y_j\big) \quad \text{und} \quad t_j = t_0 + j\tau, \quad j \in \mathbb{N}_0$$

verwendet, der notfalls durch eine Startwertprozedur berechnet wird.

Nach der Newton'schen Interpolationsformel kann das Interpolationspolynom in der folgenden Form dargestellt werden

$$p_n(t_n + \theta\tau) = \sum_{j=0}^{k-1} (-1)^j \binom{-\theta}{j} \nabla^j f_n. \tag{2.3}$$

Die in dieser Darstellung auftretenden Rückwärtsdifferenzen $\nabla^j f_n$ sind durch

$$\nabla^j f_n = \begin{cases} f_n, & j = 0 \\ \nabla^{j-1} f_n - \nabla^{j-1} f_{n-1}, & j > 0 \end{cases}$$

rekursiv definiert. Die grundlegende Idee der expliziten Adams-Verfahren wird in der Abbildung 2.1 veranschaulicht.

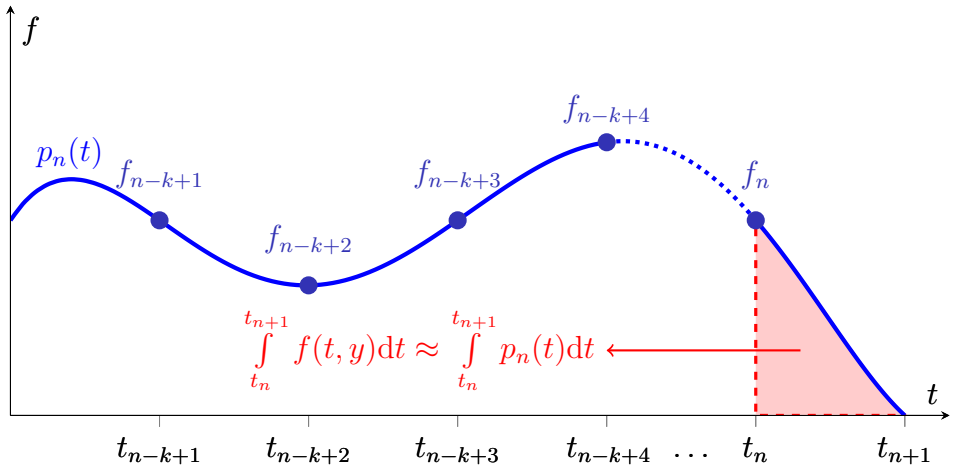

Abbildung 2.1: Graphische Darstellung der Idee der expliziten Adams-Verfahren

Die Approximation des Integrals aus (2.2) liefert uns die Beziehung:

$$\begin{array}{ccccccc} y(t_n+\tau) & = & y(t_n) & + & \displaystyle\int_{t_n}^{t_n+\tau} f(t,y)\,\mathrm{d}t & & \\ \wr & & \wr & & \wr & & \\ y_{n+1} & = & y_n & + & \displaystyle\int_{t_n}^{t_n+\tau} p_n(t)\,\mathrm{d}t & = & y_n + \tau \displaystyle\int_0^1 p_n(t_n+\theta\tau)\,\mathrm{d}\theta. \end{array} \tag{2.4}$$

Setzen wir schließlich das Polynom (2.3) in (2.4) ein, so können wir die expliziten Adams-Verfahren wie folgt beschreiben

$$y_{n+1} \;=\; y_n + \tau \sum_{j=0}^{k-1} \gamma_j^{\mathrm{eA}} \nabla^j f_n \quad \text{mit} \quad \gamma_j^{\mathrm{eA}} := \int_0^1 (-1)^j \binom{-\theta}{j} \,\mathrm{d}\theta \quad \text{und} \quad k \geq 1. \tag{2.5}$$

Bezeichnen wir mit $k \in \mathbb{N}$ die Anzahl der explizit verwendeten Interpolationspunkte, so können wir das zugehörige explizite k-Schritt Adams-Verfahren angeben. Im Folgenden sind schließlich die expliziten Adams-Verfahren für $k = 1, 2, 3$ aufgeführt

$$\begin{aligned} k = 1: &\qquad y_{n+1} = y_n + \tau f_n \\ k = 2: &\qquad y_{n+1} = y_n + \frac{\tau}{2}(3f_n - f_{n-1}) \\ k = 3: &\qquad y_{n+1} = y_n + \frac{\tau}{12}(23f_n - 16f_{n-1} + 5f_{n-2}). \end{aligned} \tag{2.6}$$

2.1.2 Implizite Adams-Verfahren

Wie im letzten Abschnitt behandelt, verwenden explizite Adams-Verfahren ausschließlich bereits berechnete Interpolationspunkte. Implizite Adams-Verfahren unterscheiden sich von expliziten Adams-Verfahren dadurch, dass ein zusätzlicher impliziter Interpolationspunkt für die Konstruktion des Interpolationspolynoms verwendet wird. Demnach ist der Datensatz eines impliziten Adams-Verfahren von der Form

$$(t_{n-k+1}, f_{n-k+1}), \ldots, (t_{n+1}, f_{n+1}) \quad \text{mit} \quad f_j := f\big(t_j, y_j\big) \quad \text{und} \quad t_j = t_0 + j\tau, \quad j \in \mathbb{N}_0.$$

Das zugehörige Interpolationspolynom kann ebenfalls mit der Newton'schen Interpolationsformel dargestellt werden

$$p_n(t_n + \theta\tau) = \sum_{j=0}^{k} (-1)^j \binom{-\theta + 1}{j} \nabla^j f_{n+1}. \tag{2.7}$$

Diese Kernidee der impliziten Adams-Verfahren und der Unterschied zu den expliziten Adams-Verfahren wird in Abbildung 2.2 graphisch dargestellt.

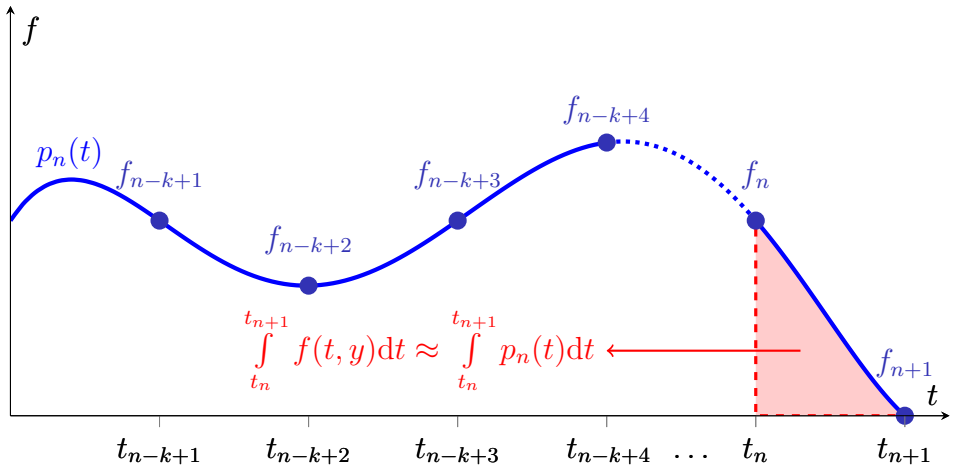

Abbildung 2.2: Graphische Darstellung der Idee der impliziten Adams-Verfahren

Setzen wir nun, wie im expliziten Fall, das Polynom (2.7) in (2.4) ein, so ergibt sich hieraus die Vorschrift der impliziten Adams-Verfahren

$$y_{n+1} = y_n + \tau \sum_{j=0}^{k} \gamma_j^{\mathrm{iA}} \nabla^j f_n \quad \text{mit} \quad \gamma_j^{\mathrm{iA}} := \int_0^1 (-1)^j \binom{-\theta+1}{j} \mathrm{d}\theta \quad \text{und} \quad k \geq 0. \tag{2.8}$$

Im vorigen Abschnitt haben wir $k \in \mathbb{N}_0$ als die Anzahl der explizit verwendeten Interpolationspunkte definiert. Implizit verwendete Interpolationspunkte sind in k nicht einbezogen, so dass die Wahl $k = 0$ durchaus möglich ist. Die impliziten Adams-Verfahren für $k = 0, 1, 2$ sind von der Form:

$$\begin{aligned} k = 0: &\quad y_{n+1} = y_n + \tau f_{n+1} \\ k = 1: &\quad y_{n+1} = y_n + \frac{\tau}{2}(f_{n+1} + f_n) \\ k = 2: &\quad y_{n+1} = y_n + \frac{\tau}{12}(5f_{n+1} + 8f_n - 1f_{n-1}). \end{aligned} \tag{2.9}$$

2.1.3 Predictor-Corrector-Verfahren

Die Predictor-Corrector-Verfahren sind explizite Verfahren, die auf den bisher vorgestellten impliziten und expliziten Adams-Verfahren beruhen. Sie stellen eine Kombination dieser Verfahren dar und haben durchaus positive Eigenschaften, die wir später noch diskutieren werden. Des Weiteren bieten diese Verfahren eine Flexibilität in der Anwendung, die auf der Fixpunktiteration beruht und unterschiedliche Konstellationen ermöglicht.

Jedes Predictor-Corrector-Verfahren lässt sich mit Hilfe der drei Prozesse **P**rediction (Vorhersage), **E**valuation (Auswertung) und **C**orrection (Korrektur) beschreiben. Im Folgenden werden wir die drei Prozesse sowohl mathematisch als auch anschaulich vorstellen.

Prediction: (P)
In diesem Schritt wird ein explizites k-Schritt Adams-Verfahren angewendet. Nach der Beschreibung in Abschnitt 2.1.1 wird mit Hilfe eines Datensatzes eine Approximation der folgenden Form berechnet

$$y(t_{n+1}) \approx \bar{y}_{n+1} = y_n + \tau \sum_{j=0}^{k-1} \gamma_j^{\mathrm{eA}} \nabla^j f_n. \tag{2.10}$$

Evaluation: (E)
Dieser Prozess verwendet die Approximation aus dem Prediction-Prozess, um eine Approximation der rechten Seite zur Zeit t_{n+1} zu bestimmen

$$f\big(t_{n+1}, y(t_{n+1})\big) \approx f(t_{n+1}, \bar{y}_{n+1}) =: \bar{f}_{n+1}.$$

Correction: (C)
Im Correction-Prozess benutzen wir im Gegensatz zum Prediction-Prozess ein implizites Adams-Verfahren, jedoch wird nicht der übliche Datensatz eines impliziten Adams-Verfahrens

$$(t_{n-k+1}, f_{n-k+1}), \ldots, (t_{n+1}, f_{n+1}) \quad \text{mit} \quad f_j := f\big(t_j, y_j\big) \quad \text{und} \quad t_j = t_0 + j\tau, \quad j \in \mathbb{N}_0$$

verwendet. Mit den Approximationen aus dem Prediction- und Evaluation-Prozess kann stattdessen der folgende Datensatz verwendet werden

$$(t_{n-k+2}, f^*_{n-k+2}), \ldots, (t_{n+1}, f^*_{n+1}) \quad \text{mit} \quad f^*_j := \begin{cases} f_j, & j \neq n+1 \\ \bar{f}_j, & j = n+1 \end{cases}, \quad j \in \mathbb{N}_0. \tag{2.11}$$

Daraus ergibt sich die Approximation

$$y_{n+1} \;=\; y_n + \tau \sum_{j=0}^{k-1} \gamma_j^{\mathrm{iA}} \nabla^j f_{n+1}^* \quad \text{mit} \quad \gamma_j^{\mathrm{iA}} := \int_0^1 (-1)^j \binom{-\theta+1}{j} \mathrm{d}\theta. \tag{2.12}$$

Die angesprochene Flexibilität der Predictor-Corrector-Verfahren besteht zum einen darin, dass die Evaluation- und Correction-Prozesse gemeinsam als eine Fixpunktiteration aufgefasst werden können. Aus dem Grund werden Predictor-Corrector-Verfahren oft in Konstellationen wie $\mathbf{P(EC)}^m\mathbf{E}$ mit $m \geq 1$ angewendet. Zum anderen kann statt (2.11) auch ein Datensatz der nachstehenden Form

$$(t_{n-k+1}, f_{n-k+1}^*), \ldots, (t_{n+1}, f_{n+1}^*) \quad \text{mit} \quad f_j^* := \begin{cases} f_j, & j \neq n+1 \\ \bar{f}_j, & j = n+1 \end{cases}, \quad j \in \mathbb{N}_0 \tag{2.13}$$

verwendet werden. In dem Fall müsste das passende implizite Adams-Verfahren angewendet werden, da der Datensatz (2.13) einen zusätzlichen Interpolationspunkt berücksichtigt. Die Approximation berechnet sich in dem Fall durch

$$y_{n+1} \;=\; y_n + \tau \sum_{j=0}^{k} \gamma_j^{\mathrm{iA}} \nabla^j f_{n+1}^* \quad \text{mit} \quad \gamma_j^{\mathrm{iA}} := \int_0^1 (-1)^j \binom{-\theta+1}{j} \mathrm{d}\theta. \tag{2.14}$$

Wir bezeichnen die Predictor-Corrector-Verfahren, die einen Datensatz der Form (2.11) verwenden, als k-Schritt Predictor-Corrector-Verfahren erster Art (kurz: $\mathrm{PC}[1]_k$). Predictor-Corrector-Verfahren, die einen Datensatz der Form (2.13) verwenden, bezeichnen wir als $(k+1)$-Schritt Predictor-Corrector-Verfahren der zweiten Art (kurz: $\mathrm{PC}[2]_k$). Bei der Anwendung von Predictor-Corrector-Verfahren werden wir in unserer Arbeit stets die Art des verwendeten Verfahrens angeben und die Konstellation **PECE** nutzen. Die Funktionsweise der Predictor-Corrector-Verfahren wird in Abbildung 2.3 und 2.4 veranschaulicht.

Weitere nützliche Eigenschaften der vorgestellten Mehrschrittverfahren werden wir im nächsten Abschnitt diskutieren. Insbesondere werden wir uns auf die Eigenschaften konzentrieren, die im weiteren Verlauf dieser Arbeit von Bedeutung sind.

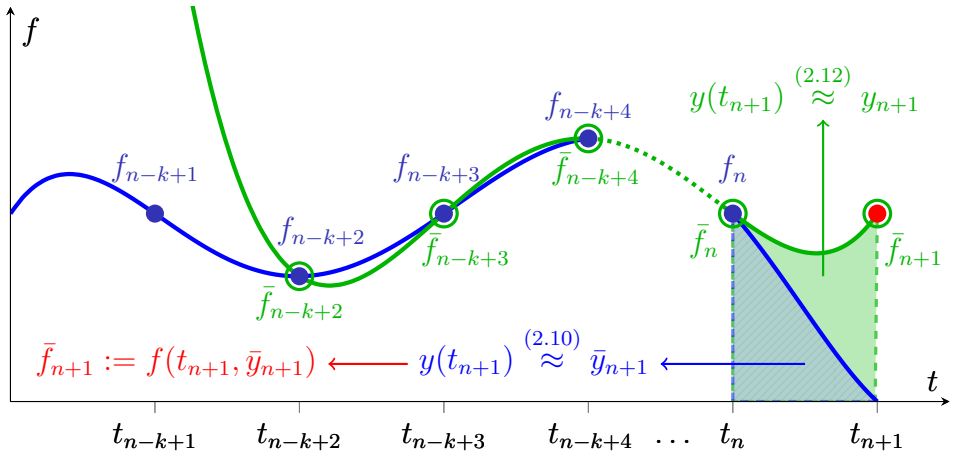

Abbildung 2.3: Funktionsweise des Predictor-Corrector-Verfahrens (**PECE**) erster Art

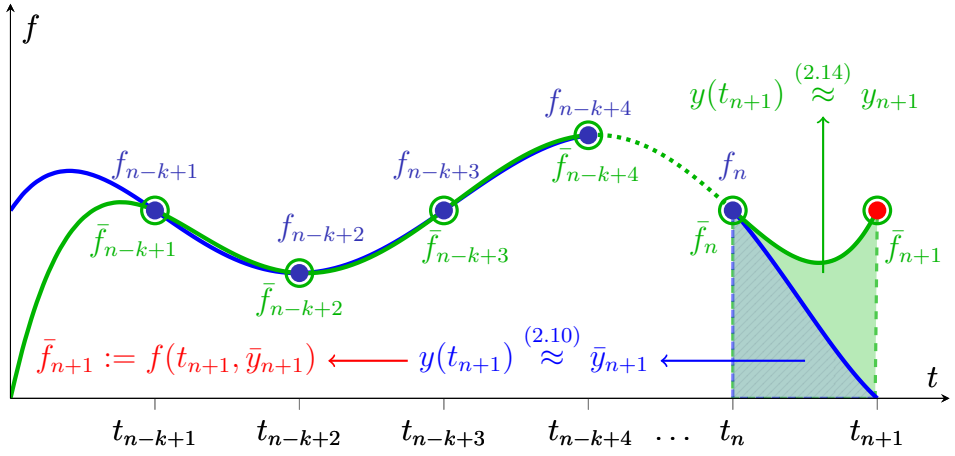

Abbildung 2.4: Funktionsweise des Predictor-Corrector-Verfahrens (**PECE**) zweiter Art

2.1.4 Konsistenz, Stabilität und Konvergenz

In diesem Abschnitt werden wir besonders wichtige und grundlegende Definitionen und Eigenschaften klassischer Mehrschrittverfahren wiedergeben. Hierfür ist es vorerst notwendig eine einheitliche Notation für die Mehrschrittverfahren einzuführen. Nach (2.6) und (2.9) können wir ein allgemeines Mehrschrittverfahren beschreiben durch

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = \tau \sum_{j=0}^{k} \beta_j f_{n+j}, \quad f_j = f(t_j, y_j), \quad t_j = t_0 + j\tau, \tag{2.15}$$

wobei die Parameter α_j, β_j für $j = 0, \ldots, k$ reell sind.

Es ist offensichtlich, dass $\alpha_k \neq 0$ gelten muss, um mit dem jeweiligen Verfahren eine Approximation zum nächsten Zeitschritt bestimmen zu können. Die Koeffizienten der Adams-Verfahren, die wir in der Arbeit behandeln, erfüllen zudem die folgenden Bedingungen:

$$\begin{aligned}
&\text{(explizite Verfahren):} \quad \alpha_k = 1, \quad \alpha_{k-1} = -1, \quad \alpha_j = 0, \quad 0 \leq j < k-2 \quad \text{und} \\
&\qquad\qquad\qquad\qquad \beta_j^{\mathrm{eA}} := \beta_j, \quad 0 \leq j < k-1, \quad \beta_k^{\mathrm{eA}} := \beta_k = 0, \\
&\text{(implizite Verfahren):} \quad \alpha_k = 1, \quad \alpha_{k-1} = -1, \quad \alpha_j = 0, \quad 0 \leq j < k-2 \quad \text{und} \\
&\qquad\qquad\qquad\qquad \beta_j^{\mathrm{iA}} := \beta_j, \quad 0 \leq j < k-1, \quad \beta_k^{\mathrm{iA}} := \beta_k \neq 0.
\end{aligned}$$

Wir beschränken uns im Folgenden auf die Definitionen und Aussagen, die eine Relevanz für diese Arbeit besitzen. Es sei in diesem Abschnitt vorausgesetzt, dass f (2.1) hinreichend glatt und Lipschitz-stetig ist mit einer Lipschitz-Konstante $\mathcal{L}_f \geq 0$.

Definition 2.1: *(lok. Fehler [24, Def. III.2.1, Lem. III.2.2], Ordnung [24, Def. III.2.3])*

- *Der **lokale Fehler** eines Mehrschrittverfahrens der Form (2.15) ist definiert als*

$$y(t_k) - y_k, \tag{2.16}$$

  wobei $y(t)$ der exakten Lösung von (2.1) und y_k der numerischen Approximation aus (2.15) mit exakten Startwerten $y_i = y(t_i)$ für $i = 0, \ldots, k-1$ entspricht.

- *Äquivalent lässt sich der **lokale Fehler** eines Mehrschrittverfahrens der Form (2.15) durch $L(y, t_0, \tau)$ beschreiben, wobei durch $L : \mathbb{R}^n \times \mathbb{R}_0^+ \times \mathbb{R}_0^+$ der Differenzenoperator*

$$L(y, t, \tau) := \sum_{j=0}^{k} \alpha_j y(t + j\tau) - \tau \sum_{j=0}^{k} \beta_j y'(t + j\tau)$$

  definiert ist.

- *Ein Mehrschrittverfahren (2.15) hat die **Konsistenzordnung** d, falls für hinreichend glatte Funktionen f (2.1) der lokale Fehler $\mathcal{O}(\tau^{d+1})$ ist.*

Die Konsistenzordnung ist über den lokalen Fehler definiert. Dahlquist konnte zur einfacheren Bestimmung der Konsistenzordnung eines Mehrschrittverfahrens Ordnungsbedingungen herleiten.

Satz 2.2: *(Ordnungsbedingungen [24, Thm. III 2.4])*

Ein Mehrschrittverfahren (2.15) besitzt die Konsistenzordnung d genau dann, wenn die Bedingungen

$$\sum_{j=0}^{k} \alpha_j = 0 \quad \textit{und} \quad \sum_{j=0}^{k} \alpha_j j^q = q \sum_{j=0}^{k} \beta_j j^{q-1} \quad \textit{für} \quad q = 1, \ldots, d \tag{2.17}$$

erfüllt sind.

Beweis: *(Satz 2.2)*
Eine Taylor-Entwicklung von $L(y, t_0, \tau)$ um $\tau = 0$ liefert bereits die Aussage. Wir führen den Beweis an dieser Stelle, weil wir im weiteren Verlauf auf diese Idee verweisen werden.

$$\begin{aligned}
L(y, t_0, \tau) &= \sum_{j=0}^{k} \alpha_j y(t_0 + j\tau) - \tau \sum_{j=0}^{k} \beta_j y'(t_0 + j\tau) \\
&= \sum_{j=0}^{k} \alpha_j \sum_{q=0}^{d} y^{(q)}(t_0) \frac{j^q}{q!} \tau^q - \tau \sum_{j=0}^{k} \beta_j \sum_{q=0}^{d-1} y^{(q+1)}(t_0) \frac{j^q}{q!} \tau^q + \mathcal{O}(\tau^{d+1}) \\
&= \sum_{j=0}^{k} \alpha_j \sum_{q=0}^{d} y^{(q)}(t_0) \frac{j^q}{q!} \tau^q - \sum_{j=0}^{k} \beta_j \sum_{q=1}^{d} y^{(q)}(t_0) \frac{j^{q-1}}{(q-1)!} \tau^q + \mathcal{O}(\tau^{d+1}) \\
&= y(t_0) \sum_{j=0}^{k} \alpha_j + \sum_{q=1}^{d} y^{(q)}(t_0) \frac{\tau^q}{q!} \left(\sum_{j=0}^{k} \alpha_j j^q - q \sum_{j=0}^{k} \beta_j j^{q-1} \right) + \mathcal{O}(\tau^{d+1}).
\end{aligned}$$

Sind die Ordnungsbedingungen erfüllt, so gilt demnach

$$L(y, t_0, \tau) = C_L \tau^{d+1} + \mathcal{O}(\tau^{d+2}) \quad \text{mit} \quad C_L := \frac{y^{(d+1)}(t_0)}{(d+1)!} \left(\sum_{j=0}^{k} \alpha_j j^{d+1} - (d+1) \sum_{j=0}^{k} \beta_j j^d \right).$$

□

Mit Hilfe der nachstehenden Definitionen können wir die Stabilität eines Zeitintegrators verstehen, die für die Konvergenz notwendig ist.

Definition 2.3: *(Stabilität, Stabilitätsbereich [31, Kap. 10,Def. 10.2])*

- *Ein numerischer Zeitintegrator heißt **stabil für** $\nu = \lambda\tau$, wenn der Integrator mit konstanter Schrittweite τ auf das Anfangswertproblem*

$$y' = \lambda y, \quad y(t_0) = y_0, \quad t_0 = 0 \tag{2.18}$$

  angewendet wird und die resultierende numerische Lösung $(y_n)_{n\in\mathbb{N}_0}$ beschränkt bleibt.

- *Das **Stabilitätsgebiet** eines numerischen Zeitintegrators ist definiert als*

$$\mathcal{S} := \{\nu \in \mathbb{C} \,|\, \nu = \tau\lambda \text{ und der numerische Zeitintegrator ist stabil für } \nu\}.$$

Betrachten wir zunächst die Anwendung eines allgemeinen Mehrschrittverfahrens (2.15) mit Schrittweite τ auf das Anfangswertproblem (2.18). Wir erhalten dadurch die Differenzengleichung

$$\sum_{j=0}^{k} \vartheta_j(\nu) y_{n+j} = 0 \quad \text{mit} \quad \nu = \tau\lambda \quad \text{und} \quad \vartheta_j(\nu) := \alpha_j - \nu\beta_j. \tag{2.19}$$

Für die Stabilität eines Mehrschrittverfahrens benötigen wir die Beschränktheit der Lösungen von (2.19).

Satz 2.4: *(Wurzelkriterium [31, Kap.9, Satz 9.6])*

Die folgenden Aussagen sind äquivalent:

(i) *Jede Lösung $(y_n)_{n\in\mathbb{N}_0}$ von (2.19) ist für $n \to \infty$ beschränkt.*

(ii) *Für die Nullstellen ζ_i mit $i = 1, \ldots, k$ des Polynoms*

$$\sum_{j=0}^{k} \vartheta_j(\nu)\zeta^j \quad \textit{gilt} \quad |\zeta_i| \leq 1, \quad i = 1, \ldots, k,$$

wobei Gleichheit nur für einfache Nullstellen erfüllt sein darf.

Nach dem Wurzelkriterium folgt aus der Definition 2.3, dass das Stabilitätsgebiet $\mathcal{S}_M$ eines Mehrschrittverfahrens wie folgt beschrieben werden kann

$$\mathcal{S}_M = \left\{ \nu \in \mathbb{C} \;\middle|\; \text{Die Nullstellen des Polynoms } \sum_{j=0}^{k} \vartheta_j(\nu)\zeta^j \text{ erfüllen das Wurzelkriterium.} \right\}.$$

Für die Anwendung eignen sich nur Mehrschrittverfahren mit $0 \in \mathcal{S}_M$ (0-Stabilität), da die Konvergenztheorie für $\tau \to 0$ beschränkte Lösungen erfordert. Zur graphischen Darstellung der Stabilitätsgebiete von Mehrschrittverfahren werden sogenannte Root-Locus-Curves [25, V.1, S. 240-245] verwendet. Diese Kurven lassen sich mit Hilfe der Differenzengleichung (2.19) ermitteln, indem die Differenzengleichung nach der Variable ν aufgelöst wird und die Terme y_{n+j} durch ζ^{n+j} substituiert werden. Anschließend wird ν als Funktion in der Variable ζ definiert und auf dem Einheitskreis, d.h. für $\zeta = e^{\mathrm{i}\theta}$ mit $\theta \in [0, 2\pi[$, ausgewertet.

Beispiele 2.5: (*Root-Locus-Curves*)

(i) Adams-Verfahren (explizit oder implizit):
Die Root-Locus-Curves zu den Adams-Verfahren erhalten wir aus der Darstellungsform (2.15) der Mehrschrittverfahren

$$\nu : \mathbb{E} \to \mathbb{C} \quad \text{mit} \quad \zeta \mapsto \frac{\zeta^k - \zeta^{k-1}}{\sum_{j=0}^{k} \beta_j \zeta^j}, \qquad \mathbb{E} := \left\{ x \in \mathbb{C} \;\middle|\; x = e^{\mathrm{i}\theta}, \theta \in [0, 2\pi[\right\}.$$

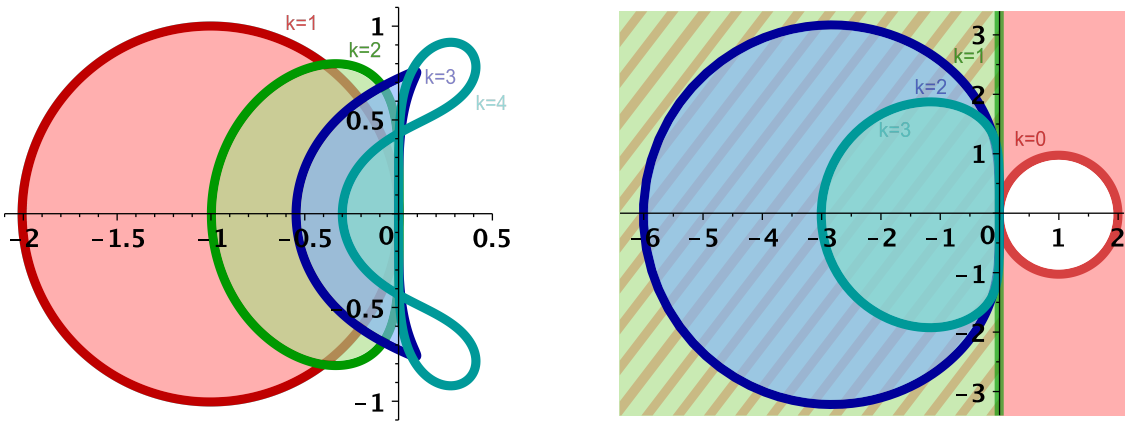

Abbildung 2.5: Stabilitätsgebiete der expliziten (links) und impliziten Adams-Verfahren (rechts).

(ii) Predictor-Corrector-Verfahren (erster Art):
Um die Root-Locus-Curves dieser Verfahren zu ermitteln, müssen wir die Vorschrift vom Prediction-Prozess in den Correction-Prozess einsetzen.

$$\begin{aligned} y_{n+k} &= y_{n+k-1} + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{iA}} f_{n+1+j}^* \\ &= y_{n+k-1} + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{iA}} f_{n+1+j} + \tau \beta_{k-1}^{\mathrm{iA}} (\bar{f}_{n+k} - f_{n+k}) \\ &= y_{n+k-1} + \nu \sum_{j=0}^{k-1} \beta_j^{\mathrm{iA}} y_{n+1+j} + \nu \beta_{k-1}^{\mathrm{iA}} (y_{n+k-1} - y_{n+k}) + \nu^2 \beta_{k-1}^{\mathrm{iA}} \sum_{j=0}^{k-1} \beta_j^{\mathrm{eA}} y_{n+j} \end{aligned}$$

Die Substitution von y_{n+j} durch ζ^j für $j \geq 0$ führt in diesem Fall zu einer quadratischen Gleichung $A\nu^2 + B\nu + C = 0$ mit den Koeffizienten

$$A := \beta_{k-1}^{\mathrm{iA}} \sum_{j=0}^{k-1} \beta_j^{\mathrm{eA}} \zeta^j, \qquad B := \sum_{j=0}^{k-1} \beta_j^{\mathrm{iA}} \zeta^{j+1} + \beta_{k-1}^{\mathrm{iA}} (\zeta^{k-1} - \zeta^k),$$
$$C := \zeta^{k-1} - \zeta^k.$$

Die Lösungen dieser quadratischen Gleichung liefern die Root-Locus-Curves, die gemeinsam das Stabilitätsgebiet des jeweiligen Predictor-Corrector-Verfahrens beschreiben

$$\nu_1, \nu_2 : \mathbb{E} \to \mathbb{C} \quad \text{mit} \quad \zeta \mapsto \frac{-B + \sqrt{B^2 - 4AC}}{2C}, \quad \zeta \mapsto \frac{-B - \sqrt{B^2 - 4AC}}{2C}.$$

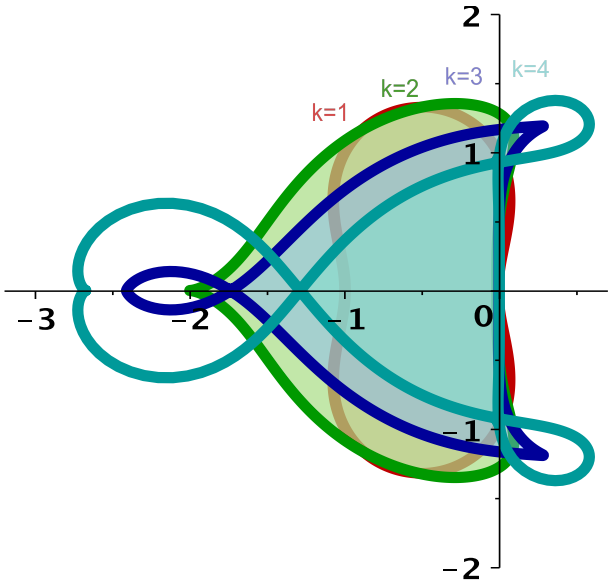

Abbildung 2.6: Stabilitätsgebiete der Predictor-Corrector-Verfahren erster Art.

(iii) Predictor-Corrector-Verfahren (zweiter Art):
Die Root-Locus-Curves zu Predictor-Corrector-Verfahren der zweiten Art lassen sich prinzipiell wie für die Predictor-Corrector-Verfahren der ersten Art ermitteln. Der einzige Unterschied besteht in den Koeffizienten der quadratischen Gleichung, die in diesem Fall durch

$$A := \beta_k^{\mathrm{iA}} \sum_{j=0}^{k-1} \beta_j^{\mathrm{eA}} \zeta^j, \qquad B := \sum_{j=0}^{k} \beta_j^{\mathrm{iA}} \zeta^j + \beta_k^{\mathrm{iA}} (\zeta^{k-1} - \zeta^k),$$
$$C := \zeta^{k-1} - \zeta^k$$

gegeben sind.

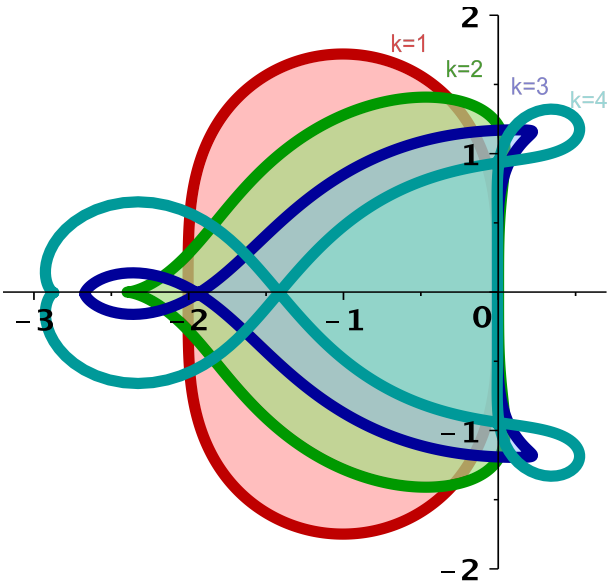

Abbildung 2.7: Stabilitätsgebiete der Predictor-Corrector-Verfahren zweiter Art.

Aus den Abbildungen 2.5, 2.6 und 2.7 können wir erkennen, dass die Predictor-Corrector-Verfahren deutlich größere Stabilitätsgebiete vorweisen als die expliziten Adams-Verfahren. In vielen Fällen führt dies dazu, dass bei Verwendung von Predictor-Corrector-Verfahren die Schrittweite mehr als doppelt so groß gewählt werden kann. Obwohl der rechnerische Aufwand bei diesen Verfahren doppelt so hoch ist, sind diese Verfahren sehr interessant, da der Mehraufwand durch die Wahl einer deutlich größeren Schrittweite mehr als kompensiert werden kann. Natürlich ist dieser Vorteil abhängig vom gestellten Problem, weil die Verteilung der Eigenwerte diesbezüglich von entscheidender Bedeutung ist.

Nachdem wir die Konsistenz und Stabilität von Mehrschrittverfahren verstanden haben, wenden wir uns jetzt der Konvergenz eines Mehrschrittverfahrens zu. Die Konvergenz wird über den globalen Fehler definiert. Neben der Tatsache, dass ein Verfahren überhaupt konvergiert, ist es ebenso wichtig zu wissen, wie akkurat die numerische Approximation ist, die mit Hilfe des Zeitintegrators berechnet wurde.

Definition 2.6: *(globaler Fehler, Konvergenzordnung [31, Kap. 9, Def. 9.8])*

- *Der **globale Fehler** eines Mehrschrittverfahrens (2.15) ist definiert durch die folgende Differenz*

$$\|y(t_n) - y_n\|, \quad t_n \in [t_0, T], \quad n \geq 0.$$

- *Ein Mehrschrittverfahren (2.15) ist konvergent der Ordnung d, falls der globale Fehler eine Abschätzung der folgenden Form*

$$\|y(t_n) - y_n\| \leq C\tau^d, \quad t_n \in [t_0, T], \quad n \geq 0$$

erfüllt, wobei C unabhängig ist von n und τ. Zudem müssen hinreichend genaue Startwerte vorliegen

$$\|y(t_j) - y_j\| \leq \bar{C}\tau^d \quad \textit{für} \quad j = 0, \ldots, k-1.$$

*Die Zahl d wird hierbei als die **Konvergenzordnung** definiert.*

Viel bedeutender als die Konsistenz ist die Konvergenz eines Zeitintegrators, daher wird für lineare Anfangswertprobleme oft der folgende Leitsatz formuliert:

KONSISTENZ + STABILITÄT = KONVERGENZ

Mit der Definition 2.6 lässt sich der „Leitsatz“ für Mehrschrittverfahren wie folgt beschreiben.

Satz 2.7: *(Allgemeine Konvergenz von Mehrschrittverfahren [31, Kap.9, Satz 9.9])*

Ein 0-stabiles Mehrschrittverfahren mit Konsistenzordnung d besitzt die Konvergenzordnung d.

Mit den Sätzen aus diesem Abschnitt können wir unmittelbar das folgende Korollar folgern, mit dessen Formulierung wir diesen Abschnitt auch abschließen werden.

Korollar 2.8: *(Konvergenz von Adams- und Predictor-Corrector-Verfahren)*

a) *Ein stabiles explizites k-Schritt Adams-Verfahren besitzt die Konvergenzordnung k mit $k \geq 1$.*

b) *Ein stabiles implizites $(k+1)$-Schritt Adams-Verfahren besitzt die Konvergenzordnung $k+1$ mit $k \geq 0$.*

c) *Ein stabiles k-Schritt Predictor-Corrector-Verfahren der ersten Art besitzt die Konvergenzordnung k mit $k \geq 1$.*

d) *Ein stabiles $(k+1)$-Schritt Predictor-Corrector-Verfahren der zweiten Art besitzt die Konvergenzordnung $k+1$ mit $k \geq 1$.*

Beweis: *(Korollar 2.8)*
Nur die Bestimmung der Konsistenzordnungen ist notwendig, denn die Stabilität ist nach Voraussetzung gegeben. Die Anwendung von Satz 2.7 liefert uns dann jeweils die Aussage.

a) Nach Konstruktion löst ein explizites k-Schritt Adams-Verfahren die Anfangswertprobleme

$$y' = qt^{q-1}, \quad y(0) = 0 \quad \text{für} \quad q = 1, \ldots, k$$

exakt. Aus dem Grund ist dies äquivalent zu

$$0 = L(t^q, 0, \tau) = \tau^q \left(\sum_{j=0}^{k} \alpha_j j^q - q \sum_{j=0}^{k} \beta_j j^{q-1} \right) \quad \text{für} \quad q = 1, \ldots, k.$$

Ein explizites k-Schritt Adams-Verfahren erfüllt die Ordnungsbedingungen aus Satz 2.2 für $q = 1, \ldots, k$ und ist damit ein Verfahren der Konsistenzordnung k.

b) Nach Konstruktion löst ein implizites Adams-Verfahren mit $k \geq 0$ die Anfangswertprobleme

$$y' = qt^{q-1}, \quad y(0) = 0 \quad \text{für} \quad q = 1, \ldots, k+1$$

exakt. Analog zu Teil a) ergibt sich die Aussage.

c) Für den lokalen Fehler der Approximation aus dem Prediction-Prozess gilt

$$\|y(t_k) - \bar{y}_k\| \leq C\tau^{k+1} \quad \text{mit} \quad C = C(C_L).$$

Die Approximation aus dem Correction-Prozess kann beschrieben werden durch

$$y_k = y_{k-1} + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{iA}} f_{j+1}^{\circ} + \tau \beta_{k-1}^{\mathrm{iA}} \big(\bar{f}_k - f(t_k, y(t_k)) \big)$$

$$\text{mit} \quad \bar{f}_k := f(t_k, \bar{y}_k) \quad \text{und} \quad f_{j+1}^{\circ} = \begin{cases} f_{j+1}, & j \neq k-1 \\ f(t_{j+1}, y(t_{j+1})), & j = k-1 \end{cases}$$

$$\Longrightarrow \quad \|y(t_k) - y_k\| \leq \left\| y(t_k) - y_{k-1} - \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{iA}} f_{j+1}^{\circ} \right\| + \tau \beta_{k-1}^{\mathrm{iA}} \mathcal{L}_f \, \|\bar{y}_k - y(t_k)\| \,,$$

wobei $\mathcal{L}_f$ der Lipschitz-Konstante von f entspricht. Für den lokalen Fehler des Predictor-Corrector-Verfahrens erhalten wir dann

$$L^{\mathrm{PC}}(y, t_0, \tau) \leq \underbrace{L^{\mathrm{iA}}(y, t_0, \tau)}_{\leq C\tau^{k+1}} + \tau \beta_{k-1}^{\mathrm{iA}} \mathcal{L}_f \underbrace{L^{\mathrm{eA}}(y, t_0, \tau)}_{\leq C\tau^{k+1}} \leq C\tau^{k+1}$$

mit $C = C(C_L, \beta_{k-1}^{\mathrm{iA}}, \mathcal{L}_f)$.

d) Analog zu c) folgt das Resultat aus

$$L^{\mathrm{PC}}(y, t_0, \tau) \leq \underbrace{L^{\mathrm{iA}}(y, t_0, \tau)}_{\leq C\tau^{k+2}} + \underbrace{\tau \beta_k^{\mathrm{iA}} \mathcal{L}_f L^{\mathrm{eA}}(y, t_0, \tau)}_{\leq C\tau^{k+2}} \leq C\tau^{k+2}$$

mit $C = C(C_L, \beta_k^{\mathrm{iA}}, \mathcal{L}_f)$. □

2.2 Exponentielle Mehrschrittverfahren

In diesem Abschnitt werden wir anhand von [33] und [34] die Konstruktion exponentieller Mehrschrittverfahren präsentieren. Der wesentliche Unterschied zwischen exponentiellen und klassischen Mehrschrittverfahren liegt in der Verwendung einer alternativen Darstellung der exakten Lösung des Anfangswertproblems (2.1), die anhand der Variation der Konstanten Formel (kurz: VdK-Formel) hergeleitet werden kann. Bei der Anwendung der VdK-Formel wird die Darstellung der rechten Seite im Vergleich zum klassischen Ansatz (2.2) ausgenutzt. Zudem kann die VdK-Formel selbst bei nichtlinearen Problemen angewendet werden.

2.2.1 Explizite exponentielle Adams-Verfahren

Ähnlich wie bei klassischen Mehrschrittverfahren beginnen wir mit der Konstruktion von expliziten exponentiellen Adams-Verfahren. Hierzu betrachten wir erneut das Anfangswertproblem (2.1), das durch

$$y' = Ay + g(t,y), \quad y(t_0) = y_0$$

gegeben ist. Die Anwendung der VdK-Formel auf dieses Anfangswertproblem liefert

$$\begin{aligned} y(t) &= e^{(t-t_0)A}y_0 + \int_{t_0}^{t} e^{(t-s)A} g(s,y(s))\mathrm{d}s \\ &= e^{(t-t_0)A}y_0 + (t-t_0)\int_0^1 e^{(1-\theta)(t-t_0)A} g\Big(t_0+\theta(t-t_0), y\big(t_0+\theta(t-t_0)\big)\Big)\mathrm{d}\theta \end{aligned} \tag{2.20}$$

für die exakte Lösung. Im Gegensatz zum klassischen Fall wird bei exponentiellen Mehrschrittverfahren nur die nichtlineare Funktion g im Integranden polynomiell approximiert. Das Interpolationspolynom zur Funktion g wird analog zum klassischen Fall mit Hilfe eines Datensatzes der Form

$$(t_{n-k+1}, g_{n-k+1}), \ldots, (t_n, g_n) \quad \text{mit} \quad g_j := g\big(t_j, y_j\big) \quad \text{und} \quad t_j = t_0 + j\tau, \quad j \in \mathbb{N}_0$$

konstruiert. Das Polynom lässt sich nach der Newton'schen Interpolationsformel durch

$$p_n(t_n+\theta\tau) = \sum_{j=0}^{k-1} (-1)^j \binom{-\theta}{j} \nabla^j g_n \tag{2.21}$$

beschreiben. Die Approximation von g in (2.20) durch ein Polynom p_n liefert uns dann die folgende Beziehung

$$\begin{array}{ccccc} y(t_n+\tau) & = & e^{\tau A}y(t_n) & + \tau \displaystyle\int_0^1 & e^{(1-\theta)\tau A} g\Big(t_n+\theta\tau, y\big(t_n+\theta\tau\big)\Big)\mathrm{d}\theta \\ \wr\!\wr & & \wr\!\wr & & \wr\!\wr \\ y_{n+1} & = & e^{\tau A}y_n & + \tau \displaystyle\int_0^1 & e^{(1-\theta)\tau A} p_n(t_n+\theta\tau)\,\mathrm{d}\theta. \end{array} \tag{2.22}$$

Eine wichtige Bemerkung an dieser Stelle ist, dass jedes numerische Verfahren, das einer Beziehung der Form (2.22) genügt, mit Hilfe von sogenannten φ-Funktionen

$$\varphi_0(x) := e^x, \quad \varphi_j(x) := \int_0^1 e^{(1-\theta)x} \frac{\theta^{j-1}}{(j-1)!} \mathrm{d}\theta\,, \quad j \in \mathbb{N} \tag{2.23}$$

berechnet werden kann, denn jedes Polynom lässt sich durch eine Linearkombination von Monomen darstellen. Setzen wir dementsprechend das Polynom (2.21) in (2.22) ein, so ergibt sich die Vorschrift der expliziten exponentiellen Adams-Verfahren

$$y_{n+1} = e^{\tau A} y_n + \tau \sum_{j=0}^{k-1} \gamma_j^{\mathrm{eA}}(\tau A) \nabla^j g_n \quad \text{mit} \quad \gamma_j^{\mathrm{eA}}(x) := \int_0^1 e^{(1-\theta)x} (-1)^j \binom{-\theta}{j} \mathrm{d}\theta. \tag{2.24}$$

Die numerische Approximation in (2.24) erfordert die Berechnung von Matrix-Funktionen bzw. die Berechnung von Produkten zwischen Matrix-Funktionen und Vektoren. Die Schwierigkeiten, die sich durch solche Berechnungen stellen, werden wir später diskutieren. Graphisch lassen sich die expliziten exponentiellen Adams-Verfahren ähnlich zu den klassischen expliziten Adams-Verfahren beschreiben. Der Unterschied liegt zum einen am Polynom p_n, dass in diesem Fall nur die nichtlineare Funktion g approximiert. Zum anderen stellt die Fläche zwischen dem Graphen des Polynoms und der Zeitachse nur den Bereich dar, in dem die Berechnungen stattfinden.

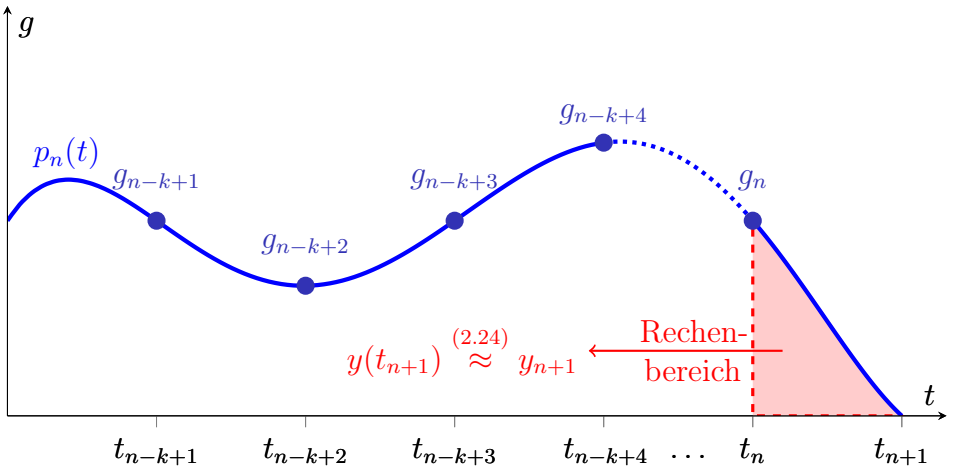

Abbildung 2.8: Graphische Darstellung der expl. exponentiellen Adams-Verfahren

Für diese Verfahren können wir die Funktionen γ_j^{eA} mit Hilfe von (2.23) wie folgt beschreiben

$$\begin{aligned}
\gamma_0^{\mathrm{eA}}(x) &= \varphi_1(x)\\
\gamma_1^{\mathrm{eA}}(x) &= \varphi_2(x)\\
\gamma_2^{\mathrm{eA}}(x) &= \tfrac{1}{2}\varphi_2(x) + \varphi_3(x)\\
\gamma_3^{\mathrm{eA}}(x) &= \tfrac{1}{3}\varphi_2(x) + \varphi_3(x) + \varphi_4(x)\\
\gamma_4^{\mathrm{eA}}(x) &= \tfrac{1}{4}\varphi_2(x) + \tfrac{11}{12}\varphi_3(x) + \tfrac{3}{2}\varphi_4(x) + \varphi_5(x).
\end{aligned}$$

Bezeichnen wir mit $k \in \mathbb{N}$ auch hier die Anzahl der explizit verwendeten Interpolationspunkte, so können wir das zugehörige explizite exponentielle k-Schritt Adams-Verfahren angeben. Im Folgenden sind schließlich die expliziten exponentiellen Adams-Verfahren für $k = 1, 2, 3$ aufgeführt

$$\begin{aligned}
k = 1: \qquad & y_{n+1} = e^{\tau A} y_n + \tau\varphi_1(\tau A) g_n\\
k = 2: \qquad & y_{n+1} = e^{\tau A} y_n + \tau\Big(\big(\varphi_1(\tau A) + \varphi_2(\tau A)\big) g_n - \varphi_2(\tau A) g_{n-1}\Big)\\
k = 3: \qquad & y_{n+1} = e^{\tau A} y_n + \tau\Big(\big(\varphi_1(\tau A) + \tfrac{3}{2}\varphi_2(\tau A) + \varphi_3(\tau A)\big) g_n\\
& \qquad - \big(2\varphi_2(\tau A) + 2\varphi_3(\tau A)\big) g_{n-1} + \big(\tfrac{1}{2}\varphi_2(\tau A) + \varphi_3(\tau A)\big) g_{n-2}\Big).
\end{aligned} \tag{2.25}$$

Aus der Darstellung (2.25) erhalten wir mit $A = 0$ und $\varphi_j(0) = \frac{1}{j!}$ die klassischen expliziten Adams-Verfahren (2.6).

2.2.2 Implizite exponentielle Adams-Verfahren

Der Unterschied zwischen impliziten und expliziten exponentiellen Mehrschrittverfahren ist wie im klassischen Fall. Denn neben den k explizit vorliegenden Interpolationspunkten wird ein zusätzlicher impliziter Interpolationspunkt für die Konstruktion der polynomialen Approximation p_n von g verwendet. Dies entspricht der Verwendung eines Datensatzes der Form

$$(t_{n-k+1}, g_{n-k+1}), \ldots, (t_{n+1}, g_{n+1}) \quad \text{mit} \quad g_j := g\big(t_j, y_j\big) \quad \text{und} \quad t_j = t_0 + j\tau, \quad j \in \mathbb{N}_0.$$

Mit der Newton'schen Interpolationsformel erhalten wir mit

$$p_n(t_n + \theta\tau) = \sum_{j=0}^{k} (-1)^j \binom{-\theta+1}{j} \nabla^j g_{n+1} \tag{2.26}$$

eine mathematische Beschreibung des Interpolationspolynoms zur Funktion g. Wie im expliziten Fall erhalten wir durch Einsetzen von (2.26) in (2.22) die Vorschrift der impliziten exponentiellen Adams-Verfahren

$$y_{n+1} = e^{\tau A} y_n + \tau \sum_{j=0}^{k} \gamma_j^{\mathrm{iA}}(\tau A) \nabla^j g_{n+1}, \quad \gamma_j^{\mathrm{iA}}(x) := \int_0^1 e^{(1-\theta)x} (-1)^j \binom{-\theta+1}{j} \,\mathrm{d}\theta. \tag{2.27}$$

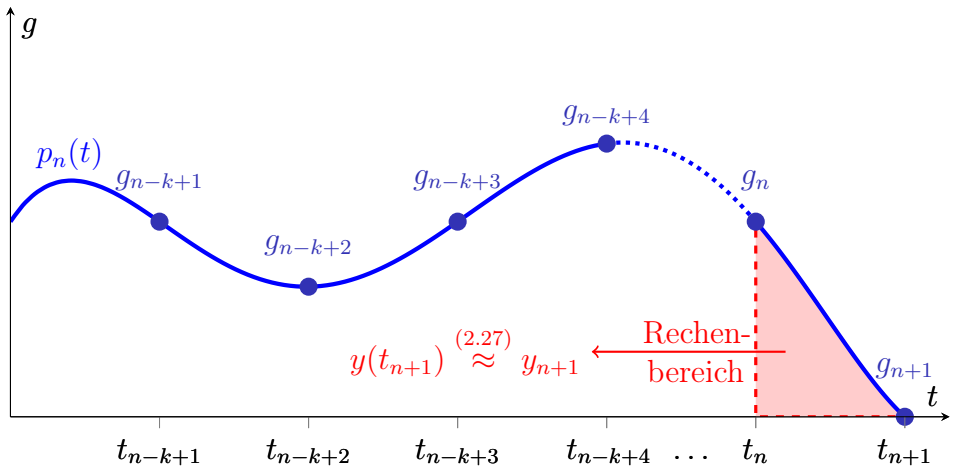

Abbildung 2.9: Graphische Darstellung der impl. exponentiellen Adams-Verfahren

Aus den Abbildungen 2.8 und 2.9 wird auch deutlich, dass die expliziten und impliziten exponentiellen Adams-Verfahren sich nur in dem Datensatz unterscheiden, der für die Konstruktion der polynomialen Approximation p_n von g verwendet wird.

Schreiben wir die Matrixfunktionen γ_j^{iA} als Linearkombinationen von φ-Funktionen

$$\begin{aligned}
\gamma_0^{\mathrm{iA}}(x) &= \varphi_1(x)\\
\gamma_1^{\mathrm{iA}}(x) &= \varphi_2(x) - \varphi_1(x)\\
\gamma_2^{\mathrm{iA}}(x) &= -\tfrac{1}{2}\varphi_2(x) + \varphi_3(x)\\
\gamma_3^{\mathrm{iA}}(x) &= -\tfrac{1}{6}\varphi_2(x) + \varphi_4(x)\\
\gamma_4^{\mathrm{iA}}(x) &= -\tfrac{1}{12}\varphi_2(x) - \tfrac{1}{12}\varphi_3(x) + \tfrac{1}{2}\varphi_4(x) + \varphi_5(x)
\end{aligned}$$

und bezeichnen wir mit $k \in \mathbb{N}_0$ die Anzahl der explizit verwendeten Interpolationspunkte, so können wir die impliziten exponentiellen Adams-Verfahren für $k = 0, 1, 2$ angeben

$$\begin{aligned}
k = 0: \qquad & y_{n+1} = e^{\tau A} y_n + \tau\varphi_1(\tau A) g_{n+1}\\
k = 1: \qquad & y_{n+1} = e^{\tau A} y_n + \tau\Big(\varphi_2(\tau A) g_{n+1} + \big(\varphi_1(\tau A) - \varphi_2(\tau A)\big) g_n\Big)\\
k = 2: \qquad & y_{n+1} = e^{\tau A} y_n + \tau\Big(\big(\tfrac{1}{2}\varphi_2(\tau A) + \varphi_3(\tau A)\big) g_{n+1}\\
& \qquad + \big(\varphi_1(\tau A) - 2\varphi_3(\tau A)\big) g_n + \big(-\tfrac{1}{2}\varphi_2(\tau A) + \varphi_3(\tau A)\big) g_{n-1}\Big).
\end{aligned} \tag{2.28}$$

Wie im expliziten Fall erhalten wir aus (2.28) die Vorschriften der klassischen impliziten Adams-Verfahren (2.9), wenn wir $A = 0$ setzen und die φ-Funktionen auswerten.

2.2.3 Exponentielle Predictor-Corrector-Verfahren

Nachdem wir die Konstruktion von expliziten und impliziten exponentiellen Adams-Verfahren verstanden haben, können wir mit Hilfe dieser Verfahren exponentielle Predictor-Corrector-Verfahren konstruieren. Die Idee im exponentiellen Fall ist identisch zum klassischen Fall. Aus dem Grund werden wir die exponentiellen Predictor-Corrector-Verfahren ebenfalls mit den Rechenprozessen Prediction, Evaluation und Correction beschreiben.

Prediction: (P)
Die Verwendung eines expliziten exponentiellen Adams-Verfahrens mit k-Schritten (2.24) liefert uns die Approximation als „Vorhersage“

$$y(t_{n+1}) \overset{(2.22)}{\underset{(2.24)}{\approx}} \bar{y}_{n+1} = e^{\tau A} y_n + \tau \sum_{j=0}^{k-1} \gamma_j^{\mathrm{eA}}(\tau A) \nabla^j g_n. \tag{2.29}$$

Evaluation: (E)
Mit Hilfe der Vorhersage (2.29) wird die Funktion g ausgewertet, um eine Approximation für $g\big(t_{n+1}, y(t_{n+1})\big)$ zu bestimmen

$$g\big(t_{n+1}, y(t_{n+1})\big) \approx g(t_{n+1}, \bar{y}_{n+1}) =: \bar{g}_{n+1}.$$

Correction: (C)
Im Correction-Prozess wird ein implizites exponentielles Adams-Verfahren angewendet. Für einen exponentiellen Predictor-Corrector-Verfahren der ersten Art wird ein Datensatz der Form

$$(t_{n-k+2}, g^*_{n-k+2}), \ldots, (t_{n+1}, g^*_{n+1}) \quad \text{mit} \quad g^*_j := \begin{cases} g_j, & j \neq n+1 \\ \bar{g}_j, & j = n+1 \end{cases}, \quad j \in \mathbb{N}_0 \tag{2.30}$$

verwendet. Die Approximation des Predictor-Corrector-Verfahrens, können wir dann insgesamt durch

$$y(t_{n+1}) \overset{(2.22)}{\underset{(2.27)}{\approx}} y_{n+1} = e^{\tau A} y_n + \tau \sum_{j=0}^{k-1} \gamma_j^{\mathrm{iA}}(\tau A) \nabla^j g^*_{n+1} \tag{2.31}$$

beschreiben.

Zur Konstruktion von exponentiellen Predictor-Corrector-Verfahren der zweiten Art wird

$$(t_{n-k+1}, g^*_{n-k+1}), \ldots, (t_{n+1}, g^*_{n+1}) \quad \text{mit} \quad g^*_j := \begin{cases} g_j, & j \neq n+1 \\ \bar{g}_j, & j = n+1 \end{cases}, \quad j \in \mathbb{N}_0 \tag{2.32}$$

statt (2.30) als Datensatz verwendet. Die Darstellung der Approximation im Correction-Prozess verändert sich dadurch zu

$$y(t_{n+1}) \overset{(2.22)}{\underset{(2.27)}{\approx}} y_{n+1} = e^{\tau A} y_n + \tau \sum_{j=0}^{k} \gamma_j^{\mathrm{iA}}(\tau A) \nabla^j g^*_{n+1}. \tag{2.33}$$

Wie bei klassischen Predictor-Corrector-Verfahren werden wir auch bei exponentiellen Predictor- Corrector-Verfahren stets die Art angeben. Zur Vollständigkeit haben wir auch die Konstruktion dieser Verfahren mit den Abbildungen 2.10 und 2.11 beschrieben.

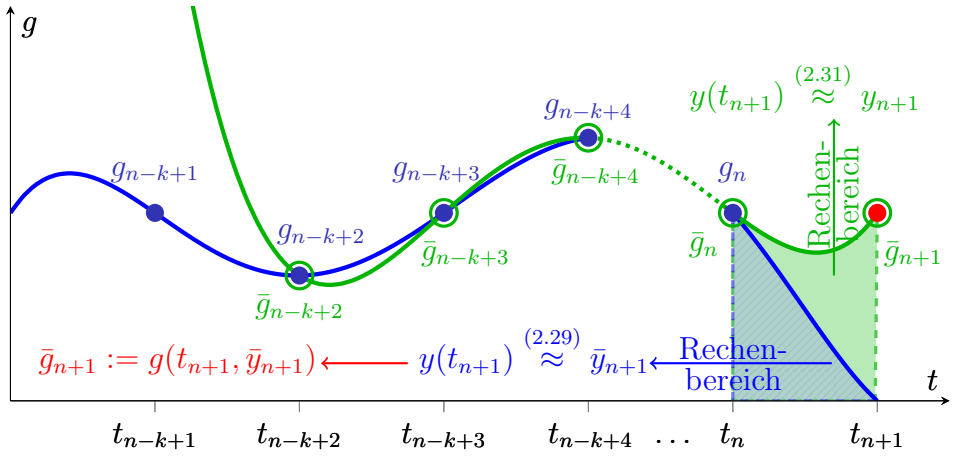

Abbildung 2.10: Funktionsweise des exponentiellen Pred.-Corr.-Verfahrens (**PECE**) 1. Art

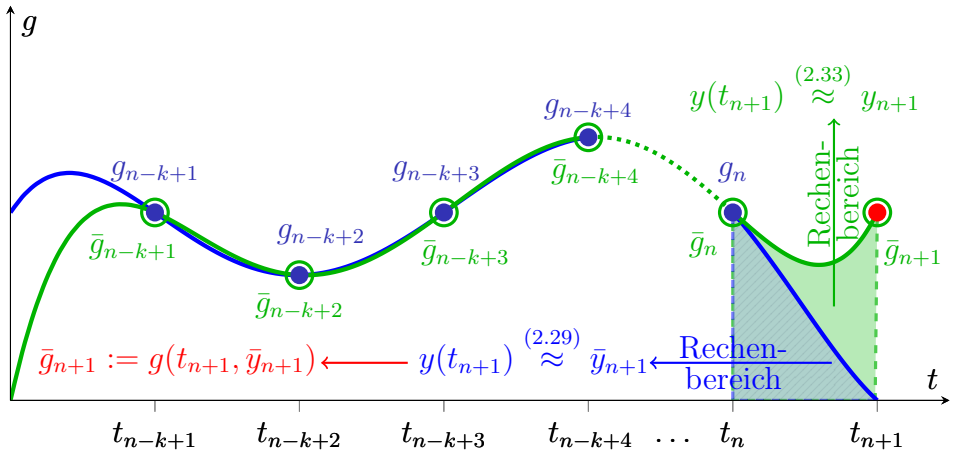

Abbildung 2.11: Funktionsweise des exponentiellen Pred.-Corr.-Verfahrens (**PECE**) 2. Art

2.2.4 Allgemeine exponentielle Mehrschrittverfahren

In diesem Abschnitt werden wir eine allgemeinere Klasse von exponentiellen Mehrschrittverfahren vorstellen. Sowohl die klassischen als auch die bisher vorgestellten exponentiellen Adams-Verfahren verwenden ausschließlich das Interpolationspolynom zur polynomialen Approximation der rechten Seite bzw. des nichtlinearen Anteils. Wir möchten die polynomiale Approximation zu einem gegebenen Datensatz und die damit zugrunde liegende Verfahrensklasse verallgemeinern.

Die Interpolationspolynome (2.21) und (2.26) lassen sich auf die folgende Form bringen (vgl. (2.25) und (2.28))

$$\text{(explizit)} \quad p_n(t_n + \theta\tau) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} b_{i,j}^{\mathrm{eM}} \frac{\theta^i}{i!} g_{n-k+1+j}, \quad b_{i,j}^{\mathrm{eM}} \in \mathbb{R}, \quad k \in \mathbb{N}, \tag{2.34}$$

$$\text{(implizit)} \quad p_n(t_n + \theta\tau) = \sum_{i=0}^{k} \sum_{j=0}^{k} b_{i,j}^{\mathrm{iM}} \frac{\theta^i}{i!} g_{n-k+1+j}, \quad b_{i,j}^{\mathrm{iM}} \in \mathbb{R}, \quad k \in \mathbb{N}_0. \tag{2.35}$$

Die Binomialkoeffizienten in (2.21) und (2.26) entsprechen jeweils einem Polynom in der Variable θ und die Rückwärtsdifferenzen stellen jeweils eine Linearkombination des zugehörigen Datensatzes dar. Mit der Verwendung von (2.34) und (2.35) mit beliebigen reellen Koeffizienten als polynomiale Approximation zur Funktion g beschränken wir uns nicht weiter auf die Verwendung der Interpolationspolynome. Zudem lassen wir die Summe über i von 0 bis $d-1$ laufen, um den verwendeten Polynomgrad kontrollieren zu können

$$\text{(explizit)} \quad p_n(t_n + \theta\tau) = \sum_{i=0}^{d-1} \sum_{j=0}^{k-1} b_{i,j}^{\mathrm{eM}} \frac{\theta^i}{i!} g_{n-k+1+j}, \quad d \leq k, \quad d \in \mathbb{N} \tag{2.36}$$

$$\text{(implizit)} \quad p_n(t_n + \theta\tau) = \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j}^{\mathrm{iM}} \frac{\theta^i}{i!} g_{n-k+1+j}, \quad d \leq k+1, \quad d \in \mathbb{N}. \tag{2.37}$$

Damit ist es uns möglich eine allgemeinere Klasse von exponentiellen Mehrschrittverfahren zu konstruieren. Die Verfahrensvorschriften der exponentiellen Mehrschrittverfahren

$$\text{(explizit)} \quad y_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d-1} \sum_{j=0}^{k-1} b_{i,j}^{\mathrm{eM}} \varphi_{i+1}(\tau A) g_{n-k+1+j}, \tag{2.38}$$

$$\text{(implizit)} \quad y_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j}^{\mathrm{iM}} \varphi_{i+1}(\tau A) g_{n-k+1+j}, \tag{2.39}$$

lassen sich durch Einsetzen von (2.36) und (2.37) in (2.22) herleiten. Ebenso können wir jede Art von exponentiellem Predictor-Corrector-Verfahren konstruieren, indem wir sowohl im Prediction- als auch im Correction-Prozess ein Verfahren dieser Verfahrensklasse verwenden.

Prediction: (P)
Die Verwendung eines expliziten exponentiellen k-Schrittverfahrens (2.38) liefert uns für $d_{\mathrm{P}} \leq k$ die Approximation

$$y(t_{n+1}) \underset{(2.38)}{\overset{(2.22)}{\approx}} \bar{y}_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d_{\mathrm{P}}-1} \sum_{j=0}^{k-1} b_{i,j}^{\mathrm{eM}} \varphi_{i+1}(\tau A) g_{n-k+1+j}. \tag{2.40}$$

Evaluation: (E)
Die Auswertung der Funktion g mit (2.40) liefert uns die Approximation

$$g\big(t_{n+1}, y(t_{n+1})\big) \approx g(t_{n+1}, \bar{y}_{n+1}) =: \bar{g}_{n+1}.$$

Correction: (C)
Analog zu Abschnitt 2.2.3 verwenden wir in diesem Prozess ein implizites exponentielles Mehrschrittverfahren. Die Approximation eines Predictor-Corrector-Verfahrens der ersten Art zum Datensatz (2.30) können wir schließlich durch

$$y(t_{n+1}) \underset{(2.39)}{\overset{(2.22)}{\approx}} y_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d_{\mathrm{C}}-1} \sum_{j=1}^{k} b_{i,j}^{\mathrm{iM}} \varphi_{i+1}(\tau A) g_{n-k+1+j}^{*} \tag{2.41}$$

mit $d_{\mathrm{C}} \leq k$ beschreiben.

Wird hingegen im Correction-Prozess ein Datensatz der Form (2.32) verwendet, so ergibt sich für $d_{\mathrm{C}} \leq k+1$ die folgende Approximation

$$y(t_{n+1}) \underset{(2.39)}{\overset{(2.22)}{\approx}} y_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d_{\mathrm{C}}-1} \sum_{j=0}^{k} b_{i,j}^{\mathrm{iM}} \varphi_{i+1}(\tau A) g_{n-k+1+j}^{*}. \tag{2.42}$$

2.2.5 Konsistenz, Stabilität und Konvergenz

In diesem Abschnitt werden wir wichtige Eigenschaften exponentieller Mehrschrittverfahren verstehen. Wie im klassischen Fall beginnen wir mit einer einheitlichen Notation, die die vorgestellten exponentiellen Mehrschrittverfahren umfasst. Nach (2.38) und (2.39) können wir ein allgemeines exponentielles Mehrschrittverfahren durch

$$y_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \varphi_{i+1}(\tau A) g_{n-k+1+j} = e^{\tau A} y_n + \tau \sum_{j=0}^{k} \beta_j(\tau A) g_{n-k+1+j} \quad (2.43)$$

beschreiben, wobei $\beta_j(x) := \sum_{i=0}^{d-1} b_{i,j} \varphi_{i+1}(x)$ gilt. Die Koeffizienten dieser exponentiellen Mehrschrittverfahren genügen für $0 \leq i \leq d-1$ und $0 \leq j \leq k$ den folgenden Bedingungen

$$\begin{aligned}
&\text{(explizite Verfahren):} \quad && b_{i,j}^{\mathrm{eM}} := b_{i,j}, \quad b_{i,k}^{\mathrm{eM}} = 0, \quad \text{bzw.} \\
&&& \beta_j^{\mathrm{eM}}(\tau A) := \beta_j(\tau A), \quad \beta_k^{\mathrm{eM}}(\tau A) = 0, \\
&\text{(implizite Verfahren):} \quad && b_{i,j}^{\mathrm{iM}} := b_{i,j}, \quad \exists\, 0 \leq i \leq d-1 : \; b_{i,k}^{\mathrm{iM}} \neq 0 \quad \text{bzw.} \\
&&& \beta_j^{\mathrm{iM}}(\tau A) := \beta_j(\tau A), \quad \beta_k^{\mathrm{iM}}(\tau A) \neq 0.
\end{aligned}$$

Ein allgemeines Mehrschrittverfahren (2.43) ist eindeutig durch eine Koeffizientenmatrix $\mathcal{B}$ bestimmt. Explizit können die Koeffizientenmatrizen eines allgemeinen expliziten und impliziten Mehrschrittverfahrens durch

$$\mathcal{B}_k^{\mathrm{eM}} := \begin{pmatrix} b_{0,0}^{\mathrm{eM}} & \cdots & b_{0,k-1}^{\mathrm{eM}} \\ \vdots & \vdots & \vdots \\ b_{d-1,0}^{\mathrm{eM}} & \cdots & b_{d-1,k-1}^{\mathrm{eM}} \end{pmatrix} \quad \text{und} \quad \mathcal{B}_k^{\mathrm{iM}} := \begin{pmatrix} b_{0,0}^{\mathrm{iM}} & \cdots & b_{0,k}^{\mathrm{iM}} \\ \vdots & \vdots & \vdots \\ b_{d-1,0}^{\mathrm{iM}} & \cdots & b_{d-1,k}^{\mathrm{iM}} \end{pmatrix} \quad (2.44)$$

beschrieben werden. Die Fehleranalyse exponentieller Integratoren beruht auf der Halbgruppen-Theorie. Diese funktionalanalytische Herangehensweise bietet den entscheidenden Vorteil, eine von der Diskretisierung unabhängige Fehleranalyse durchzuführen. In dieser Arbeit werden wir nur die nötigen Voraussetzungen der Halbgruppen-Theorie nach [33] und [34] vornehmen und kurz erläutern, um die Aussagen in einer mathematisch sinnvollen Art beschreiben zu können. Für weitere Details verweisen wir auf die Quellen [17], [26] und [47].

Voraussetzung 2.9: *(Allgemeine Konfiguration [34, Ass. 2.2])*

Sei X ein Banach-Raum mit der Norm $\|\cdot\|$ und $A : X \to X$ ein linearer Operator auf X. Zudem sei A der infinitesimale Erzeuger einer stark stetigen Halbgruppe $e^{\tau A}$ auf X.

Voraussetzung 2.10: *(Konfiguration für parabolische Probleme [34, Ass. 2.9, Ass. 2.10])*

a) *Sei X ein Banach-Raum mit der Norm $\|\cdot\|$ und $A : X \to X$ ein linearer Operator auf X. Zudem sei A der infinitesimale Erzeuger einer analytischen Halbgruppe $e^{\tau A}$ auf X.*

b) *Das Spektrum $\sigma(A) \subset \mathbb{C}^-$ sei von Null weg beschränkt. Für $0 \leq \alpha < 1$ sei V ein Banach-Raum mit*

$$V = \{v \in X | (-A)^\alpha v \in X\} = \mathcal{D}\big((-A)^\alpha\big)$$

und der Norm $\|v\|_V = \|(-A)^\alpha v\|$. Des Weiteren sei die Abbildung $g : [0,T] \times V \to X$ lokal Lipschitz-stetig mit $\mathcal{L}_g > 0$ in einer Umgebung der exakten Lösung, d.h. für alle $v, w \in V$ nahe der exakten Lösung gilt

$$\|g(t,v) - g(t,w)\| \leq \mathcal{L}_g \|v - w\|_V \quad \textit{für} \quad t \in [0,T].$$

Diese Voraussetzungen möchten wir kurz mit den folgenden Bemerkungen erläutern.

Bemerkungen 2.11:

1) Die Voraussetzung 2.9 impliziert eine Abschätzung von $\|\varphi_j(\tau A)\|$ für $j \in \mathbb{N}_0$ mit $\varphi_0(\tau A) := e^{\tau A}$ und zwar existieren für $0 \leq \tau \leq T$ Konstanten $C_A > 0$ und $\omega \in \mathbb{R}$, so dass

$$\begin{aligned} \|e^{\tau A}\| &\leq C_A e^{\omega \tau} \\ \text{und} \quad \|\varphi_j(\tau A)\| &\overset{(2.23)}{\leq} \tfrac{1}{j!} C_A e^{\omega \tau} \end{aligned} \tag{2.45}$$

gelten. Diese Voraussetzung stellt für semilineare Probleme aus numerischer Perspektive eine Minimalvoraussetzung dar, denn bezüglich der Beispiele in der Anwendung werden damit alle relevanten Problemstellungen abgedeckt, in der der Wertebereich des linearen Operators in der Halbebene mit $\mathrm{Re} z \leq \omega$ enthalten ist.

2) Die Voraussetzung 2.10 a) ist für parabolische Probleme mit sektoriellen Operatoren von Interesse und stellt eine stärkere Voraussetzung dar als 2.9. Sie impliziert für ein $\sigma \geq 0$, dass $(-A + \sigma I)^\alpha$ mit $\alpha \in [0,1)$ wohldefiniert ist. Mit einer äquivalenten Umformulierung von (2.1) kann ein semilineares Problem mit linearem Operator $\widetilde{A} = A - \sigma I$ bestimmt werden. Der lineare Operator $\widetilde{A}$ würde die Voraussetzung 2.10 a) erfüllen. Gleichzeitig wären die Ausdrücke $(-\widetilde{A} + \sigma I)^\alpha$ mit $\alpha \in [0,1)$ für $\sigma = 0$ wohldefiniert. Aus dem Grund setzen wir ohne Einschränkung voraus, dass $(-A)^\alpha$ mit $\alpha \in [0,1)$ wohldefiniert ist. Des Weiteren gilt unter der Voraussetzung 2.10 a) die sogenannte parabolische Glättung

$$\|e^{\tau A}\| + \|\tau^\gamma (-A)^\gamma e^{\tau A}\| \leq C_P, \quad \gamma, \tau \geq 0. \tag{2.46}$$

3) Aus (2.46) ergeben sich nützliche Abschätzungen. Zum einen gilt

$$\|(-A)^{\alpha}\varphi_j(\tau A)\| \leq C\tau^{-\alpha} \quad \text{mit} \quad C = C(C_P) \tag{2.47}$$

und zum anderen

$$\|(-A)^{\alpha}\beta_j(\tau A)\| \leq \sum_{i=0}^{d-1} |b_{i,j}| \, \|(-A)^{\alpha}\varphi_{i+1}(\tau A)\| \leq C\tau^{-\alpha} \tag{2.48}$$

mit $C = (C_P, \mathcal{B}, d)$.

Die Konvergenzbeweise werden wir für parabolische Probleme 2.10 führen. Für hyperbolische Probleme ist diese Voraussetzung nicht verwendbar, denn in diesem Fall liegt kein sektorieller Operator vor. Allerdings werden wir im weiteren Verlauf hyperbolische Beispiele betrachten und in diesem Fall sind wir auf die Voraussetzung 2.9 angewiesen. Allerdings sind folgende zusätzliche Voraussetzungen notwendig, damit die Beweise auch in diesen Fällen ihre Gültigkeit bewahren.

Voraussetzung 2.12: *(Konfiguration für hyperbolische Probleme)*

a) *Die Voraussetzung 2.9 sei erfüllt. Zudem gelte (2.45) mit $C_A > 0$ und $\omega = 0$.*

b) *Die lokale Lipschitz-Bedingung aus der Voraussetzung 2.10 b) sei mit $\alpha = 0$ und $V = X$ erfüllt.*

Nachdem wir die nötigen Vorkehrungen getroffen haben, können wir uns den Eigenschaften der exponentiellen Mehrschrittverfahren widmen. Wie im klassischen Fall beginnen wir mit der Konsistenzordnung. Diese wird, genau wie im klassischen Fall, über den lokalen Fehler definiert. Ebenso können wir anhand der Taylor-Entwicklung Ordnungsbedingungen für die exponentiellen Mehrschrittverfahren herleiten.

Satz 2.13: *(Ordnungsbedingungen)*

Sei $G(t) := g(t, y(t)) \in C^d([0,T])$, so besitzt ein exponentielles k-Schrittverfahren (2.43) genau dann die Konsistenzordnung d, falls die Ordnungsbedingungen

$$\sum_{j=0}^{k} \frac{b_{i,j}(j+1-k)^l}{l!} = \delta_{l,i} \quad \textit{mit} \quad i = 0, \ldots, d-1 \quad \textit{und} \quad \delta_{l,i} = \begin{cases} 1, & l = i \\ 0, & l \neq i \end{cases}$$

für $l = 0, \ldots, d-1$ erfüllt sind.

Beweis: *(Satz 2.13)*
Die exakte Lösung zur Zeit t_k ist nach (2.20) gegeben durch

$$y(t_k) = e^{\tau A} y(t_{k-1}) + \tau \int_0^1 e^{(1-\theta)\tau A} G(t_{k-1} + \theta\tau) \mathrm{d}\theta. \tag{2.49}$$

Mithilfe der φ-Funktionen (2.23) und der Taylor-Entwicklung der Funktion G um t_{k-1} können wir die exakte Lösung zur Zeit t_k wie folgt darstellen

$$\begin{aligned} y(t_k) = \; & e^{\tau A} y(t_{k-1}) + \tau \sum_{l=0}^{d-1} \int_0^1 e^{(1-\theta)\tau A} G^{(l)}(t_{k-1}) \frac{(\theta\tau)^l}{l!} \mathrm{d}\theta \\ & + \tau \int_0^1 e^{(1-\theta)\tau A} G^{(d)}(\xi) \frac{(\theta\tau)^d}{d!} \mathrm{d}\theta \\ = \; & e^{\tau A} y(t_{k-1}) + \sum_{l=0}^{d-1} \tau^{l+1} \varphi_{l+1}(\tau A) G^{(l)}(t_{k-1}) + \tau^{d+1} \varphi_{d+1}(\tau A) G^{(d)}(\xi). \end{aligned} \tag{2.50}$$

Zur Abschätzung des lokalen Fehlers bestimmen wir die Taylor-Entwicklung der numerischen Approximation zu den exakten Startwerten

$$\big(t_j, G_j\big) \quad \text{mit} \quad G_j := G(t_j) \quad \text{und} \qquad j = 0, \ldots, k-1$$

und erhalten hierfür die folgende Darstellung

$$\begin{aligned} y_k = \; & e^{\tau A} y(t_{k-1}) + \tau \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \varphi_{i+1}(\tau A) G(t_j) \\ = \; & e^{\tau A} y(t_{k-1}) + \sum_{l=0}^{d-1} \tau^{l+1} \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \frac{(j+1-k)^l}{l!} \varphi_{i+1}(\tau A) G^{(l)}(t_{k-1}) \\ & + \tau^{d+1} \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \varphi_{i+1}(\tau A) \frac{(j+1-k)^d}{d!} G^{(d)}(\xi_2). \end{aligned} \tag{2.51}$$

Definieren wir mit $e_k := y(t_k) - y_k$ den lokalen Fehler, so ergibt sich aus (2.50) und (2.51)

$$\begin{aligned} e_k = \; & \sum_{l=0}^{d-1} \tau^{l+1} \left(\varphi_{l+1}(\tau A) - \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \frac{(j+1-k)^l}{l!} \varphi_{i+1}(\tau A) \right) G^{(l)}(t_{k-1}) \\ & + \tau^{d+1} \left(\varphi_{d+1}(\tau A) G^{(d)}(\xi_1) - \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \varphi_{i+1}(\tau A) \frac{(j+1-k)^d}{d!} G^{(d)}(\xi_2) \right). \end{aligned} \tag{2.52}$$

Der führende Term liefert die gewünschten Ordnungsbedingungen aufgrund der linearen Unabhängigkeit der φ-Funktionen

$$\varphi_{l+1}(x) = \sum_{i=0}^{d-1} \sum_{j=0}^{k} \frac{b_{i,j}(j+1-k)^l}{l!} \varphi_{i+1}(x), \quad \text{für alle} \quad x \in \mathbb{C}$$

$$\iff \qquad \delta_{l,i} = \sum_{j=0}^{k} \frac{b_{i,j}(j+1-k)^l}{l!} \quad \text{mit} \quad i = 0, \ldots, d-1\,. \qquad \square$$

Bemerkung: 2.14:

Für hinreichend glatte Funktionen lässt sich der lokale Fehler eines exponentiellen Mehrschrittverfahrens der Konsistenzordnung d durch

$$\|e_k\| \leq C_O \tau^{d+1} \quad \text{mit} \quad C_O = C_O(C_A, G^{(d)}(t_{k-1}), \mathcal{B}, d, k) \tag{2.53}$$

abschätzen.

In Anlehnung an die klassischen Mehrschrittverfahren fahren wir mit der Stabilität von exponentiellen Mehrschrittverfahren fort. Beylkin, Keiser und Vozovoi [5] haben diesbezüglich einen ersten Schritt geleistet, indem Sie zur Bestimmung von Stabilitätsgebieten exponentieller Mehrschrittverfahren eine alternative Testgleichung verwendet haben. Diese Testgleichung wurde zuvor von Karniadakis, Israeli und Orszag [37] für die Stabilitätsanalyse von IMEX-Verfahren verwendet. Einen ähnlichen Ansatz verfolgten Hundsdorfer und Verwer [36] bei der Stabilitätsanalyse von „IMEX-θ-Verfahren".

Folgen wir der Stabilitätsanalyse aus [5], so können wir die Stabilität eines exponentiellen Mehrschrittverfahren für skalare Gleichungen wie folgt definieren.

Definition 2.15: *(Stabilität exponentieller Mehrschrittverfahren)*

- *Ein exponentielles Mehrschrittverfahren (2.43) heißt* ***stabil für*** $\nu := \tau\lambda$ ***in Abhängigkeit von*** $a := -\tau A$*, wenn der Integrator mit konstanter Schrittweite* τ *auf das skalare Anfangswertproblem*

$$y' = Ay + \lambda y = Ay + g(y), \quad y(t_0) = y_0, \quad t_0 = 0, \quad A \leq 0 \tag{2.54}$$

angewendet wird und die resultierende numerische Lösung $(y_n)_{n \in \mathbb{N}_0}$ *beschränkt bleibt.*

· *Die **Stabilität** eines exponentiellen Mehrschrittverfahrens wird durch die Familie von Mengen* $(\mathcal{S}_a)_{a\in\mathbb{R}_0^+}$ *mit*

$$\mathcal{S}_a := \{\nu \in \mathbb{C} \mid \nu = \tau\lambda \text{ und (2.43) ist stabil für } \nu \text{ in Abhängigkeit von } a = -\tau A\}$$

beschrieben.

Wir betonen nochmals, dass dieser Stabilitätsbegriff nur für skalare Gleichungen relevant ist. Denn betrachten wir (2.1) mit $g(y) = By$ und $A, B \in \mathbb{R}^{n\times n}$, so kommutieren A und B in der Regel nicht, d.h. nur mit Hilfe der Kommutativität können wir das semilineare Problem auf skalare Gleichungen zurückführen.

Für die graphische Darstellung von $(\mathcal{S}_a)_{a\in\mathbb{R}_0^+}$ können wir die Root-Locus-Curves verwenden, denn für ein fixiertes $a \in \mathbb{R}_0^+$ ergibt sich eine Differenzengleichung für die wir das Wurzelkriterium aus Satz 2.4 anwenden können. In Beispiel 2.16 beschreiben wir dies für exponentielle Mehrschrittverfahren und deren Predictor-Corrector Varianten.

Beispiel 2.16: *(Root-Locus-Curves exponentieller Mehrschrittverfahren)*

(i) Exponentielle Mehrschrittverfahren (explizit und implizit):
Aus der Darstellung (2.43) ergibt sich

$$\nu_a : \mathbb{E} \to \mathbb{C} \quad \text{mit} \quad \zeta \mapsto \frac{\zeta^k - e^{-a}\zeta^{k-1}}{\sum_{i=0}^{k} \beta_i(-a)\zeta^i}, \qquad \mathbb{E} := \left\{ x \in \mathbb{C} \,\middle|\, x = e^{\mathrm{i}\theta}, \theta \in [0, 2\pi[\right\}.$$

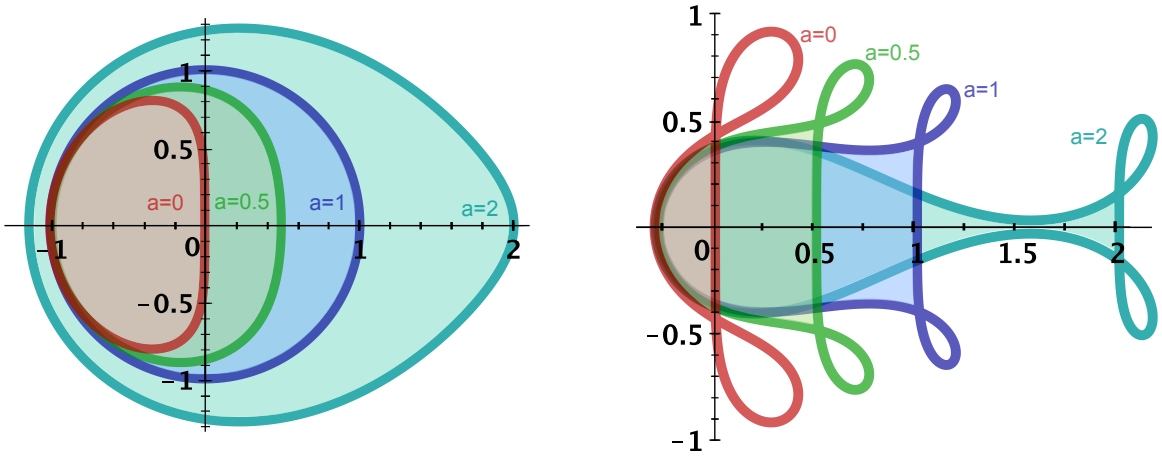

Abbildung 2.12: Stabilitätskurven $\mathcal{S}_a$ für explizite exponentielle Adams-Verfahren. Links für $k = 2$ und rechts für $k = 4$.

(ii) Predictor-Corrector-Verfahren (erster und zweiter Art):
Für exponentielle Predictor-Corrector-Verfahren erhalten wir wie in Beispiel 2.5 eine quadratische Gleichung $A\nu_a^2 + B\nu_a + C = 0$, deren Lösungen die Stabilitätskurven beschreiben. Die Koeffizienten A, B und C sind im exponentiellen Fall wie folgt definiert:

$$\text{Erster Art:} \quad A := \beta_{k-1}^{\mathrm{iM}}(-a)\sum_{j=0}^{k-1}\beta_j^{\mathrm{eM}}(-a)\zeta^j$$

$$B := \sum_{j=0}^{k-1}\beta_j^{\mathrm{iM}}(-a)\zeta^{j+1} + \beta_{k-1}^{\mathrm{iM}}(-a)(e^{-a}\zeta^{k-1} - \zeta^k)$$

$$C := e^{-a}\zeta^{k-1} - \zeta^k$$

$$\text{Zweiter Art:} \quad A := \beta_k^{\mathrm{iM}}(-a)\sum_{j=0}^{k-1}\beta_j^{\mathrm{eM}}(-a)\zeta^j$$

$$B := \sum_{j=0}^{k}\beta_j^{\mathrm{iM}}(-a)\zeta^j + \beta_k^{\mathrm{iM}}(-a)(e^{-a}\zeta^{k-1} - \zeta^k)$$

$$C := e^{-a}\zeta^{k-1} - \zeta^k.$$

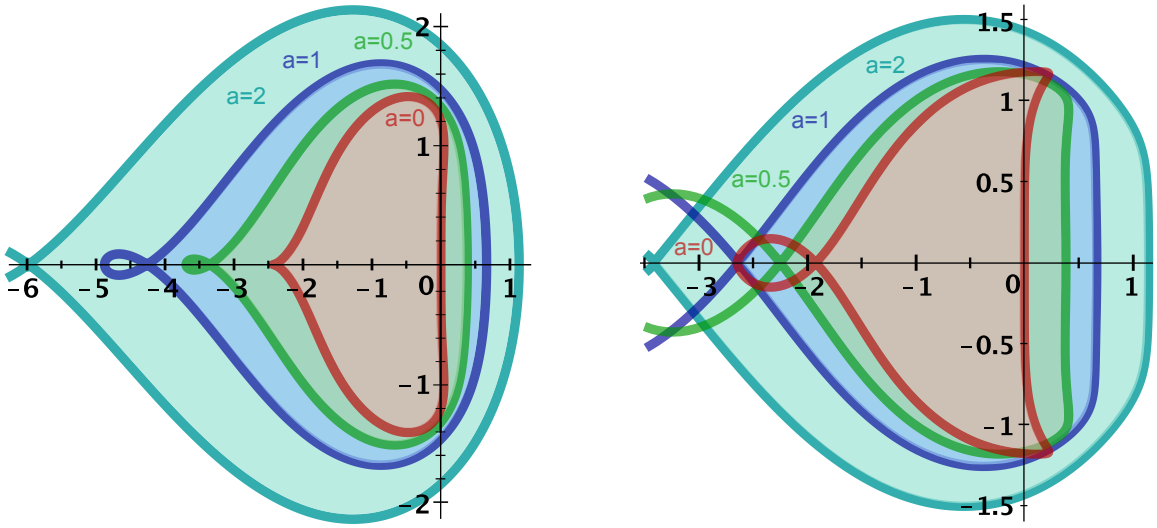

Abbildung 2.13: Stabilitätskurven $\mathcal{S}_a$ für exponentielle Pred.-Corr.-Verfahren zweiter Art. Links für $k = 2$ und rechts für $k = 3$.

Betrachten wir das Testproblem (2.54), so ist der lineare Teil, der Teil der Differentialgleichung, der „exakt“ integriert wird und der Parameter a in der Stabilitätsanalyse stellt eine Art „Maß“ für die Steifigkeit des linearen Anteils dar. Aus den Abbildungen 2.12 und 2.13 können wir erkennen, dass exponentielle Mehrschrittverfahren für semilineare Probleme eine recht gute Eigenschaft besitzen. Die Schrittweitenwahl hängt in diesen Fällen im Grunde genommen vom nichtsteifen nichtlinearen Anteil ab. Vergleichen wir die Stabilitätskonturen S_a so stellen wir fest, dass die der exponentiellen Predictor-Corrector-Verfahren sich in Abhängigkeit von a deutlich stärker vergrößern, als für explizite exponentielle Adams-Verfahren. Generell sind exponentielle Integratoren [34] für semilineare Probleme konstruiert worden, bei denen die Steifigkeit im linearen Anteil enthalten ist. Mithilfe der Halbgruppen-Theorie ist für diese Integratoren eine Konvergenzanalyse möglich, die von der räumlichen Diskretisierung unabhängig ist.

Schließlich widmen wir uns der Konvergenz von exponentiellen Mehrschrittverfahren. Die Fehleranalyse von Hochbruck und Ostermann in [33] ist in dieser Hinsicht von maßgebender Bedeutung. Hierfür werden wir zwei wichtige Abschätzungen vorstellen, die wir für die Fehleranalyse benötigen. Zum einen benötigen wir Gronwall-Lemmata, die in vielen Konvergenzbeweisen eine besondere Rolle spielen:

Lemma 2.17: *(Gronwall-Lemma [34, Lem. 2.15])*

Seien $\tau, T > 0$ zwei positive reelle Zahlen mit $0 \leq t_n := n\tau \leq T$ und sei $(\varepsilon_n)_{n\in\mathbb{N}}$ eine Folge, die die Ungleichung

$$\varepsilon_n \leq a\tau \sum_{i=1}^{n-1} t_{n-i}^{-\rho}\varepsilon_i + bt_n^{-\sigma} \quad \textit{mit} \quad 0 \leq \rho, \sigma < 1 \quad \textit{und} \quad a, b \geq 0$$

erfüllt, so genügen die Folgenglieder der Folge $(\varepsilon_n)_{n\in\mathbb{N}}$ der Abschätzung

$$\varepsilon_n \leq Cbt_n^{-\sigma},$$

wobei die Konstante C von ρ, σ, a und T abhängig ist.

Zum anderen benötigen wir eine polynomiale Abschätzung, um die Fehleranalyse aus [33] für allgemeine explizite Mehrschrittverfahren führen zu können. Wir zeigen im folgenden Lemma, dass Polynome der Form (2.36) und (2.37), deren Koeffizienten die Ordnungsbedingungen aus Satz 2.13 erfüllen, eine polynomiale Approximation des nichtlinearen Anteils darstellen.

Lemma 2.18:

Sei eine hinreichend glatte Funktion $G(t) := g(t, y(t)) \in C^d([0,T])$ *gegeben und sei* $\tilde{p}$ *ein Polynom der Form*

$$\tilde{p}(t_n + \theta\tau) := \sum_{i=0}^{d-1} \sum_{j=0}^{k-1} b_{i,j} \frac{\theta^i}{i!} G(t_{n-k+1+j}) \quad mit \quad k \geq d.$$

Sind zudem die Ordnungsbedingungen aus Satz 2.13 erfüllt, so gilt für $\theta \in [0,1]$ *und* $[0, t_{n+1}] \subset [0,T]$

$$\|G(t_n + \theta\tau) - \tilde{p}(t_n + \theta\tau)\| \leq C_{\tilde{P}} \tau^d \sup_{0 \leq t \leq t_{n+1}} \left\|G^{(d)}(t)\right\| \quad mit \quad C_{\tilde{P}} = C_{\tilde{P}}(k, d, \mathcal{B}).$$

Beweis: *(Lemma 2.18)*
Mit der Taylor-Entwicklung von G um t_n ergibt sich die folgende Darstellung für die Differenz

$$G(t_n + \theta\tau) - \tilde{p}(t_n + \theta\tau) = \sum_{l=0}^{d-1} \tau^l \underbrace{\left(\frac{\theta^l}{l!} - \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} \frac{b_{i,j}(1+j-k)^l}{l!} \frac{\theta^i}{i!} \right)}_{\overset{\text{Satz 2.13}}{=} 0} G^{(l)}(t_n) + R_d$$

mit dem Restterm

$$\begin{aligned} R_d := & \int_{t_n}^{t_n+\theta\tau} \frac{(t_n + \theta\tau - t)^{d-1}}{(d-1)!} G^{(d)}(t) \mathrm{d}t \\ & - \sum_{i=0}^{d-1} \sum_{j=0}^{k-1} b_{i,j} \frac{\theta^i}{i!} \int_{t_n}^{t_{n+1+j-k}} \frac{(t_{n+1+j-k} - t)^{d-1}}{(d-1)!} G^{(d)}(t) \mathrm{d}t. \end{aligned}$$

Mithilfe einer Abschätzung ergibt sich aus

$$\|R_d\| \leq C_{\tilde{P}} \tau^d \sup_{0 \leq t \leq t_{n+1}} \left\|G^{(d)}(t)\right\|$$

die Behauptung. □

Nun folgt das Resultat zur Konvergenz von exponentiellen Mehrschrittverfahren, das sich mit dem Lemma von Gronwall 2.17 und dem Lemma 2.18 beweisen lässt.

Satz 2.19: *(Konvergenz von exponentiellen Mehrschrittverfahren)*

Das Anfangswertproblem (2.1) erfülle die Voraussetzungen 2.10. Zudem sei die Funktion G mit $G(t) := g\big(t, y(t)\big)$ hinreichend glatt mit $G \in C^d([0,T], X)$ und die Schrittweite τ sei beschränkt durch eine hinreichend kleine Konstante $\mathcal{T}$ mit $0 < \tau \leq \mathcal{T}$. Betrachten wir unter diesen Voraussetzungen die numerische Lösung eines expliziten exponentiellen Mehrschrittverfahrens (2.43) der Konsistenzordnung d mit hinreichend genauen Startwerten

$$\|y_j - y(t_j)\|_V \leq c_0 \tau^d, \quad j = 1, \ldots, k-1, \quad k \geq d, \tag{2.55}$$

zur Schrittweite τ, so ist der Fehler

$$\|y_n - y(t_n)\|_V \leq C\tau^d \quad \textit{mit} \quad C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g, T, c_0, \alpha)$$

gleichmäßig beschränkt für $0 \leq n\tau \leq T$.

Beweis: *(Satz 2.19)*
Wir definieren $\tilde{p}_n$ als das Polynom (2.36)

$$\tilde{p}_n(t_n + \theta\tau) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} b_{i,j}^{\mathrm{eM}} \frac{\theta^i}{i!} G_{n-k+1+j}. \tag{2.56}$$

zu den exakten Daten $\big(t_j, G_j\big)$ mit $n-k+1 \leq j \leq n$, wobei $t_j := j\tau$ und $G_j := G(t_j)$ gilt. Damit lässt sich die exakte Lösung nach der VdK-Formel (2.20) durch

$$\begin{aligned} y(t_{n+1}) &= e^{\tau A} y(t_n) + \tau \int_0^1 e^{(1-\theta)\tau A} \tilde{p}_n(t_n + \theta\tau) \mathrm{d}\theta + \delta_{n+1} \\ &= e^{\tau A} y(t_n) + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau A) G_{n-k+1+j} + \delta_{n+1}. \end{aligned}$$

beschreiben, wobei δ_{n+1} mit

$$\delta_{n+1} := \tau \int_0^1 e^{(1-\theta)\tau A} \big(G(t_n + \theta\tau) - \tilde{p}_n(t_n + \theta\tau)\big) \mathrm{d}\theta$$

den Defekt bzw. den lokalen Fehler charakterisiert. Den Defekt können wir mit Hilfe von

Voraussetzung 2.10 und Lemma 2.18 wie folgt abschätzen

$$\begin{aligned}\|\delta_{n+1}\| &= \left\| \tau \int_0^1 e^{(1-\theta)\tau A}\big(G(t_n+\theta\tau) - \tilde{p}_n(t_n+\theta\tau)\big)\mathrm{d}\theta \right\| \\ &\leq \tau \int_0^1 \left\| e^{(1-\theta)\tau A} \right\| \cdot \|G(t_n+\theta\tau) - \tilde{p}_n(t_n+\theta\tau)\| \,\mathrm{d}\theta \\ &\leq C\tau^{d+1} \sup_{0\leq t\leq t_{n+1}} \left\| G^{(d)}(t) \right\|\end{aligned}$$

und (2.57)

$$\begin{aligned}\|\delta_{n+1}\|_V &= \left\| \tau \int_0^1 (-A)^\alpha e^{(1-\theta)\tau A}\big(G(t_n+\theta\tau) - \tilde{p}_n(t_n+\theta\tau)\big)\mathrm{d}\theta \right\| \\ &\leq C\tau^{1-\alpha} \int_0^1 \frac{1}{(1-\theta)^\alpha} \|G(t_n+\theta\tau) - \tilde{p}_n(t_n+\theta\tau)\| \,\mathrm{d}\theta \\ &\leq C\tau^{d+1-\alpha} \sup_{0\leq t\leq t_{n+1}} \left\| G^{(d)}(t) \right\|,\end{aligned}$$

mit einer Konstante C, die nur von C_P und $C_{\tilde{P}}$ abhängt. Definieren wir den Fehler eines expliziten exponentiellen Mehrschrittverfahrens (2.43) zur Zeit t_n durch $e_n := y_n - y(t_n)$, so liefert dies uns die folgende Fehlerrekursion

$$e_{n+1} = e^{\tau A} e_n + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau A)\big(g_{n-k+1+j} - G_{n-k+1+j}\big) - \delta_{n+1}$$

mit $g_i := g(t_i, y_i)$ für $i \geq 0$. Durch Auflösung der Rekursion erhalten wir wegen $e_0 = 0$ die nachfolgende Gleichung

$$e_n = \tau \sum_{l=0}^{n-1} e^{(n-1-l)\tau A} \left(\sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau A)\big(g_{l-k+1+j} - G_{l-k+1+j}\big) - \frac{1}{\tau}\delta_{l+1} \right). \tag{2.58}$$

Bei der Abschätzung von (2.58) werden Terme der Form

$$e^{t_{n-l}A}\big(g_j - G_j\big)$$

auftreten. Diese Terme lassen sich für $l < n$ mithilfe der Voraussetzung 2.10 abschätzen

$$\begin{aligned}\left\| e^{t_{n-l}}\big(g_j - G_j\big) \right\|_V &\leq t_{n-l}^{-\alpha} \left\| t_{n-l}^\alpha (-A)^\alpha e^{t_{n-l}A} \right\| \cdot \|g_j - G_j\| \\ &\leq C_P \mathcal{L}_g t_{n-l}^{-\alpha} \|e_j\|_V .\end{aligned} \tag{2.59}$$

Unter Verwendung der Ungleichung

$$t_j^{-\alpha} \leq C t_{j+m}^{-\alpha} \quad \text{für} \quad 0 \leq m \leq k-1 \quad \text{mit} \quad C = C(k) \tag{2.60}$$

können wir aus (2.58) folgende Fehlerabschätzung folgern

$$\begin{aligned}
\|e_n\|_V &= \left\| \tau \sum_{l=0}^{n-1} e^{(n-1-l)\tau A} \left(\sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau A)\big(g_{l-k+1+j} - G_{l-k+1+j}\big) - \frac{1}{\tau}\delta_{l+1} \right) \right\|_V \\
&\leq \tau \sum_{l=0}^{n-2} \sum_{j=0}^{k-1} \left\| \beta_j^{\mathrm{eM}}(\tau A)(-A)^{\alpha} e^{(n-1-l)\tau A}\big(g_{l-k+1+j} - G_{l-k+1+j}\big) \right\| \\
&\quad + \sum_{l=0}^{n-2} \left\| (-A)^{\alpha} e^{(n-1-l)\tau A} \delta_{l+1} \right\| + \tau \sum_{j=0}^{k-1} \left\| (-A)^{\alpha} \beta_j^{\mathrm{eM}}(\tau A)\big(g_{n-k+j} - G_{n-k+j}\big) \right\| \\
&\quad + \|\delta_n\|_V \\
&\leq \tau \sum_{l=0}^{n-2} \left\| (-A)^{\alpha} e^{(n-1-l)\tau A} \right\| \left(\sum_{j=0}^{k-1} \left\| \beta_j^{\mathrm{eM}}(\tau A) \right\| \cdot \|g_{l-k+1+j} - G_{l-k+1+j}\| + \frac{\|\delta_{l+1}\|}{\tau} \right) \\
&\quad + \tau \sum_{j=0}^{k-1} \underbrace{\left\| (-A)^{\alpha} \beta_j^{\mathrm{eM}}(\tau A) \right\|}_{\leq C\tau^{-\alpha}} \cdot \|g_{n-k+j} - G_{n-k+j}\| + \|\delta_n\|_V \\
&\overset{2.10,(2.46)}{\underset{(2.59)}{\leq}} \tau C \sum_{l=0}^{n-2} t_{n-1-l}^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\|_V + \tau^d \right) + \tau C \sum_{j=0}^{k-1} \tau^{-\alpha} \|e_{n-k+j}\|_V + \|\delta_n\|_V \\
&\overset{(2.60)}{\leq} \tau C \sum_{l=0}^{n-2} t_{n-l}^{-\alpha} \left(\|e_l\|_V + \tau^d \right) + \tau C t_{n-(n-1)}^{-\alpha} \sum_{j=n-k}^{n-1} \|e_j\|_V + \|\delta_n\|_V \\
&\leq C \max_{l=1,\ldots,k-1} \|e_l\|_V + \tau C \sum_{l=0}^{n-1} t_{n-l}^{-\alpha} \left(\|e_l\|_V + \tau^d \right) + \|\delta_n\|_V
\end{aligned}$$

mit einer Konstante $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g)$. Mit der Voraussetzung zu den Startwerten (2.55) sowie den Abschätzungen (2.57) und

$$\tau \sum_{l=0}^{n-1} t_{n-l}^{-\alpha} \leq \int_0^T t^{-\alpha} \mathrm{d}t = \frac{T^{1-\alpha}}{1-\alpha} \tag{2.61}$$

erhalten wir für $\varepsilon_j := \|e_j\|_V$ mit $1 \leq j \leq n$ die Ungleichung

$$\varepsilon_n \leq C\tau \sum_{l=1}^{n-1} t_{n-l}^{-\alpha} \varepsilon_l + C\tau^d.$$

Die Anwendung des Gronwall-Lemmas 2.17 liefert uns die Behauptung mit einer Konstante $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g, T, c_0, \alpha)$. □

Die Konvergenz von exponentiellen Predictor-Corrector-Verfahren können wir mit ähnlichen Techniken beweisen. Dieses Resultat haben wir im nächsten Satz verfasst.

Satz 2.20: *(Konvergenz von exponentiellen Predictor-Corrector-Verfahren)*

a) *Betrachten wir unter den Voraussetzungen von Satz 2.19 ein exponentielles Predictor-Corrector-Verfahren mit hinreichend genauen Startwerten*

$$\|y_j - y(t_j)\|_V \leq c_0 \tau^d, \quad j = 1, \ldots, k-1, \quad k \geq d, \tag{2.62}$$

deren Predictor- und Corrector-Verfahren von der Konsistenzordnung d sind, so ist der Fehler der numerischen Lösung eines solchen Verfahrens zur Schrittweite τ durch

$$\|y_n - y(t_n)\|_V \leq C\tau^d \quad mit \quad C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g, T, c_0, \alpha)$$

gleichmäßig beschränkt für $0 \leq n\tau \leq T$.

b) *Im Vergleich zur Aussage in a) setzen wir stärker voraus, dass $G \in C^{d+1}([0,T], X)$ gilt und die Startwerte statt (2.62) der Abschätzung*

$$\|y_j - y(t_j)\|_V \leq c_0 \tau^{d+1}, \quad j = 1, \ldots, k-1, \quad k \geq d \tag{2.63}$$

genügen. Wenn zudem das Corrector-Verfahren von der Konsistenzordnung $d+1$ ist, dann ist unter diesen Voraussetzungen der Fehler der numerischen Lösung eines exponentiellen Predictor-Corrector-Verfahrens gleichmäßig beschränkt durch

$$\|y_n - y(t_n)\|_V \leq C\tau^{d+1-\alpha} \quad mit \quad C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g, T, c_0, \alpha)$$

für $0 \leq n\tau \leq T$.

Beweis: *(Satz 2.20)*
Aufgrund der Analogie der Beweise für exponentielle Predictor-Corrector-Verfahren der ersten und zweiten Art haben wir diesen Satz allgemein formuliert. Aus demselben Grund werden wir den Beweis anhand von exponentiellen Predictor-Corrector-Verfahren der zweiten Art ausführen.

Teil a): Definieren wir $\tilde{p}_n$ und $\tilde{q}_n$ als die Polynome (2.36) und (2.37)

$$\begin{aligned}\tilde{p}_n(t_n+\theta\tau) &= \sum_{i=0}^{d-1}\sum_{j=0}^{k-1} b_{i,j}^{\mathrm{eM}}\frac{\theta^i}{i!}G_{n-k+1+j},\\ \tilde{q}_n(t_n+\theta\tau) &= \sum_{i=0}^{d-1}\sum_{j=0}^{k} b_{i,j}^{\mathrm{iM}}\frac{\theta^i}{i!}G_{n-k+1+j},\end{aligned} \tag{2.64}$$

zu den exakten Daten $\big(t_j, G(t_j)\big)$ mit $n-k+1 \le j \le n+1$, wobei $t_j := j\tau$ und $G_j := G(t_j)$ gilt. Die exakte Lösung können wir damit nach der VdK-Formel (2.20) zum einen durch

$$\begin{aligned}y(t_{n+1}) &= e^{\tau A}y(t_n) + \tau\int_0^1 e^{(1-\theta)\tau A}\tilde{p}_n(t_n+\theta\tau)\mathrm{d}\theta + \delta_{n+1}\\ &= e^{\tau A}y(t_n) + \tau\sum_{j=0}^{k-1}\beta_j^{\mathrm{eM}}(\tau A)G_{n-k+1+j} + \delta_{n+1}\end{aligned}$$

und zum anderen durch

$$\begin{aligned}y(t_{n+1}) &= e^{\tau A}y(t_n) + \tau\int_0^1 e^{(1-\theta)\tau A}\tilde{q}_n(t_n+\theta\tau)\mathrm{d}\theta + \psi_{n+1}\\ &= e^{\tau A}y(t_n) + \tau\sum_{j=0}^{k}\beta_j^{\mathrm{iM}}(\tau A)G_{n-k+1+j} + \psi_{n+1}\end{aligned}$$

beschreiben. Die Defekte δ_{n+1} und ψ_{n+1} entsprechen damit den Ausdrücken

$$\begin{aligned}\delta_{n+1} &:= \tau\int_0^1 e^{(1-\theta)\tau A}\big(G(t_n+\theta\tau) - \tilde{p}_n(t_n+\theta\tau)\big)\mathrm{d}\theta,\\ \psi_{n+1} &:= \tau\int_0^1 e^{(1-\theta)\tau A}\big(G(t_n+\theta\tau) - \tilde{q}_n(t_n+\theta\tau)\big)\mathrm{d}\theta\end{aligned}$$

und können mit der Voraussetzung 2.10 und dem Lemma 2.18 genau wie (2.57) abgeschätzt werden

$$\begin{aligned}\|\delta_{n+1}\| &\le C\tau^{d+1}\sup_{0\le t\le t_{n+1}}\left\|G^{(d)}(t)\right\|, \quad \|\delta_{n+1}\|_V \le C\tau^{d+1-\alpha}\sup_{0\le t\le t_{n+1}}\left\|G^{(d)}(t)\right\|,\\ \|\psi_{n+1}\| &\le C\tau^{d+1}\sup_{0\le t\le t_{n+1}}\left\|G^{(d)}(t)\right\|, \quad \|\psi_{n+1}\|_V \le C\tau^{d+1-\alpha}\sup_{0\le t\le t_{n+1}}\left\|G^{(d)}(t)\right\|\end{aligned} \tag{2.65}$$

mit einer Konstante $C = C(C_P, C_{\tilde{P}})$. Es ergibt sich die Fehlerrekursion

$$e_{n+1} = e^{\tau A} e_n + \tau \sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau A)\big(g^*_{n-k+1+j} - G_{n-k+1+j}\big) - \psi_{n+1},$$

mit

$$g^*_{n+1+j-k} = \begin{cases} g(t_{n-k+1+j}, y_{n-k+1+j}), & j \neq k \\ g(t_{n-k+1+j}, \bar{y}_{n-k+1+j}), & j = k \end{cases} \quad \text{und} \quad \bar{y}_n = e^{\tau A} y_n + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau A) g_{n-k+1+j}.$$

Lösen wir die Fehlerrekursion auf, so ergibt sich wegen $e_0 = 0$ die Gleichung

$$e_n = \tau \sum_{l=0}^{n-1} e^{(n-1-l)\tau A} \left(\sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau A)\big(g^*_{l-k+1+j} - G_{l-k+1+j}\big) - \frac{1}{\tau}\psi_{l+1} \right). \tag{2.66}$$

Für die Abschätzung von (2.66) benötigen wir neben (2.59) und (2.60) auch eine Abschätzung für Terme der Form

$$e^{t_{n-l}A}\big(\bar{g}_j - G_j\big) \quad \text{mit} \quad \bar{g}_j := g(t_j, \bar{y}_j),$$

die für $l < n$ wie folgt abgeschätzt werden können

$$\begin{aligned} \big\| e^{t_{n-l}A}\big(\bar{g}_j - G_j\big) \big\|_V &\overset{(2.59)}{\leq} C\mathcal{L}_g t_{n-l}^{-\alpha} \left\| \bar{y}_j - y(t_j) \right\|_V \\ &\leq C t_{n-l}^{-\alpha} \left\| e^{\tau A} e_{j-1} + \tau \sum_{s=j-k}^{j-1} \beta_{s-j+k}^{\mathrm{eM}}(\tau A)\big(g_s - G_s\big) + \delta_j \right\|_V \\ &\leq C t_{n-l}^{-\alpha} \left(C \left\| e_{j-1} \right\|_V + C\tau^{1-\alpha} \sum_{s=j-k}^{j-1} \left\| e_s \right\|_V + \left\| \delta_j \right\|_V \right) \\ &\leq C t_{n-l}^{-\alpha} \left(\sum_{s=j-k}^{j-1} \left\| e_s \right\|_V + \left\| \delta_j \right\|_V \right), \end{aligned} \tag{2.67}$$

wobei die Konstante C in (2.67) von C_P, $\mathcal{L}_g$ und den Koeffizienten $\mathcal{B}$ abhängt. Insgesamt können wir (2.66) ähnlich wie im Beweis von Satz 2.19 abschätzen

$$
\begin{aligned}
\|e_n\|_V &= \left\| \tau \sum_{l=0}^{n-1} e^{(n-1-l)\tau A} \left(\sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau A)\big(g^*_{l-k+1+j} - G_{l-k+1+j}\big) - \frac{1}{\tau}\psi_{l+1} \right) \right\|_V \\
&\leq \tau \sum_{l=0}^{n-2} \sum_{j=0}^{k-1} \left\| \beta_j^{\mathrm{iM}}(\tau A)(-A)^{\alpha} e^{(n-1-l)\tau A}\big(g_{l-k+1+j} - G_{l-k+1+j}\big) \right\| \\
&\quad + \tau \sum_{l=0}^{n-2} \left\| \beta_k^{\mathrm{iM}}(\tau A)(-A)^{\alpha} e^{(n-1-l)\tau A}\big(\bar{g}_{l+1} - G_{l+1}\big) \right\| + \sum_{l=0}^{n-2} \left\| (-A)^{\alpha} e^{(n-1-l)\tau A} \psi_{l+1} \right\| \\
&\quad + \tau \sum_{j=0}^{k-1} \left\| (-A)^{\alpha} \beta_j^{\mathrm{iM}}(\tau A)\big(g_{n-k+j} - G_{n-k+j}\big) \right\| + \tau \left\| (-A)^{\alpha} \beta_k^{\mathrm{iM}}(\tau A)\big(\bar{g}_n - G_n\big) \right\| \\
&\quad + \|\psi_n\|_V \\
&\overset{\substack{2.10,(2.46)\\(2.59),(2.67)}}{\leq} \tau C \sum_{l=0}^{n-2} t_{n-1-l}^{-\alpha} \sum_{j=0}^{k-1} \|e_{l-k+1+j}\|_V + \tau C \sum_{l=0}^{n-2} t_{n-1-l}^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\|_V + \|\delta_{l+1}\|_V \right) \\
&\quad + \tau C \sum_{l=0}^{n-2} t_{n-1-l}^{-\alpha} \frac{\|\psi_{l+1}\|}{\tau} + \tau C \sum_{j=0}^{k-1} \tau^{-\alpha} \|e_{n-k+j}\|_V \\
&\quad + \tau C \tau^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{n-k+j}\|_V + \|\delta_n\|_V \right) + \|\psi_n\|_V \\
&\leq \tau C \sum_{l=0}^{n-2} t_{n-1-l}^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\|_V + \|\delta_{l+1}\|_V + \frac{\|\psi_{l+1}\|}{\tau} \right) \\
&\quad + \tau C \tau^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{n-k+j}\|_V + \|\delta_n\|_V \right) + \|\psi_n\|_V \\
&\overset{\substack{(2.60)\\(2.65)}}{\leq} \tau C \sum_{l=0}^{n-2} t_{n-l}^{-\alpha} \left(\|e_l\|_V + \tau^d \right) + \tau C t_{n-(n-1)}^{-\alpha} \left(\sum_{j=n-k}^{n-1} \|e_j\|_V + \tau^{d+1-\alpha} \right) + \|\psi_n\|_V \\
&\leq C \max_{j=1,\ldots,k-1} \|e_j\|_V + \tau C \sum_{l=0}^{n-1} t_{n-l}^{-\alpha} \left(\|e_l\|_V + \tau^d \right) + \|\psi_n\|_V .
\end{aligned}
$$

Hierbei besitzt die Konstante C die folgenden Abhängigkeiten $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g)$. Verwenden wir die Voraussetzung zu den Startwerten (2.62) sowie die Abschätzungen (2.65) und (2.61), so ergibt sich für $\varepsilon_j := \|e_j\|_V$ mit $1 \leq j \leq n$ die Ungleichung

$$
\varepsilon_n \leq C\tau \sum_{l=1}^{n-1} t_{n-l}^{-\alpha} \varepsilon_l + C\tau^d .
$$

Die Anwendung des Gronwall-Lemmas 2.17 liefert uns erneut die Behauptung mit einer Konstanten $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_g, g, T, c_0, \alpha)$.

Analog zu Teil a) erhalten wir die folgende Abschätzung für den Fehler

$$\begin{aligned}
\|e_n\|_V &\leq \tau C \sum_{l=0}^{n-2} t_{n-1-l}^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\|_V + \|\delta_{l+1}\|_V + \frac{\|\psi_{l+1}\|}{\tau} \right) \\
&\quad + \tau C \tau^{-\alpha} \left(\sum_{j=0}^{k-1} \|e_{n-k+j}\|_V + \|\delta_n\|_V \right) + \|\psi_n\|_V \\
&\underset{(2.65)}{\overset{(2.60)}{\leq}} \tau C \sum_{l=0}^{n-2} t_{n-l}^{-\alpha} \left(\|e_l\|_V + \tau^{d+1-\alpha} \right) + \tau C t_{n-(n-1)}^{-\alpha} \left(\sum_{j=n-k}^{n-1} \|e_j\|_V + \tau^{d+1-\alpha} \right) + \|\psi_n\|_V \\
&\leq C \max_{j=1,\ldots,k-1} \|e_j\|_V + \tau C \sum_{l=0}^{n-1} t_{n-l}^{-\alpha} \left(\|e_l\|_V + \tau^{d+1-\alpha} \right) + \|\psi_n\|_V .
\end{aligned}$$

In diesem Fall verwenden wir die Voraussetzung (2.63) sowie die Abschätzungen (2.65) und (2.61). Damit ergibt sich die folgende Ungleichung für die Folge $(\varepsilon_j)_{j \in \mathbb{N}}$ mit $\varepsilon_j := \|e_j\|_V$

$$\varepsilon_n \leq C\tau \sum_{l=1}^{n-1} t_{n-l}^{-\alpha} \varepsilon_l + C\tau^{d+1-\alpha}.$$

Mit dem Gronwall-Lemma 2.17 erhalten wir schließlich die Behauptung. Die Konstanten in Teil b) sind zwar nicht identisch mit den Konstanten aus Teil a), aber Sie besitzen dieselben Abhängigkeiten. □

Für die Konvergenzbeweise ist die Fehleranalyse von Hochbruck und Ostermann [33] von fundamentaler Bedeutung. Aus dem Grund wurden die Beweise möglichst detailliert mit einer alternativen Notation beschrieben. Das Resultat konnten wir dabei für allgemeinere Polynomapproximationen erweitern. Zudem haben wir auf dieser Basis die Konvergenzanalyse für die Predictor-Corrector-Verfahren übertragen. Die Konvergenz haben wir stets unter der Voraussetzung 2.10 bewiesen, da wir damit Abschätzungen der Form (2.59) und (2.67) erhalten konnten. Mit der parabolischen Konfiguration (s. Voraussetzung 2.10) konnten wir die Aussagen aus Satz 2.19 und 2.20 unter der $\|\cdot\|_V$-Norm beweisen und zwar aufgrund der Eigenschaften einer analytischen Halbgruppe [47, Thm. 6.13]. Für Operatoren, die aus den Maxwell-Gleichungen resultieren sind stärkere Voraussetzungen an die nichtlineare Funktion g zu treffen, damit auch in diesem Fall Abschätzungen der Form

(2.59) und (2.67) gelten. Verwenden wir statt der Voraussetzung 2.10 die Konfiguration für hyperbolische Probleme aus der Voraussetzung 2.12, so sind die Sätze 2.19 und 2.20 auch für diese Art von Problemen gültig.

2.2.6 Optimierte exponentielle Mehrschrittverfahren

Wir haben in Abschnitt 2.2.4 allgemeine exponentielle Mehrschrittverfahren kennengelernt und in 2.2.5 deren Eigenschaften analysiert. Wir haben festgestellt, dass exponentielle Adams-Verfahren und exponentielle Predictor-Corrector-Verfahren, die auf expliziten und impliziten exponentiellen Adams-Verfahren beruhen, gute Stabilitätseigenschaften aufweisen. Die Stabilität hängt im Grunde genommen vom nichtlinearen Anteil g des semilinearen Problems (2.1) ab, sofern die Steifheit im linearen Anteil enthalten ist. Allerdings ist uns auch bekannt, dass sich die Stabilitätsgebiete von Adams-Verfahren und der Predictor-Corrector-Verfahren sowohl im exponentiellen Fall als auch im klassischen Fall mit zunehmender Ordnung verkleinern (s. Abbildungen 2.5, 2.6 und 2.7).

Aus diesen Gründen werden wir in diesem Abschnitt exponentielle Mehrschrittverfahren konstruieren, die bezüglich der Stabilität optimiert sind. Die Stabilitätsresultate (s. Abbildungen 2.12 und 2.13) aus Abschnitt 2.2.5 motivieren die Optimierung bezüglich der klassischen Stabilität, da die Stabilität, wie bereits erwähnt, im Grunde genommen vom nichtlinearen Anteil abhängt. Dazu ist es wichtig den Zusammenhang zwischen den Koeffizienten $b_{i,j}$ exponentieller Mehrschrittverfahren und den Koeffizienten β_j klassischer Mehrschrittverfahren zu verstehen.

Ein exponentielles Mehrschrittverfahren besitzt nach (2.43) die Darstellung

$$y_{n+1} = e^{\tau A} y_n + \tau \sum_{i=0}^{d-1} \sum_{j=0}^{k} b_{i,j} \varphi_{i+1}(\tau A) g_{n-k+1+j} = e^{\tau A} y_n + \tau \sum_{j=0}^{k} \beta_j(\tau A) g_{n-k+1+j},$$

wobei

$$\beta_j(x) := \sum_{i=0}^{d-1} b_{i,j} \varphi_{i+1}(x)$$

gilt. Aus dieser Darstellung können wir leicht erkennen, dass ein exponentielles Mehrschrittverfahren für $A = 0$ einem klassischen Mehrschrittverfahren (2.15) entspricht, das durch die Koeffizienten

$$\beta_j(0) = \sum_{i=0}^{k} \frac{b_{i,j}}{(i+1)!}, \quad j = 0, \ldots, k \tag{2.68}$$

beschrieben wird. Des Weiteren können wir folgern, dass die Ordnungsbedingungen exponentieller Mehrschrittverfahren auch die klassischen Ordnungsbedingungen „enthalten“.

Korollar 2.21:

Sind die Ordnungsbedingungen (Satz 2.13) exponentieller Mehrschrittverfahren (2.43) für ein $d \geq 0$ erfüllt, so besitzt das zugrundeliegende eindeutige klassische Mehrschrittverfahren ($A = 0$) dieselbe Konsistenzordnung wie die entsprechende exponentielle Variante.

Beweis: *(Korollar 2.21)*
Betrachten wir ein klassisches Mehrschrittverfahren mit den Koeffizienten

$$\alpha_k = 1,\ \alpha_{k-1} = -1,\ \alpha_j = 0,\ 0 \leq j \leq k-2, \quad \text{und} \quad \beta_j \overset{(2.68)}{=} \sum_{i=0}^{d-1} \frac{b_{i,j}}{(i+1)!},\ j = 0, \ldots, k,$$

wobei die Koeffizienten den Ordnungsbedingungen aus Satz 2.13 genügen, so gilt für den Differenzenoperator

$$L(y, t_0, \tau) = \sum_{j=k-1}^{k} y(t_0 + j\tau) - \tau \sum_{j=0}^{k} \beta_j y'(t_0 + j\tau).$$

Die Taylorentwicklung des Differenzenoperators um $t_{k-1} := t_0 + (k-1)\tau$ liefert

$$\begin{aligned} L(y, t_0, \tau) &= \sum_{l=1}^{d} \frac{1}{l!} \tau^l y^{(l)}(t_{k-1}) - \sum_{l=0}^{d-1} \sum_{j=0}^{k} \sum_{i=0}^{d-1} \frac{b_{i,j}}{(i+1)!} \frac{(j+1-k)^l}{l!} \tau^{l+1} y^{(l+1)}(t_{k-1}) + \mathcal{O}(\tau^{d+1}) \\ &= \sum_{l=0}^{d} \underbrace{\left(\frac{1}{(l+1)!} - \sum_{i=0}^{d-1} \frac{1}{(i+1)!} \sum_{j=0}^{k} b_{i,j} \frac{(j+1-k)^l}{l!} \right)}_{\overset{2.13}{=} 0} \tau^{l+1} y^{(l+1)}(t_{k-1}) + \mathcal{O}(\tau^{d+1}) \\ &= \mathcal{O}(\tau^{d+1}), \end{aligned}$$

womit die Behauptung bewiesen ist. □

Das Korollar 2.21 zeigt, dass mit der Bestimmung eines optimierten exponentiellen Mehrschrittverfahrens gleichzeitig ein optimiertes klassisches Mehrschrittverfahren derselben

Konsistenzordnung eindeutig bestimmt wird. Wenn wir hingegen ein optimiertes klassisches Mehrschrittverfahren bestimmen, so ist die Eindeutigkeit des optimierten exponentiellen Mehrschrittverfahrens, aufgrund der Beziehung (2.68) nicht gegeben. Für die Optimierung würde dies jedoch einen Vorteil darstellen, da klassische Verfahren deutlich weniger Parameter haben und die Optimierung damit weniger Aufwand beanspruchen würde. Deswegen werden wir später zusätzliche Bedingungen treffen, die eine eindeutige Bestimmung eines optimierten exponentiellen Mehrschrittverfahrens mithilfe eines optimierten klassischen Mehrschrittverfahren gewährleisten.

Die Konstruktion optimierter Verfahren beruht auf der Idee der Stabilitätsoptimierung von Runge-Kutta-Verfahren. Niegemann, Diehl und Busch haben in Ihrer Arbeit [41] low-storage Runge-Kutta-Verfahren mit optimalen Stabilitätseigenschaften konstruiert. Auf der Basis dieser Idee werden wir ein Optimierungsproblem für explizite exponentielle Mehrschrittverfahren der Form (2.43) formulieren. Für exponentielle Predictor-Corrector-Verfahren funktioniert dies analog.

Es ist offensichtlich, dass diese Optimierung problemabhängig sein wird, weil die Stabilität vom Spektrum des jeweiligen Problems abhängt. Es gibt Fälle, wo das Spektrum einer Teilmenge der reellen Achse bzw. der imaginären Achse entspricht. Hesthaven und Warburton [27, Kap. 8, Abb. 8.8, Abb. 8.13, Abb. 8.23] haben die Spektraleigenschaften für Discontinuous-Galerkin-Diskretisierungen von Maxwell-Problemen mit verschiedenen Flüssen untersucht. Die Verwendung von einem Upwind-Flux führt in diesem Fall dazu, dass die Eigenwerte des diskretisierten Operators nicht mehr rein imaginär sind, sondern in der linken Halbebene liegen. Niegemann, Diehl und Busch [42] konnten diesbezüglich neben einem rein imaginären auch kreis- und ellipsenförmige Spektralprototypen identifizieren.

Dementsprechend ist es wichtig abhängig vom gegebenen Problem ein Zielgebiet $\Lambda_{\text{target}}^{\text{opt}}$ zu definieren, das die Kontur des Spektrums beschreibt oder zumindest gut approximiert. Außerdem sollte zu diesem Zielgebiet $\Lambda_{\text{target}}^{\text{opt}}$ eine Distanzfunktion mit folgenden Eigenschaften definiert werden

$$\text{dist} : \mathbb{R}^2 \times \mathbb{R}_{\geq 0} \to \mathbb{R}, \quad \begin{cases} \text{dist}(x, y, \mu) > 0, & (x, y) \notin \Lambda_{\text{target}}(\mu) \\ \text{dist}(x, y, \mu) < 0, & (x, y) \in \Lambda_{\text{target}}(\mu) \setminus \partial\Lambda_{\text{target}}(\mu) \\ \text{dist}(x, y, \mu) = 0, & (x, y) \in \partial\Lambda_{\text{target}}(\mu) \end{cases},$$

mit

$$\Lambda_{\text{target}}(\mu) := \{(\mu x, \mu y) \,|\, (x,y) \in \Lambda^{\text{opt}}_{\text{target}}\} \quad \text{für} \quad 0 \leq \mu \leq 1 \quad \text{mit} \quad \Lambda_{\text{target}}(1) = \Lambda^{\text{opt}}_{\text{target}}.$$

Zur Bestimmung von optimierten exponentiellen Mehrschrittverfahren bezeichnen wir mit $\mathcal{B}_k \in \mathbb{R}^{d\times k}$ die Koeffizientenmatrix eines expliziten exponentiellen Mehrschrittverfahrens (2.44). Damit erfüllt ein exponentielles Mehrschrittverfahren der Konsistenzordnung d (s. Satz 2.13) die Gleichung

$$\mathcal{V}\mathcal{B}_k^{\mathrm{T}} = \mathbb{1}^{d\times d} \quad \text{mit} \quad \mathcal{V} \in \mathbb{R}^{d\times k} \quad \text{und} \quad \mathcal{V}_{i,j} = \frac{(j-k)^{i-1}}{(i-1)!},$$

wobei $\mathbb{1}^{d\times d}$ die $d \times d$ Einheitsmatrix beschreibt. Soweit $d = k$ gilt, ist die Matrix $\mathcal{V}$ invertierbar und die Koeffizienten sind eindeutig bestimmt. In diesem Fall erhalten wir die exponentiellen Adams-Verfahren. Für $d < k$ ist diese Gleichung unterbestimmt mit $(k-d)\cdot d$ frei wählbaren Koeffizienten, denn wegen

$$\mathcal{B}_{\text{det}} = \mathcal{V}_d^{-1}\left(\mathbb{1}^{d\times d} - \mathcal{V}_{k-d}\,[\,\mathcal{B}_{\text{col}} \,|\, \mathcal{B}_{\text{free}}\,]\right) \tag{2.69}$$

für $\mathcal{V} := [\,\mathcal{V}_{k-d} \,|\, \mathcal{V}_d\,]$ mit $\mathcal{V}_d \in \mathbb{R}^{d\times d}$, $\mathcal{V}_{k-d} \in \mathbb{R}^{d\times(k-d)}$ und

$$\mathcal{B}_k^{\mathrm{T}} = \left(\begin{array}{cc} \mathcal{B}_{\text{col}} & \mathcal{B}_{\text{free}} \\ \multicolumn{2}{c}{\mathcal{B}_{\text{det}}} \end{array} \right) \tag{2.70}$$

mit $\mathcal{B}_{\text{det}} \in \mathbb{R}^{d\times d}$, $\mathcal{B}_{\text{col}} \in \mathbb{R}^{(k-d)\times 1}$ und $\mathcal{B}_{\text{free}} \in \mathbb{R}^{(k-d)\times(d-1)}$ sind die Koeffizienten $\mathcal{B}_{\text{det}}$ bestimmt durch die übrigen Koeffizienten. Wählen wir $\mathcal{B}_{\text{free}} = 0$, so impliziert dies

$$\mathcal{B}_{\text{col}} \overset{(2.68)}{=} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-d-1} \end{pmatrix},$$

wobei die Koeffizienten β_j für $j = 0, \ldots, k-d-1$ den Koeffizienten des klassischen Verfahrens (2.68) entsprechen. Diese Darstellung ermöglicht uns das Optimierungsproblem von exponentiellen Mehrschrittverfahren auf ein Optimierungsproblem klassischer Mehrschrittverfahren zu reduzieren mit $k-d$ freien Koeffizienten. Mithilfe der Zusatzvoraussetzung $\mathcal{B}_{\text{free}} = 0$ können wir zu einem klassischen Mehrschrittverfahren ein exponentielles Mehrschrittverfahren derselben Konsistenzordnung eindeutig bestimmen. Anhand der Gleichung (2.69) ergeben sich die Koeffizienten des optimierten exponentiellen Mehrschrittverfahrens aus den Koeffizienten des optimierten klassischen Verfahrens. Verwenden wir die Kurzschreibweise $\mathrm{MS}_k(\boldsymbol{\beta})$ mit $\boldsymbol{\beta} := [\beta_0, \ldots, \beta_{k-1}]$ für ein klassisches k-Schrittverfahren, deren Stabilitätsgebiet durch $\mathcal{S}_M(\boldsymbol{\beta})$ beschrieben wird, so können wir das Optimierungsproblem für ein k-Schrittverfahren der Konsistenzordnung d wie folgt beschreiben

$$\mu_{\max} := \max\{\, \mu \,|\, \mathrm{MS}_k(\boldsymbol{\beta}) \text{ besitzt die Konsistenzordnung } d,\ \Lambda_{\text{target}}(\mu) \subset \mathcal{S}_M(\boldsymbol{\beta}) \,\}. \quad (2.71)$$

Die Optimierung von Predictor-Corrector-Verfahren lässt sich ähnlich realisieren, doch aus speichertechnischen Gründen ist eine Optimierung von Predictor-Corrector-Verfahren der ersten Art vorzuziehen, denn der Speicheraufwand eines $\mathrm{PC}[2]_k$-Verfahrens ist identisch mit dem Speicheraufwand eines $\mathrm{PC}[1]_{k+1}$-Verfahrens. Zudem ist ein $\mathrm{PC}[2]_k$-Verfahren ein Spezialfall eines $\mathrm{PC}[1]_{k+1}$-Verfahrens mit $\beta_0^{\mathrm{eM}} = 0$. Wegen der Aussage aus Satz 2.20 b) ist es von Vorteil, wenn wir zur Bestimmung eines optimierten exponentiellen Predictor-Corrector-Verfahrens der Konsistenzordnung d ein Predictor-Verfahren der Konsistenzordnung $d-1$ und ein Corrector-Verfahren der Konsistenzordnung d verwenden. Die optimierten exponentiellen Predictor-Corrector-Verfahren, die wir in unserer Arbeit verwenden, werden ausschließlich von diesem Typ sein.

Die Koeffizientenmatrizen eines optimierten exponentiellen Predictor-Corrector-Verfahrens der Konsistenzordnung d sind gegeben durch

$$\mathcal{B}_k^{\mathrm{Pred}} := \begin{pmatrix} b_{0,0}^{\mathrm{eM}} & \cdots & b_{0,k-1}^{\mathrm{eM}} \\ \vdots & \vdots & \vdots \\ b_{d-2,0}^{\mathrm{eM}} & \cdots & b_{d-2,k-1}^{\mathrm{eM}} \end{pmatrix}, \quad \text{und} \quad \mathcal{B}_k^{\mathrm{Corr}} := \begin{pmatrix} b_{0,1}^{\mathrm{iM}} & \cdots & b_{0,k}^{\mathrm{iM}} \\ \vdots & \vdots & \vdots \\ b_{d-1,1}^{\mathrm{iM}} & \cdots & b_{d-1,k}^{\mathrm{iM}} \end{pmatrix}, \quad (2.72)$$

mit $\mathcal{B}_k^{\text{Pred}} \in \mathbb{R}^{(d-1)\times k}$ und $\mathcal{B}_k^{\text{Corr}} \in \mathbb{R}^{d\times k}$ und erfüllen die folgenden Gleichungen

$$\mathcal{V}^{\text{Pred}} \left(\mathcal{B}_k^{\text{Pred}}\right)^{\text{T}} = \mathbb{1}^{(d-1)\times(d-1)} \quad \text{mit} \quad \mathcal{V}^{\text{Pred}} \in \mathbb{R}^{(d-1)\times k} \quad \text{und} \quad \mathcal{V}_{i,j}^{\text{Pred}} = \frac{(j-k)^{i-1}}{(i-1)!},$$
$$\mathcal{V}^{\text{Corr}} \left(\mathcal{B}_k^{\text{Pred}}\right)^{\text{T}} = \mathbb{1}^{d\times d} \quad \text{mit} \quad \mathcal{V}^{\text{Corr}} \in \mathbb{R}^{d\times k} \quad \text{und} \quad \mathcal{V}_{i,j}^{\text{Corr}} = \frac{(j+1-k)^{i-1}}{(i-1)!}.$$

Analog zur Optimierung exponentieller Mehrschrittverfahren lässt sich die Optimierung exponentieller Predictor-Corrector-Verfahren zu einem Optimierungsproblem klassischer Predictor-Corrector-Verfahren reduzieren. Notieren wir ein klassisches Predictor-Corrector-Verfahren der ersten Art mit $\text{PC}[1]_k(\boldsymbol{\beta}^{\text{eM}}, \boldsymbol{\beta}^{\text{iM}})$. Das Stabilitätsgebiet dieses Verfahrens ist durch $\mathcal{S}_M^{\text{PC}}(\boldsymbol{\beta}^{\text{eM}}, \boldsymbol{\beta}^{\text{iM}})$ gegeben, wobei $\boldsymbol{\beta}^{\text{eM}} = \left[\beta_0^{\text{eM}}, \ldots, \beta_{k-1}^{\text{eM}}\right]$ und $\boldsymbol{\beta}^{\text{iM}} = \left[\beta_1^{\text{iM}}, \ldots, \beta_k^{\text{iM}}\right]$ gilt. Damit können wir das Optimierungsproblem wie folgt beschreiben

$$\max \left\{ \mu \;\middle|\; \begin{array}{c} \text{PC}[1]_k(\boldsymbol{\beta}^{\text{eM}}, \boldsymbol{\beta}^{\text{iM}}) \text{ besitzt die Konsistenzordnung } d, \\ \Lambda_{\text{target}}(\mu) \subset \mathcal{S}_M^{\text{PC}}\left(\boldsymbol{\beta}^{\text{eM}}, \boldsymbol{\beta}^{\text{iM}}\right) \end{array} \right\}. \tag{2.73}$$

Folgende Details zur Implementierung sollten beachtet werden. Die für die Optimierung verwendeten Nebenbedingungen entsprechen den Ordnungsbedingungen und können anhand eines Gleichungssystems beschrieben werden. Darüberhinaus benötigen wir eine Zielfunktion, die für gegebene Koeffizienten β_k den Parameter μ größtmöglich wählt und zwar so, dass $\Lambda_{\text{target}} \subset \mathcal{S}(\boldsymbol{\beta})$ gilt. Der Parameter μ lässt sich z.B. mit einem Bisektionsverfahren bestimmen, das zur Überprüfung der Beziehung $\Lambda_{\text{target}} \subset \mathcal{S}(\boldsymbol{\beta})$ die Distanzfunktion des Zielgebiets verwendet. Bei dieser Prüfung sind jeweils die diskretisierten Konturen zu verwenden.

Falls für ein Zielgebiet keine Distanzfunktion vorliegt, so können wir alternativ die Verifikation der Eigenschaft $\Lambda_{\text{target}} \subset \mathcal{S}(\boldsymbol{\beta})$ mithilfe der zugehörigen Differenzengleichung (2.19) und dem Wurzelkriterium (Satz 2.4) durchführen. Allerdings müssen wir in diesem Fall die Lösungen der jeweiligen Differenzengleichung bestimmen, um anschließend überprüfen zu können, ob das Wurzelkriterium erfüllt ist oder nicht. Wir schließen dieses Kapitel mit zwei Beispielen ab, die die erarbeiteten theoretischen Ergebnisse widerspiegeln.

Beispiel 2.22: *(Gedämpfte Wellengleichung)*
Als Zielgebiet wählen wir

$$\Lambda_{\text{circle}} := \{\, \lambda \in \mathbb{C} \,|\, \text{Re}(\lambda) \leq 0, |\lambda - \lambda_0| < 1 \,\} \quad \text{mit} \quad \lambda_0 := -\cos\left(\sin^{-1}(\tfrac{1}{2})\right). \tag{2.74}$$

Nach [42] approximiert (2.74) eine Obermenge des Spektrums des diskretisierten Operators einer linearen hyperbolischen partiellen Differentialgleichung, wenn für die Diskretisierung ein Discontinuous-Galerkin-Verfahren mit Upwind-Flux verwendet wurde. Die Optimierung (2.71) bezüglich des Zielgebiets Λ_{circle} haben wir sowohl für klassische Mehrschrittverfahren (kurz: MS_k) als auch für klassische Predictor-Corrector-Verfahren (PECE) der ersten Art (kurz: $\text{PC}[1]_k$) ausgeführt und damit k-Schrittverfahren der Konsistenzordnungen $d = 3$ und $d = 4$ mit $d + 1 \leq k \leq 12$ bestimmt. Die effektiven Optimierungsparameter

$$\mu_{\text{eff}} := \frac{\mu_{\max}}{N_{\text{eval}}} \quad \text{mit} \quad N_{\text{eval}} = \begin{cases} 2, & \text{für Predictor-Corrector-Verfahren (PECE)} \\ 1, & \text{sonst} \end{cases}$$

zu den bestimmten optimierten Verfahren haben wir in der Abbildung 2.14 dargestellt. Diese spielen für den Vergleich eine enorm wichtige Rolle, denn in vielen Fällen stellt die Anzahl der Funktionsauswertungen N_{eval} einen erheblichen Rechenaufwand dar. Daher ermöglicht der effektive Optimierungsparameter nahezu einen vom Aufwand unabhängigen Vergleich.

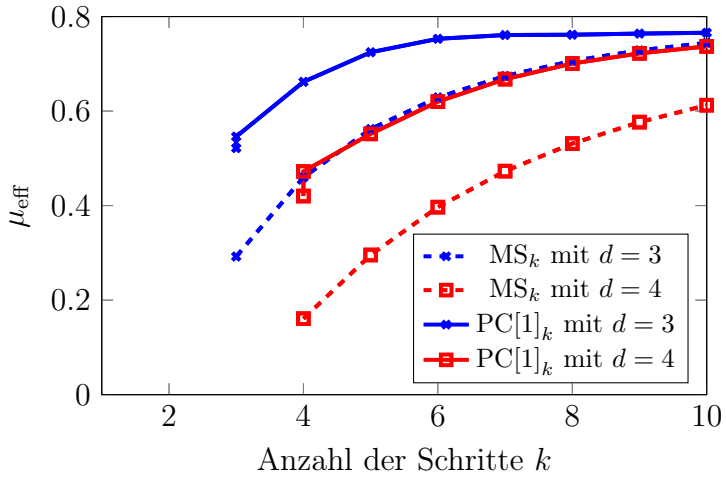

Abbildung 2.14: Effektive Optimierungsparameter μ_{eff} zu MS_k und $\text{PC}[1]_k$.

In Abbildung 2.14 können wir erkennen, dass die Predictor-Corrector-Verfahren die Wahl einer deutlich größeren Zeitschrittweite im Vergleich zu den anderen Mehrschrittverfahren erlauben. Beispielsweise beobachten wir für $\text{PC}[1]_k$ mit $d = 3$ einen klaren Zuwachs (ca. 40 %) des effektiven Optimierungsparameters zwischen $k = 3$ und $k = 6$. Für $\text{PC}[1]_k$ mit

$d = 4$ ist der effektive Optimierungsparamter für $k = 8$ in etwa 65 % größer als für $k = 4$. Ähnliches beobachten wir für die effektiven Optimierungsparamter der MS_k-Verfahren.

Für weitere Beobachtungen beschränken wir uns auf die Verfahren MS_6 und $\mathrm{PC}[1]_6$ mit $d = 3$, sowie MS_8 und $\mathrm{PC}[1]_8$ mit $d = 4$. Für die Koeffizientmatrizen der zugehörigen optimierten exponentiellen Mehrschrittverfahren ergeben sich die Matrizen:

$$\mathcal{B}_6^{\mathrm{T}} = \begin{bmatrix} 0.082869620566909 & 0 & 0 \\ 0.009030721597236 & 0 & 0 \\ -0.090587927705982 & 0 & 0 \\ -0.611116752134563 & \frac{1}{2} & 1 \\ 1.043526298163582 & -2 & -2 \\ 0.566278039512818 & \frac{3}{2} & 1 \end{bmatrix}, \qquad \mathcal{B}_8^{\mathrm{T}} = \begin{bmatrix} -0.083750405791054 & 0 & 0 & 0 \\ 0.037090783605478 & 0 & 0 & 0 \\ 0.140692782646747 & 0 & 0 & 0 \\ 0.116487176855052 & 0 & 0 & 0 \\ 0.316571996689645 & -\frac{1}{3} & -1 & -1 \\ -1.853170110136760 & \frac{3}{2} & 4 & 3 \\ 1.950919748455149 & -3 & -5 & -3 \\ 0.375158027675743 & \frac{11}{6} & 2 & 1 \end{bmatrix},$$

$$(\mathcal{B}_6^{\mathrm{Pred}})^{\mathrm{T}} = \begin{bmatrix} -0.025938543988869 & 0 \\ -0.002399106821385 & 0 \\ -0.002263605135602 & 0 \\ -0.071001515909671 & 0 \\ 0.288082994456032 & -1 \\ 0.813519777399494 & 1 \end{bmatrix}, \qquad (\mathcal{B}_6^{\mathrm{Corr}})^{\mathrm{T}} = \begin{bmatrix} 0.023771212979208 & 0 & 0 \\ 0.001181772062257 & 0 & 0 \\ -0.054777954339211 & 0 & 0 \\ -0.080468899147988 & -\frac{1}{2} & 1 \\ 1.201688508168542 & 0 & -2 \\ -0.091394639722807 & \frac{1}{2} & 1 \end{bmatrix},$$

$$(\mathcal{B}_8^{\mathrm{Pred}})^{\mathrm{T}} = \begin{bmatrix} 0.052784915021949 & 0 & 0 \\ -0.013966117568827 & 0 & 0 \\ -0.033335451277338 & 0 & 0 \\ 0.005518957037481 & 0 & 0 \\ -0.017219201474101 & 0 & 0 \\ -0.547093076957733 & \frac{1}{2} & 1 \\ 1.004747486833853 & -2 & -2 \\ 0.548562488384716 & \frac{3}{2} & 1 \end{bmatrix}, \quad (\mathcal{B}_8^{\mathrm{Corr}})^{\mathrm{T}} = \begin{bmatrix} -0.025695305445742 & 0 & 0 & 0 \\ 0.024730765685403 & 0 & 0 & 0 \\ 0.046513838588392 & 0 & 0 & 0 \\ 0.015585205908213 & 0 & 0 & 0 \\ -0.122758832623870 & \frac{1}{6} & 0 & -1 \\ -0.021733194382055 & -1 & 1 & 3 \\ 1.148315414068683 & \frac{1}{2} & -2 & -3 \\ -0.064957891799024 & \frac{1}{3} & 1 & 1 \end{bmatrix}.$$

Die relativen Stabilitätsgebiete dieser Verfahren sind in Abbildung 2.15 visualisiert, wobei die relativen Stabilitätsgebiete den Stabilitätsgebieten entsprechen, die durch N_{eval} skaliert sind und sich für einen vom Aufwand unabhängigen Vergleich eignen. Zum besseren Vergleich sind die Stabilitätsgebiete der $\mathrm{PC}[2]_k$-Verfahren (rot) sowie die Stabilitätsgebiete der Adams-Verfahren jeweils für $k = 3$ und $k = 4$ (blau) mit transparenten Konturen dar-

gestellt. Damit stellen wir fest, dass das Optimierungsproblem in Bezug auf das Zielgebiet Λ_{circle} deutlich stabilere Verfahren liefert. Es lässt sich auch erkennen, dass die optimierten Predictor-Corrector-Verfahren die „beste“ Stabilität aufweisen. Die klassischen Stabilitäts-

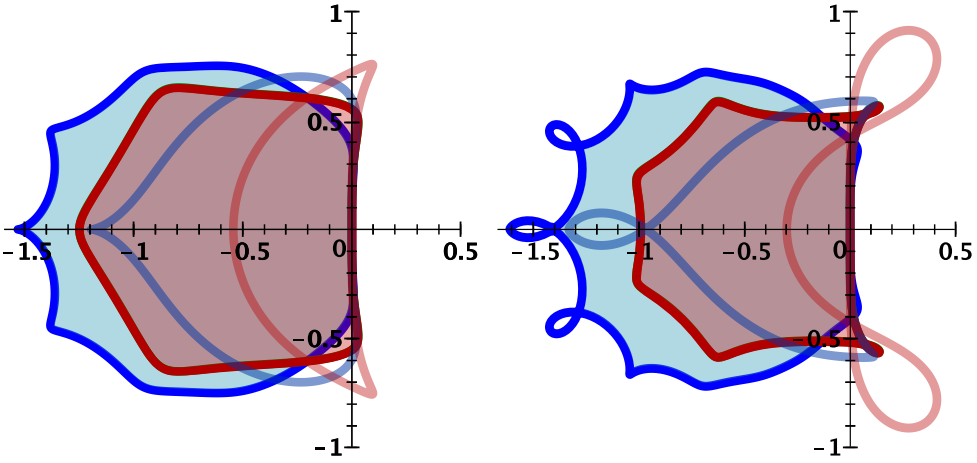

Abbildung 2.15: Stabilitätsgebiete zu EXPPC[1]$_k$ (blau) und EXPMS$_k$ (rot). Links: Mit $k = 6$ und $d = 3$, Rechts: Mit $k = 8$ und $d = 4$.

gebiete entsprechen den Stabilitätskonturen $\mathcal{S}_a$ mit $a = 0$ in der Stabilitätsanalyse von Beylkin (s. Def. 2.15). Es ist wichtig zu prüfen, wie sich die Stabilitätskonturen für unterschiedliche Werte von a verhalten. In Abbildung 2.16 sind für $a = 1$ und $a = 2$ die Stabilitätskonturen $\mathcal{S}_a$ der Verfahren EXPPC[1]$_8$ mit $d = 4$ und EXPMS$_6$ mit $d = 3$ dargestellt. Im Vergleich sind auch die Stabilitätskonturen $\mathcal{S}_a$ des exponentiellen Adams-Verfahrens mit $k = 3$ und des exponentiellen Predictor-Corrector-Verfahrens der zweiten Art mit $k = 3$ in der jeweiligen Farbe mit Transparenz aufgeführt.

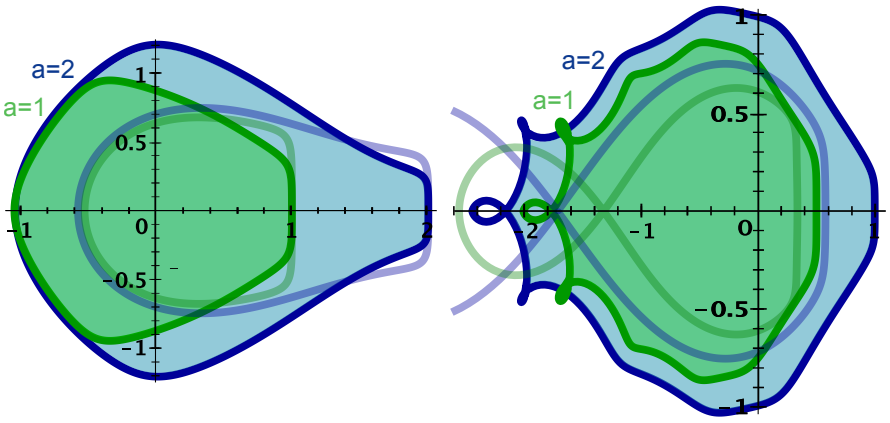

Abbildung 2.16: $\mathcal{S}_a$ mit $a = 1$ (grün) und $a = 2$ (blau). Links: EXPMS$_6$ vs. exp. Adams mit $k = 3$, Rechts: EXPPC[1]$_8$ vs. exp. PECE-Verfahren zweiter Art mit $k = 3$.

Mit der Abbildung 2.16 können wir feststellen, dass die optimierten exponentiellen Mehrschrittverfahren und die optimierten exponentiellen Predictor-Corrector-Verfahren das gleiche Stabilitätsverhalten aufweisen wie die exponentiellen Mehrschrittverfahren aus Beispiel 2.16. Wir sehen auch, dass das Optimierungsproblem die Stabilitätskonturen $\mathcal{S}_a$ für beliebiges $a \geq 0$ vergrößern, denn die Stabilitätskonturen des exponentiellen Predictor-Corrector-Verfahrens zweiter Art mit $k = 3$ sind vollständig im Stabilitätsbereich des optimierten Predictor-Corrector-Verfahrens $\mathrm{PC}[1]_8$ mit $d = 4$ enthalten. Zwar liegen die Stabilitätskonturen des exponentiellen Adams-Verfahrens mit $k = 3$ nicht vollständig im Stabilitätsbereich des optimierten exponentiellen Mehrschrittverfahrens MS_6 mit $d = 3$, doch dies trifft im relevanten Teil der linken Halbebene zu. Mithilfe der Abbildungen 2.15 und 2.16 (vgl. auch Beispiel 2.16) können wir zudem beobachten, dass die Stabilitätskonturen $\mathcal{S}_a$ exponentieller Mehrschrittverfahren für $a \in (0, C)$ mit $C < \infty$ sich verkleinern können. Ein monotones Wachstum der Stabilitätskonturen in der linken Halbebene tritt erst für $a \geq C$ auf (s. Abbildung 2.16), worauf in [5] auch hingewiesen wurde. Hingegen ist bei exponentiellen Predictor-Corrector-Verfahren ein solches Verhalten nicht zu beobachten. Bei exponentiellen Predictor-Corrector-Verfahren wachsen die Stabilitätskonturen $\mathcal{S}_a$ mit dem Parameter $a \geq 0$.

Beispiel 2.23: *(Wärmeleitungsgleichung in 1D)*
In diesem Beispiel werden wir ein Anfangswertproblem zu einer linearen Wärmeleitungsgleichung

$$\frac{\partial}{\partial t} y = \Delta y \quad \text{mit} \quad y(t_0, x) = y_0(x) = \sin\left(\frac{x\pi}{6}\right) \tag{2.75}$$

auf dem Gebiet $\Omega = [0, 6]$ mit homogenen Dirichlet-Randbedingungen betrachten. Die Diskretisierung mit zentralen finiten Differenzen liefert ein lineares gewöhnliches Differentialgleichungssystem

$$y'(t) = Ay(t), \quad y(t_0) = y_0, \tag{2.76}$$

deren exakte Lösung durch

$$y(t) = e^{tA} y_0$$

gegeben ist. Die Diskretisierung A des Laplace-Operators ist dann symmetrisch negativ semidefinit, somit gilt für das Spektrum $\sigma(A) \subset (-\infty, 0]$. In diesem Fall haben wir im

Gegensatz zu (2.74) aus Beispiel 2.22

$$\Lambda_{\text{rectangle}} = \{\, \lambda \in \mathbb{C} \,|\, -6 \leq \text{Re}(\lambda) \leq 0,\ -0.05 \leq \text{Im}(\lambda) \leq 0.05 \,\}$$

als Zielgebiet gewählt und haben mit dem Optimierungsproblem für $4 \leq k \leq 6$ sowohl exponentielle Mehrschrittverfahren als auch exponentielle Predictor-Corrector-Verfahren der Konsistenzordnung $d = 3$ bestimmt. In Abbildung 2.17 haben wir den zeitlichen Fehler auf dem Zeitintervall tspan $= [0, 10]$ in Abhängigkeit der gewählten Zeitschrittweite τ logarithmisch veranschaulicht, wobei wir zur räumlichen Diskretisierung die räumliche Schrittweite $h = 0.3$ verwendet haben.

Aus Abbildung 2.17 erkennen wir, dass die optimierten Verfahren die entsprechende Konvergenzordnung vorweisen, denn die farblichen Linien sind parallel zur schwarzen Hilfslinie zu τ^3. Des Weiteren sehen wir, dass die Predictor-Corrector-Verfahren deutlich stabiler sind als die anderen exponentiellen Mehrschrittverfahren. Bemerkenswert ist, dass sich zusammen mit der Stabilität auch die Fehlerkonstante bei den optimierten Mehrschrittverfahren, die keine Predictor-Corrector-Verfahren sind, mit der Anzahl k der explizit verwendeten Punkte vergrößert. Bei Predictor-Corrector-Verfahren ist ein solches Verhalten dagegen nicht erkennbar, ein besseres Stabilitätsverhalten hat nicht unbedingt eine größere Fehlerkonstante zur Folge.

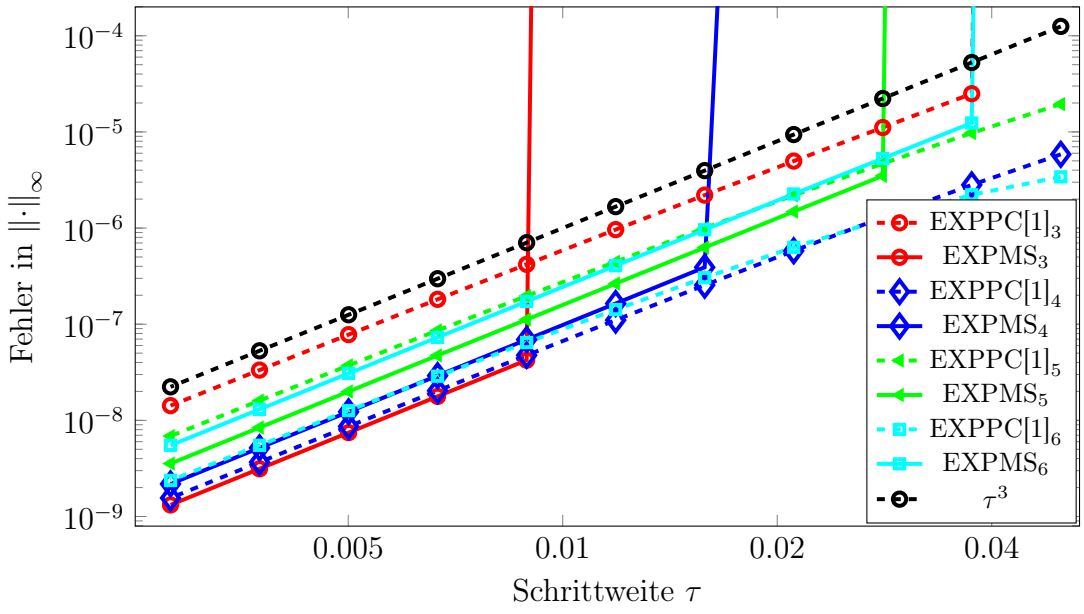

Abbildung 2.17: Logarithmische Darstellung des zeitlichen Fehlers von optimierten exp. Mehrschrittverfahren der Konsistenzordung $d = 3$ in Abhängigkeit der Schrittweite τ.

3 Berechnung von Matrix-Funktionen

Wir haben im vorigen Kapitel gesehen, dass exponentielle Mehrschrittverfahren ein besseres Stabilitätsverhalten besitzen als die klassischen Mehrschrittverfahren. Dieser Vorteil ist natürlich mit entsprechendem Aufwand verbunden, der sich aufgrund der Matrixfunktionen ergibt, worüber die exponentiellen Mehrschrittverfahren (2.43) definiert sind. Die Auswertung von Matrixfunktionen oder die Berechnung von Matrixfunktion-Vektor-Produkten ist die Grundlage für die Anwendbarkeit von exponentiellen Integratoren, doch insbesondere für Matrizen höherer Dimensionen ist dies recht kompliziert. Dieser Aufwand war die Hauptursache dafür, dass diese Integratoren erst mit der Erforschung neuer Berechnungsmöglichkeiten wieder an Aufmerksamkeit gewonnen haben. Demzufolge werden wir uns in diesem Kapitel mit möglichen Realisierungen von exponentiellen Mehrschrittverfahren beschäftigen. Wir erläutern sowohl die Krylov-Verfahren zur Berechnung von Matrixfunktion-Vektor-Produkten als auch multiple time stepping Techniken, die keine Berechnung von Matrixfunktionen erfordern.

3.1 Krylov-Verfahren

Bei der Anwendung exponentieller Integratoren treten üblicherweise Matrixfunktionen der φ-Funktionen (2.23) sowie die Matrixfunktion der Exponentialfunktion (kurz: Matrix-Exponential) auf, deren Linearkombinationen wir bei den exponentiellen Mehrschrittverfahren (2.43) als β_j-Funktionen zusammengefasst haben.

Die Matrixfunktionen zu Matrizen mit einer kleinen Dimension lassen sich durch eine Ähnlichkeitstransformation zur Eigenvektorbasis [39] (für diagonalisierbare Matrizen) oder mithilfe von Padé-Approximationen (beim Matrix-Exponential in Kombination mit einem „Scaling and Squaring“-Verfahren [29]) berechnen. Für weitere Möglichkeiten verweisen wir auf die Arbeit von van Loan und Moler [39], sowie auf die Arbeit von Al-Mohy und Higham [1] und das Buch von Higham [30], in dem diverse Verfahren zur Bestimmung von Matrixfunktionen beschrieben werden.

Wesentlicher für die Lösung zeitabhängiger partieller Differentialgleichungen ist die Berechnung von Matrixfunktionen von Matrizen mit einer großen Dimension. In diesem Fall ist die direkte Berechnung der Matrixfunktionen im Allgemeinen zu aufwendig, aber auch nicht notwendig. Stattdessen reicht es, wie angesprochen Matrixfunktion-Vektor-Produkte zu berechnen und diese für die numerische Approximation eines exponentiellen Integra-

tors zu verwenden. Hierfür stehen inzwischen ebenfalls diverse Verfahren zur Verfügung, wie z.B. die Chebychev-Verfahren [4] (für hermitesche oder schief-hermitesche Matrizen), die Krylov-Verfahren oder die Leja-Interpolation [7, 6]. Wir werden die Berechnung der Matrixfunktion-Vektor-Produkte in dieser Arbeit anhand der Krylov-Verfahren nach [31] und [34] erklären, da wir diese Verfahren in der Implementierung verwendet haben. Für die übrigen Möglichkeiten verweisen wir auf die Arbeit von Hochbruck und Ostermann [34] sowie auf die angegebenen Referenzen.

Wir beginnen mit einigen Notationen. Sei $A \in \mathbb{C}^{n\times n}$ eine Matrix der Dimension $n \times n$ und $u \in \mathbb{C}^n$ ein Vektor der Dimension n. Zudem setzen wir voraus, dass ϕ eine Funktion ist, die in einer offenen Umgebung des Wertebereichs der Matrix A,

$$\mathcal{F}(A) = \left\{ \frac{x^{\mathrm{H}}Ax}{x^{\mathrm{H}}x} \,\middle|\, x \in \mathbb{C}^n,\, x \neq 0 \right\} = \left\{ x^{\mathrm{H}}Ax \,\middle|\, x \in \mathbb{C}^n,\, \|x\| = 1 \right\},$$

analytisch ist. Matrixfunktionen können wir damit über die Cauchy-Integralformel definieren.

Definition 3.1: *(Matrixfunktion)*

Die Matrixfunktion $\phi(A)$ ist definiert durch

$$\phi(A) := \frac{1}{2\pi \mathbb{i}} \int_\Gamma \phi(z)(zI - A)^{-1}\mathrm{d}z, \tag{3.1}$$

wobei ϕ eine Funktion ist, die in einer offenen Umgebung $\mathcal{U}$ von $\mathcal{F}(A)$ analytisch ist und Γ eine positiv orientierte Kurve ist, die den Rand einer offenen Umgebung $U \subset \mathcal{U}$ des Wertebereichs $\mathcal{F}(A)$ mit $\mathcal{F}(A) \subsetneq U$ beschreibt.

Wir möchten an dieser Stelle bemerken, dass wir uns zur Einfachheit auf den Matrixfall beschränken, obwohl die Cauchy-Integralformel für Operatoren gültig ist. Speziell bei den exponentiellen Integratoren dieser Arbeit interessieren wir uns für Matrixfunktion-Vektor-Produkte der Art $\phi(A)u = \varphi_k(A)u$ mit $k \geq 0$ (2.23). Mit der Definition 3.1 ergibt sich für diese Matrixfunktion-Vektor-Produkte die Darstellung

$$\phi(A)u = \frac{1}{2\pi \mathbb{i}} \int_\Gamma \phi(z)(zI - A)^{-1}u \; \mathrm{d}z. \tag{3.2}$$

Definieren wir mit $x(z)$ die Lösung des Gleichungssystems

$$(zI - A)x(z) = u \tag{3.3}$$

für alle $z \in \Gamma$, so ergibt sich für (3.2) die nachfolgende Beziehung

$$\phi(A)u = \frac{1}{2\pi\mathrm{i}} \int_\Gamma \phi(z)x(z)\ \mathrm{d}z. \tag{3.4}$$

Wegen (3.3) können wir ein Krylov-Verfahren zur Approximation der Lösung dieses linearen Gleichungssystems verwenden. Wir definieren hierfür den Krylov-Raum $\mathcal{K}_m$ der Dimension m zu (3.3), der durch die Vektoren $(zI - A)^j u$ mit $0 \leq j \leq m-1$ aufgespannt wird, d.h. es gilt

$$\mathcal{K}_m(zI - A, u) = \mathrm{span}\{\, u, (zI - A)u, \ldots, (zI - A)^{m-1}u \,\}.$$

Offensichtlich ist

$$\mathcal{K}_m(A, u) = \mathcal{K}_m(zI - A, u) \tag{3.5}$$

erfüllt, was zur Folge hat, dass eine Basis $V_m \in \mathbb{C}^{n\times m}$ von $\mathcal{K}_m(A, u)$ auch eine Basis von $\mathcal{K}_m(zI - A, u)$ für jedes $z \in \mathbb{C}$ ist. Demzufolge bestimmen wir mit dem Arnoldi-Verfahren [31] eine Orthonormalbasis V_m von $\mathcal{K}_m(A, u)$, die die folgende Beziehung erfüllt

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^{\mathrm{T}} \quad \text{mit} \quad V_m^{\mathrm{H}} V_m = I_m \quad \text{und} \quad u = \|u\|\, V_m e_1, \tag{3.6}$$

wobei H_m eine obere Hessenberg-Matrix und I_m die Einheitsmatrix der Dimension $m \times m$ ist und e_m den m-ten Einheitsvektor mit der passenden Dimension darstellt.

Mithilfe von (3.6) erhalten wir die Gleichung

$$\begin{aligned}(zI - A)V_m &= zV_m - AV_m \\ &\overset{(3.6)}{=} V_m(zI_m - H_m) - h_{m+1,m} v_{m+1} e_m^{\mathrm{T}},\end{aligned} \tag{3.7}$$

die uns ermöglicht $V_m^{\mathrm{H}}(zI - A)V_m$ durch

$$V_m^{\mathrm{H}}(zI - A)V_m \overset{(3.6)}{=} zI_m - H_m \tag{3.8}$$

umzuschreiben. Anhand von (3.6) und (3.8) können wir dann die Galerkin-Iterierte $x_m(z) := V_m y_m(z) \in \mathcal{K}_m(A,u)$ mit $y_m(z) \in \mathbb{C}^m$ als Approximation zu (3.3) bestimmen, indem wir die Orthogonalitätsbedingung an das Residuum

$$r_m := u - (z\mathbf{I} - A)x_m(z)$$

mit $r_m \perp \mathcal{K}_m(A,u)$ fordern. Die Galerkin-Iterierte zu (3.3) entspricht damit dem Ausdruck

$$\begin{aligned} r_m \perp \mathcal{K}_m(A,u) \quad &\iff \quad V_m^{\mathrm{H}}(zI - A)x_m(z) = V_m^{\mathrm{H}}u \\ &\iff \quad (zI_m - H_m)y_m(z) = \|u\|\, e_1 \\ &\iff \quad y_m(z) = (zI_m - H_m)^{-1}\,\|u\|\, e_1 \\ &\implies \quad x_m(z) = V_m(zI_m - H_m)^{-1}\,\|u\|\, e_1. \end{aligned} \tag{3.9}$$

Wegen

$$\begin{aligned} \mathcal{F}(H_m) &= \left\{\, y^{\mathrm{H}}H_m y \;\middle|\; y \in \mathbb{C}^m,\, \|y\| = 1 \,\right\} \\ &\overset{(3.6)}{=} \left\{\, y^{\mathrm{H}}V_m^{\mathrm{H}}AV_m y \;\middle|\; y \in \mathbb{C}^m,\, \|y\| = 1 \,\right\} \\ &= \left\{\, x^{\mathrm{H}}Ax \;\middle|\; x \in \mathcal{K}_m(A,u),\, \|x\| = 1 \,\right\} \\ &\subset \mathcal{F}(A) \end{aligned}$$

existieren die Galerkin-Iterierten für jedes m, da die Eigenwerte nicht auf der Kurve Γ aus Definition 3.1 liegen können. Eine Approximation des Matrixfunktion-Vektor-Produkts (3.4) lässt sich schließlich durch Einsetzen von $x_m(z) \approx x(z)$ aus (3.9) in (3.4) bestimmen

$$\begin{aligned} \phi(A)u &\approx \frac{1}{2\pi\mathrm{i}} \int_\Gamma \phi(z)x_m(z)\,\mathrm{d}z \\ &\overset{(3.9)}{=} \frac{1}{2\pi\mathrm{i}} \int_\Gamma \phi(z)V_m(zI_m - H_m)^{-1}\,\|u\|\, e_1\,\mathrm{d}z \\ &= V_m \left(\frac{1}{2\pi\mathrm{i}} \int_\Gamma \phi(z)(zI_m - H_m)^{-1}\mathrm{d}z \right) \|u\|\, e_1 \\ &= V_m\phi(H_m)\,\|u\|\, e_1. \end{aligned} \tag{3.10}$$

Wir sehen, dass die Approximation eines Matrixfunktion-Vektor-Produkts sich auf die Berechnung einer Matrixfunktion zurückführen lässt. Der Vorteil besteht darin, dass die Matrix H_m im Vergleich zu A von der Dimension her recht klein ist. Die zugehörige Matrixfunktion lässt sich dann mit Hilfe der genannten Verfahren berechnen, die sich zur Approximation von Matrixfunktionen kleiner Matrizen eignen.

Die Konvergenz von Krylov-Verfahren ist weitgehend analysiert worden durch die Arbeiten von Saad [49], Saad und Gallopoulos [18], Druskin und Knizhnerman [12, 13, 14], Hochbruck und Lubich [32]. Besonders erwähnenswert ist jedoch, dass die Krylov-Verfahren superlinear konvergieren und die Konvergenz erst ab einer bestimmten Dimension des Krylov-Raums eintritt, die von $\|A\|$ abhängig ist.

Bessere Konvergenzeigenschaften weisen rationale Krylov-Verfahren auf, denn deren Konvergenz ist unabhängig von $\|A\|$. Der Unterschied liegt darin, dass rationale Krylov-Verfahren zur Approximation von (3.3) einen Krylov-Raum der Form

$$\mathcal{K}_m((I+\gamma A)^{-1}, u) = \operatorname{span}\{\, u, (I+\gamma A)^{-1}u, \ldots, (I+\gamma A)^{-(m-1)}u \,\}, \quad \gamma > 0$$

verwenden. Der Preis, den wir für die bessere Konvergenzeigenschaft zahlen müssen, wird hier deutlich, da wir zur Konstruktion einer Orthonormalbasis in jedem Schritt ein Gleichungssystem lösen müssen. Bestimmen wir mit dem Arnoldi-Algorithmus auch in diesem Fall eine Orthonormalbasis $\widehat{V}_m$ dieses Krylov-Raums, so gilt die Eigenschaft

$$(I+\gamma A)^{-1}\widehat{V}_m = \widehat{V}_m\widehat{H}_m + \widehat{h}_{m+1,m}\widehat{v}_{m+1}e_m^{\mathrm{T}} \text{ mit } \widehat{V}_m^{\mathrm{H}}\widehat{V}_m = I_m \text{ und } u = \|u\|\,\widehat{V}_m e_1, \quad (3.11)$$

wobei die Matrix $\widehat{H}_m$ erneut eine obere Hessenberg-Matrix ist. Wenn wir $B := (I+\gamma A)^{-1}$ definieren, so erhalten wir nach (3.2) die Darstellung

$$\psi(B)u = \frac{1}{2\pi\mathbb{i}}\int_\Gamma \psi(z)(zI-B)^{-1}u\mathrm{d}z \quad (3.12)$$

für eine Funktion ψ. Mit derselben Theorie (3.3)-(3.10) würden wir in diesem Fall eine Galerkin-Iterierte $x_m(z) \in \mathcal{K}_m(B,u)$ erhalten mit

$$x_m(z) = \widehat{V}_m(zI_m - \widehat{H}_m)^{-1}\,\|u\|\,e_1 \quad \text{und} \quad \psi(B)u = \widehat{V}_m\psi(\widehat{H}_m)\,\|u\|\,e_1. \quad (3.13)$$

Um eine Approximation an $\phi(A)u$ zu erhalten, wählen wir

$$\psi(z) := \phi\left(\frac{z^{-1}-1}{\gamma}\right). \quad (3.14)$$

Mit (3.14) erhalten wir schließlich die folgende Approximation

$$\phi(A)u \approx \widehat{V}_m\phi(\widetilde{H}_m)\,\|u\|\,e_1 \quad \text{mit} \quad \widetilde{H}_m := \frac{1}{\gamma}(\widehat{H}_m^{-1} - I). \quad (3.15)$$

Wir möchten auf die Wahl des Parameters γ nicht im Detail eingehen, doch es ist möglich γ optimal zu einer gegebenen Funktion und einer gegebenen Toleranz zu wählen. Für hermitesche Matrizen haben Van den Eshof und Hochbruck in [51] die optimalen γ-Werte zur Exponentialfunktion vorgestellt. Ebenfalls für hermitesche Matrizen wurden die optimalen γ-Werte für trigonometrische Funktionen durch Grimm und Hochbruck [20] bestimmt. Der Vorteil einer optimalen Wahl vom Parameter γ, die auf a-priori-Fehlerabschätzungen beruht, besteht darin, dass die gewünschte Toleranz mit möglichst wenigen Krylov-Schritten erreicht wird.

Von größerer Bedeutung ist die Verwendung eines Residuumschätzers, der sich anhand von a-posteriori-Fehlerabschätzungen herleiten lässt. Die Residuenschätzer stellen ein Maß für den Fehler dar mit deren Verwendung wir in der Lage sind, eine Approximation zu einer gegebenen Toleranz zu bestimmen. In unserer Implementierung haben wir uns für das folgende Residuum von Saad [49]

$$\|r_m\| = h_{m+1,m} \|u\| \left| e_m^{\mathrm{T}} \phi(H_m) e_1 \right| \quad \text{bzw.} \quad \|r_m\| = h_{m+1,m} \|u\| \left| e_m^{\mathrm{T}} \phi(\widetilde{H}_m) e_1 \right|$$

entschieden. Prinzipiell ist damit die Berechnung von Matrixfunktionen mit Krylov-Verfahren geklärt, doch in der Praxis kann es vorkommen, dass die gewünschte Toleranz erst nach vielen Krylov-Schritten erreicht wird. Der Aufwand ist dann oft zu hoch, denn mit jedem Schritt steigt der Aufwand im Arnoldi-Algorithmus bei der Konstruktion der Orthonormalbasis. In diesem Fall stellen Lanczos-Verfahren [31] eine Alternative dar, wo der Aufwand in jedem Schritt konstant ist. Des Weiteren können die „restarted"-Varianten der Krylov-Verfahren [15, 45] verwendet werden, die nur endlich viele Krylov-Schritte ausführen. Sollte die gewünschte Toleranz nicht erreicht werden, so wird das Krylov-Verfahren erneut aufgerufen. Allerdings wirkt sich diese Alternative negativ auf die Konvergenzgeschwindigkeit aus.

Abschließend möchten wir ein wichtiges Resultat von Higham und Al-Mohy [2] aufführen, das speziell für exponentielle Integratoren sehr interessant ist.

Satz 3.2: *(Linearkombinationen von φ-Funktionen)*

Sei $A \in \mathbb{C}^{n\times n}$, $W = [w_1, \ldots, w_p] \in \mathbb{C}^{n\times p}$ und $\tau \in \mathbb{C}$, sowie

$$\widetilde{A} = \begin{bmatrix} A & W \\ 0 & J \end{bmatrix} \in \mathbb{C}^{n+p\times n+p} \quad \textit{und} \quad J = \begin{bmatrix} 0 & I_{p-1} \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{p\times p},$$

dann gilt für $X = \varphi_l(\tau\widetilde{A})$ *mit* $l \geq 0$

$$X(1:n, n+j) = \sum_{k=1}^{j} \tau^k \varphi_{l+k}(\tau A) w_{j-k+1}, \quad j = 1, \ldots, p.$$

Der Satz 3.2 zeigt, dass die Linearkombination von Matrixfunktion-Vektor-Produkten zu φ-Funktionen auf ein einziges Matrixfunktion-Vektor-Produkt zu einer größeren Matrix $\widetilde{A}$ zurückgeführt werden kann. Niesen [46] hat anhand dieser Idee einen effizienten Algorithmus entwickelt, der ebenfalls auf Krylov-Verfahren beruht.

3.2 Multiple time stepping

In diesem Abschnitt werden wir auf der Basis der Idee von Hochbruck und Ostermann [33] eine alternative Umsetzung von exponentiellen Mehrschrittverfahren beschreiben, die keine Auswertung von Matrixfunktionen erfordert. Mit der Notation aus (1.5) betrachten wir

$$y' = f_{\text{fine}}(t, y) + f_{\text{coarse}}(t, y) = Ay + g(t, y).$$

Wir haben darüberhinaus vorausgesetzt, dass das Anfangswertproblem aus einer räumlichen Diskretisierung einer partiellen Differentialgleichung über einem Gebiet Ω resultiert und die Diskretisierung des Gebiets Ω eine lokale Verfeinerung aufweist. In diesem Fall ist sogar nur ein kleiner Anteil von A für die Steifigkeit des Anfangswertproblems verantwortlich. Aus diesem Grund werden wir eine umformulierte Version des semilinearen Anfangswertproblems (2.1) betrachten, das sich allgemein wie folgt beschreiben lässt

$$\begin{aligned} y' &= My + \mathbf{g}(t, y), \quad y(t_0) = y_0 \\ \text{mit} \quad \mathbf{g}(t, y) &= (A - M)y + g(t, y) \quad \text{und} \quad \|A\| \gg \|A - M\|. \end{aligned} \tag{3.16}$$

Mit einer solchen Umformulierung können beispielsweise die Matrixfunktionen mit geringerem Rechenaufwand berechnet werden. Aus der theoretischer Perspektive bleibt die Klasse des Anfangswertproblems unverändert, da (3.16) ebenfalls ein semilineares Anfangswertproblem darstellt. Wir möchten im Folgenden zuerst das grundlegende Prinzip der multiple time stepping Varianten exponentieller Mehrschrittverfahren präsentieren und im Anschluss mit der Wahl von M den Bezug zu bekannten Verfahren herstellen.

3.2.1 Multiple time stepping Varianten exponentieller Mehrschrittverfahren

Die multiple time stepping Varianten werden wir zuerst für allgemeine exponentielle Mehrschrittverfahren der Form (2.38) und anschließend auch für die allgemeinen exponentiellen Predictor-Corrector-Verfahren aus Abschnitt 2.2.4 präsentieren. Die Approximation eines allgemeinen exponentiellen Mehrschrittverfahrens der Form (2.38) erhalten wir, indem wir das Polynom p_n aus (2.36) in (2.22) einsetzen. Für die Approximation der Lösung des Anfangswertproblems (3.16) ergibt sich damit die folgende Darstellung

$$y(t_n + \tau) \approx y_{n+1} = e^{\tau M} y_n + \tau \int_0^1 e^{(1-\theta)\tau M} p_n(t_n + \theta\tau) \mathrm{d}\theta. \tag{3.17}$$

An dieser Stelle folgen wir der Idee von Hochbruck und Ostermann [33] und erhalten mit Hilfe der VdK-Formel („rückwärts“ angewendet) das zugehörige Anfangswertproblem

$$\mathbf{y}_n' = M\mathbf{y}_n + p_n(t_n + t), \;\; \mathbf{y}_n(0) = y_n \;\text{ mit }\; t \in [0, \tau] \;\text{ und }\; \mathbf{y}_n(\tau) = y_{n+1}. \tag{3.18}$$

Wir sehen, dass die Approximation des exponentiellen Mehrschrittverfahrens (3.17) der exakten Lösung des Anfangswertproblems (3.18) entspricht. Doch statt der Bestimmung der exakten Lösung von (3.18) anhand von Matrixfunktionen können wir dieses Anfangswertproblem mit einem weiteren numerischen Integrator mit der konstanten Schrittweite $\tau_{\text{inner}} := \frac{\tau}{r}$ für ein festes $r \in \mathbb{N}$ lösen. Wir bezeichnen die auf diese Art bestimmte Approximation mit

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n(\tau) = y_{n+1} \approx y(t_n + \tau). \tag{3.19}$$

Des Weiteren ist klar, dass diese Verfahren nur dann effizient sind, wenn die Auswertung von p_n deutlich weniger aufwändig ist als die Auswertung der Funktion $\mathbf{g}$, da die Anfangswertprobleme (3.16) und (3.18) sich nur darin unterscheiden. Zur Beschreibung eines solchen multiple time stepping Verfahrens ist es notwendig, sowohl das zugrunde liegende exponentielle Mehrschrittverfahren als auch den numerischen Integrator anzugeben, der zur Bestimmung der Approximation $\mathbf{y}_{n+1}$ angewendet wurde. Hierzu werden wir die Bezeichnungen äußerer Integrator und innerer Integrator verwenden und mit $\text{MTS} - \text{EXPMS}_k^d$ ein multiple time stepping Verfahren bezeichnen, das ein allgemeines exponentielles k-Schrittverfahren der Konsistenzordnung d als äußeren Integrator besitzt. Die Wahl des inneren Integrators stellen wir frei und werden diese zusätzliche Information bei der Anwendung immer angeben.

Nachdem die Funktionsweise der MTS – EXPMS$_k^d$-Verfahren geklärt ist, stellt sich die Frage, wie die Konvergenz eines solchen Verfahrens ist. Wir werden zunächst notwendige Voraussetzungen formulieren, um mit Hilfe der Fehleranalyse aus Abschnitt 2.2.5 Konvergenzresultate zu beweisen.

Voraussetzung 3.3: *(Splitting-Konfiguration für parabolische Probleme)*

M erfülle die Voraussetzung 2.10 a) sowie 2.10 b) mit $\alpha = 0$ und $V = X$, so dass (2.46) mit $\gamma = 0$ und einer Konstanten C_P erfüllt ist.

Voraussetzung 3.4: *(Splitting-Konfiguration für hyperbolische Probleme)*

M erfülle die Voraussetzung 2.12, so dass (2.45) mit einer Konstanten $C_M > 0$ (statt C_A) und $\omega = 0$ erfüllt ist.

Voraussetzung 3.5: *(Innerer Integrator mit Schrittweite τ_{inner})*

Für die Approximation $\mathbf{y}_j \approx y_j = \mathbf{y}_{j-1}(\tau)$ des inneren Integrators zum Anfangswertproblem (3.18) gilt

$$\|\mathbf{y}_j - y_j\| \leq C_I \tau^{d+1} \quad \textit{mit} \quad C_I = C_I(M, r, d, T), \tag{3.20}$$

sofern p_n die Ordnungsbedingungen (Satz 2.13) für $l = 0, \ldots, d-1$ erfüllt.

Die Voraussetzungen 3.3 und 3.4 zu (3.16) entsprechen den funktionalanalytischen Voraussetzungen 2.10, 2.9 und 2.12. Wir erlauben damit die Anwendung expliziter innerer Integratoren, soweit die Konstante C_I für ein fest gewähltes $r \in \mathbb{N}$ beschränkt bleibt. Zur besseren Unterscheidung werden wir in den folgenden Beweisen die Konstanten, die von M abhängig sind in roter Farbe notieren.

Satz 3.6: *(Konvergenz von* MTS – EXPMS$_k^d$*)*

Das Anfangswertproblem (3.16) erfülle die Voraussetzung 3.3 und der innere Integrator des MTS – EXPMS$_k^d$*-Verfahrens genüge der Voraussetzung 3.5. Zudem sei die Funktion G mit $G(t) := \mathbf{g}\big(t, y(t)\big)$ hinreichend glatt mit $G \in C^d([0,T], X)$ und die Schrittweite τ sei beschränkt durch eine hinreichend kleine Konstante $\mathcal{T}$ mit $0 \leq \tau \leq \mathcal{T}$. Dann ist der Fehler der numerischen Approximation zur Schrittweite τ des* MTS – EXPMS$_k^d$*-Verfahrens mit $k \geq d$ für hinreichend genaue Startwerte*

$$\|\mathbf{y}_j - y(t_j)\| \leq c_0 \tau^d, \quad j = 1, \ldots, k-1 \tag{3.21}$$

gleichmäßig beschränkt durch

$$\|\mathbf{y}_n - y(t_n)\| \leq C\tau^d$$

für $0 \leq n\tau \leq T$ *mit einer Konstante* $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_{\mathbf{g}}, \mathbf{g}, T, c_0, C_I)$*, die unabhängig ist von* n *und* τ*.*

Beweis: *(Satz 3.6)*
Wegen $G(t) := \mathbf{g}(t, y(t))$ definieren wir das Polynom (2.56) zu den exakten Daten $\big(t_j, G_j\big)$ mit $n-k+1 \leq j \leq n$, $G_j := G(t_j)$ und $t_j := j\tau$. Mit der VdK-Formel erhalten wir (s. Beweis zu Satz 2.19)

$$y(t_{n+1}) = e^{\tau M} y(t_n) + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau M) G_{n-k+1+j} + \delta_{n+1} \tag{3.22}$$

als Darstellung der exakten Lösung von (3.16), wobei der Defekt durch

$$\begin{aligned} \delta_{n+1} &:= \tau \int_0^1 e^{(1-\theta)\tau M} \big(G(t_n + \theta\tau) - \tilde{p}_n(t_n + \theta\tau)\big) \\ \text{mit} \quad \|\delta_{n+1}\| &\overset{(2.57)}{\leq} C\tau^{d+1} \sup_{0 \leq t \leq t_{n+1}} \big\|G^{(d)}(t)\big\| \end{aligned} \tag{3.23}$$

beschrieben wird und die Konstante C nur von C_P und $C_{\tilde{P}}$ abhängig ist. Definieren wir den Fehler des MTS – $\mathrm{EXPMS}_{\mathrm{k}}^{\mathrm{d}}$-Verfahrens zur Zeit t_n durch $e_n := \mathbf{y}_n - y(t_n)$, so ergibt sich mit der Dreiecksungleichung die folgende Beziehung

$$\|e_n\| = \|\mathbf{y}_n - y(t_n)\| \leq \|\mathbf{y}_n - y_n\| + \|y_n - y(t_n)\|\,,$$

wobei

$$y_n = e^{\tau M} \mathbf{y}_{n-1} + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau M) \mathbf{g}_{n-k+j} \tag{3.24}$$

mit $\mathbf{g}_i := \mathbf{g}(t_i, \mathbf{y}_i)$ für $n-k+1 \leq i \leq n$ gilt. Der Ausdruck $\|\mathbf{y}_n - y_n\|$ stellt den Fehler des inneren Integrators dar und kann nach Voraussetzung 3.5 abgeschätzt werden, so dass folgende Ungleichung gilt

$$\|e_n\| \leq C_I \tau^{d+1} + \|y_n - y(t_n)\|\,.$$

Darüber hinaus ergibt sich aus der Differenz von (3.22) und (3.24) für $n+1$ die Fehlerrekursion

$$\begin{aligned} e_{n+1} &= y_{n+1} - y(t_{n+1}) + \mathbf{y}_{n+1} - y_{n+1} \\ &= e^{\tau M} e_n + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau M)\big(\mathbf{g}_{n-k+1+j} - G_{n-k+1+j}\big) - \delta_{n+1} + \mathbf{y}_{n+1} - y_{n+1}, \end{aligned}$$

deren Auflösung die folgende Darstellung des Fehlers liefert

$$e_n = \tau \sum_{l=0}^{n-1} e^{(n-l-1)\tau M} \left(\sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau M)\big(\mathbf{g}_{l-k+1+j} - G_{l-k+1+j}\big) + \frac{1}{\tau}(\mathbf{y}_{l+1} - y_{l+1} - \delta_{l+1}) \right).$$

Zur Abschätzung des Fehlers verwenden wir für $l \leq n$ Abschätzungen der Form

$$\left\| e^{(n-l)\tau M}\big(\mathbf{g}_j - G_j\big) \right\| \leq C_P \mathcal{L}_{\mathbf{g}} \left\| e_j \right\|, \tag{3.25}$$

die das Analogon zu (2.59) darstellen. Den Fehler können wir damit wie folgt abschätzen

$$\begin{aligned} \|e_n\| &\leq \tau \sum_{l=0}^{n-1} \sum_{j=0}^{k-1} \left\| \beta_j^{\mathrm{eM}}(\tau M) \right\| \left\| e^{(n-l-1)\tau M}(\mathbf{g}_{l-k+1+j} - G_{l-k+1+j}) \right\| \\ &\quad + \sum_{l=0}^{n-1} \left\| e^{(n-l-1)\tau M}(\mathbf{y}_{l+1} - y_{l+1}) \right\| + \sum_{l=0}^{n-1} \left\| e^{(n-l-1)\tau M} \delta_{l+1} \right\| \\ &\overset{(3.25)}{\leq} \tau C \sum_{l=0}^{n-1} \sum_{j=0}^{k-1} \|e_{l-k+1+j}\| + C \sum_{l=0}^{n-1} \|\mathbf{y}_{l+1} - y_{l+1}\| + C \sum_{l=0}^{n-1} \|\delta_{l+1}\| \\ &\leq \tau C \sum_{l=0}^{n-1} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\| + \frac{\|\delta_{l+1}\|}{\tau} \right) + C \sum_{l=0}^{n-1} \|\mathbf{y}_{l+1} - y_{l+1}\| \\ &\leq \tau C \sum_{l=0}^{n-1} (\|e_l\| + \tau^d) + C_P C_I \sum_{l=0}^{n-1} \tau^{d+1} \\ &\leq C \max_{l=1,\ldots,k-1} \|e_l\| + \tau C \sum_{l=0}^{n-1} (\|e_l\| + \tau^d) + C_P C_I \tau^d. \end{aligned}$$

Wie in den Beweisen zu Satz 2.19 und 2.20 gilt für die Konstante $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_{\mathbf{g}}, \mathbf{g})$. Mit den Abschätzungen (3.21) und (3.23) stellen wir fest, dass die Folge $(\varepsilon_j)_{j \in \mathbb{N}}$ mit den

Folgegliedern $\varepsilon_j := \|e_j\|_V$ die Abschätzung

$$\varepsilon_n \leq C\tau \sum_{l=1}^{n-1} \varepsilon_l + C\tau^d$$

erfüllt. Die Behauptung folgt dann aus der Anwendung des Gronwall-Lemmas 2.17 mit einer Konstante $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_{\mathbf{g}}, \mathbf{g}, T, c_0, C_I)$. □

Die multiple time stepping Varianten zu exponentiellen Predictor-Corrector-Verfahren können auf dieselbe Art konstruiert werden. Die Beschreibung dieser Verfahren werden wir erneut mit den Prozessen Prediction, Evaluation und Correction vornehmen.

Prediction: (P)
Das Predictor-Verfahren entspricht einem MTS − EXPMS$_k^d$-Verfahren und liefert die Approximation

$$y(t_{n+1}) \approx \bar{\mathbf{y}}_{n+1}. \tag{3.26}$$

Evaluation: (E)
Mit der Auswertung der Funktion $\mathbf{g}$ mit (3.26) erhalten wir

$$\mathbf{g}\big(t_{n+1}, y(t_{n+1})\big) \approx \mathbf{g}(t_{n+1}, \bar{\mathbf{y}}_{n+1}) =: \bar{\mathbf{g}}_{n+1}.$$

Correction: (C)
In diesem Prozess wird ein allgemeines implizites exponentielles Mehrschrittverfahren als äußerer Integrator gewählt. Hierzu wird zuerst ein Polynom q_n der Form (2.37) in (2.22) eingesetzt. Anschließend erhalten wir analog zu (3.18) mit der VdK-Formel das zugehörige Anfangswertproblem

$$\mathbf{y}_n' = M\mathbf{y}_n + q_n(t_n + t), \;\; \mathbf{y}_n(0) = y_n \;\text{ mit }\; t \in [0, \tau] \;\text{ und }\; \mathbf{y}_n(\tau) = y_{n+1}. \tag{3.27}$$

Die Verwendung eines inneren Integrators liefert uns schließlich die Approximation

$$\mathbf{y}_{n+1} \approx \mathbf{y}(\tau) \approx y(t_n + \tau). \tag{3.28}$$

Wir notieren kurz mit $\mathrm{MTS}-\mathrm{EXPPC}[1]_k^{(d_\mathrm{P},d_\mathrm{C})}$ ein multiple time stepping Predictor-Corrector-Verfahren der ersten Art, das aus einem allgemeinen expliziten exponentiellen k-Schrittverfahren der Konsistenzordnung d_P als äußeren Integrator des Predictor-Verfahrens und einem allgemeinen impliziten exponentiellen k-Schrittverfahren der Konsistenzordnung d_C als äußeren Integrator des Corrector-Verfahrens entsteht. Für Predictor-Corrector-Verfahren der zweiten Art schreiben wir kurz $\mathrm{MTS}-\mathrm{EXPPC}[2]_k^{(d_\mathrm{P},d_\mathrm{C})}$. Der äußere Integrator des Corrector-Verfahrens entspricht in diesem Fall einem allgemeinen impliziten exponentiellen $(k+1)$-Schrittverfahren der Konsistenzordnung d_C. Die Konvergenz dieser Verfahren ist im folgenden Satz formuliert.

Satz 3.7: *(Konvergenz von multiple time stepping Predictor-Corrector-Verfahren)*

a) *Betrachten wir unter den Voraussetzungen von Satz 3.6 ein* $\mathrm{MTS}-\mathrm{EXPPC}[\mathbf{n}]_k^{(d,d)}$*-Verfahren* ($\mathbf{n}=1,2$) *mit* $k\geq d$ *und hinreichend genauen Startwerten*

$$\|\mathbf{y}_j - y(t_j)\| \leq c_0\tau^d, \quad j=1,\ldots,k-1, \tag{3.29}$$

so ist der Fehler der numerischen Lösung eines solchen Verfahrens zur Schrittweite τ *gleichmäßig beschränkt durch*

$$\|\mathbf{y}_n - y(t_n)\| \leq C\tau^d$$

für $0\leq n\tau\leq T$ *mit einer Konstante* $C=C(C_P,k,d,\mathcal{B},\mathcal{L}_\mathbf{g},\mathbf{g},T,c_0,C_I)$*, die unabhängig ist von* n *und* τ*.*

b) *Im Vergleich zur Aussage in a) setzen wir stärker voraus, dass* $G\in C^{d+1}([0,T],X)$ *gilt. Der Fehler der numerischen Approximation des* $\mathrm{MTS}-\mathrm{EXPPC}[\mathbf{n}]_k^{(d,d+1)}$*-Verfahrens mit* $\mathbf{n}=1,2$ *ist dann für hinreichend genaue Startwerte*

$$\|\mathbf{y}_j - y(t_j)\| \leq c_0\tau^{d+1}, \quad j=1,\ldots,k-1 \tag{3.30}$$

gleichmäßig beschränkt durch

$$\|\mathbf{y}_n - y(t_n)\| \leq C\tau^{d+1}$$

für $0\leq n\tau\leq T$ *mit einer Konstante* $C=C(C_P,k,d,\mathcal{B},\mathcal{L}_\mathbf{g},\mathbf{g},T,c_0,C_I)$*, die unabhängig ist von* n *und* τ*.*

Beweis: *(Satz 3.7)*
Teil a): Aufgrund der Analogie reicht es, den Beweis anhand der MTS – EXPPC$[2]_k^{(d,d)}$-Verfahren zu führen. Wegen $G(t) := \mathbf{g}(t, y(t))$ definieren wir die Polynome zu den exakten Daten $\big(t_j, G_j\big)$ mit $n-k+1 \le j \le n$, $G_j := G(t_j)$ und $t_j := j\tau$ als die Polynome (2.64). Mit der VdK-Formel (s. Beweis zu Satz 2.20) können wir damit die exakte Lösung zum einen durch

$$y(t_{n+1}) = e^{\tau M} y(t_n) + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau M) G_{n-k+1+j} + \delta_{n+1}$$

und zum anderen durch

$$y(t_{n+1}) = e^{\tau M} y(t_n) + \tau \sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau M) G_{n-k+1+j} + \psi_{n+1}$$

mit den Defekten

$$\begin{aligned} \delta_{n+1} &:= \tau \int_0^1 e^{(1-\theta)\tau M} \big(G(t_n + \theta\tau) - \tilde{p}_n(t_n + \theta\tau)\big) \\ \psi_{n+1} &:= \tau \int_0^1 e^{(1-\theta)\tau M} \big(G(t_n + \theta\tau) - \tilde{q}_n(t_n + \theta\tau)\big) \end{aligned}$$

beschreiben. Die Defekte können wir wie im Beweis von Satz 2.20 abschätzen durch

$$\begin{aligned} \|\delta_{n+1}\| &\le C\tau^{d+1} \sup_{0 \le t \le t_{n+1}} \left\|G^{(d)}(t)\right\|, \\ \|\psi_{n+1}\| &\le C\tau^{d+1} \sup_{0 \le t \le t_{n+1}} \left\|G^{(d)}(t)\right\|, \end{aligned} \tag{3.31}$$

wobei für die Konstanten $C = C(C_P, C_{\widetilde{P}})$ gilt. Wir definieren mit $e_n := \mathbf{y}_n - y(t_n)$ den Fehler eines MTS – EXPPC$[2]_k^{(d,d)}$-Verfahrens und notieren mit

$$\begin{aligned} \bar{y}_n &= e^{\tau M} \mathbf{y}_{n-1} + \tau \sum_{j=0}^{k-1} \beta_j^{\mathrm{eM}}(\tau M) \mathbf{g}_{n-k+j} \\ y_n &= e^{\tau M} \mathbf{y}_{n-1} + \tau \sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau M) \mathbf{g}^*_{n-k+j}, \end{aligned}$$

die Approximationen des Predictor- und Corrector-Verfahrens mit exakter Berechnung

der Matrixfunktionen, wobei

$$\mathbf{g}_i^* := \begin{cases} \mathbf{g}_i := \mathbf{g}(t_i, \mathbf{y}_i), & i \neq k \\ \bar{\mathbf{g}}_i := \mathbf{g}(t_i, \bar{\mathbf{y}}_i), & i = k \end{cases} \quad \text{für} \quad n-k+1 \leq i \leq n+1$$

gilt. Die Voraussetzung 3.5 liefert zusammen mit der Dreiecksungleichung die Abschätzungen

$$\begin{aligned} \|\bar{\mathbf{y}}_n - y(t_n)\| &\leq \|\bar{y}_n - y(t_n)\| + \|\bar{\mathbf{y}}_n - \bar{y}_n\|, \\ \|\mathbf{y}_n - y(t_n)\| &\leq \|y_n - y(t_n)\| + \|\mathbf{y}_n - y_n\|. \end{aligned} \tag{3.32}$$

Mit dieser Vorarbeit können wir die Fehlerrekursion des MTS $-$ EXPPC$[2]_k^{(d,d)}$-Verfahrens durch

$$\begin{aligned} e_{n+1} &= y_{n+1} - y(t_{n+1}) + \mathbf{y}_{n+1} - y_{n+1} \\ &= e^{\tau M} e_n + \tau \sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau M)\big(\mathbf{g}^*_{n-k+1+j} - G_{n-k+1+j}\big) - \psi_{n+1} + \mathbf{y}_{n+1} - y_{n+1} \end{aligned}$$

beschreiben und die Auflösung dieser Fehlerrekursion liefert uns die folgende Darstellung des Fehlers

$$e_{n+1} = \tau \sum_{l=0}^{n-1} e^{(n-l-1)\tau M} \left(\sum_{j=0}^{k} \beta_j^{\mathrm{iM}}(\tau M)\big(\mathbf{g}^*_{l-k+1+j} - G_{l-k+1+j}\big) + \frac{1}{\tau}(\mathbf{y}_{l+1} - y_{l+1} - \psi_{l+1}) \right).$$

Verwenden wir neben (3.25) auch die Abschätzung

$$\|\bar{y}_j - y(t_j)\| \overset{(2.67)}{\leq} C \sum_{s=j-k}^{j-1} \|e_s\| + \|\delta_j\|, \quad j \geq 1, \quad C = C(C_P, \mathcal{L}_{\mathbf{g}}, \mathcal{B}), \tag{3.33}$$

so können wir den Fehler folgendermaßen abschätzen

$$\begin{aligned} \|e_n\| \leq\ & \tau \sum_{l=0}^{n-1} \sum_{j=0}^{k-1} \left\|e^{(n-l-1)\tau M}\right\| \left\|\beta_j^{\mathrm{iM}}(\tau M)\right\| \|\mathbf{g}_{l-k+1+j} - G_{l-k+1+j}\| \\ & + \tau \sum_{l=0}^{n-1} \left\|e^{(n-l-1)\tau M}\right\| \left(\left\|\beta_k^{\mathrm{iM}}(\tau M)\right\| \|\bar{\mathbf{g}}_{l+1} - G_{l+1}\| + \frac{\|\psi_{l+1}\|}{\tau} \right) \\ & + \sum_{l=0}^{n-1} \left\|e^{(n-l-1)\tau M}\right\| \|\mathbf{y}_{l+1} - y_{l+1}\| \end{aligned}$$

$$
\begin{aligned}
&\overset{(3.25)}{\leq} \tau C \sum_{l=0}^{n-1} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\| + \|\bar{\mathbf{y}}_{l+1} - y(t_{l+1})\| + \frac{\|\psi_{l+1}\|}{\tau} \right) \\
&\quad + C \sum_{l=0}^{n-1} \|\mathbf{y}_{l+1} - y_{l+1}\| \\
&\overset{\substack{(3.32)\\(3.33)}}{\leq} \tau C \sum_{l=0}^{n-1} \left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\| + \|\delta_{l+1}\| + \|\bar{\mathbf{y}}_{l+1} - \bar{y}_{l+1}\| + \frac{\|\psi_{l+1}\|}{\tau} \right) \\
&\quad + C \sum_{l=0}^{n-1} \|\mathbf{y}_{l+1} - y_{l+1}\| \\
&\leq \tau C \sum_{l=0}^{n-1} \left(\|e_l\| + \|\delta_{l+1}\| + \|\bar{\mathbf{y}}_{l+1} - \bar{y}_{l+1}\| + \frac{\|\psi_{l+1}\|}{\tau} \right) \\
&\quad + C \sum_{l=0}^{n-1} \|\mathbf{y}_{l+1} - y_{l+1}\| \\
&\leq C \max_{l=1,\dots,k-1} \|e_l\| + \tau C \sum_{l=0}^{n-1} \left(\|e_l\| + \|\delta_{l+1}\| + \frac{\|\psi_{l+1}\|}{\tau} \right) \\
&\quad + C \sum_{l=0}^{n-1} \left(\|\mathbf{y}_{l+1} - y_{l+1}\| + \tau \|\bar{\mathbf{y}}_{l+1} - \bar{y}_{l+1}\| \right).
\end{aligned}
$$

Für die Konstante in den bisherigen Abschätzungen gilt $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_{\mathbf{g}}, \mathbf{g})$. Mit den Abschätzungen (3.31) und (3.29) sowie der Voraussetzung 3.5 ergibt sich für $(\varepsilon_j)_{j\in\mathbb{N}}$ mit $\varepsilon_j := \|e_j\|$ die Ungleichung

$$
\varepsilon_n \leq C\tau \sum_{l=0}^{n-1} \varepsilon_l + C\tau^d. \tag{3.34}
$$

Die Anwendung des Gronwall-Lemmas 2.17 liefert uns die Behauptung mit einer Konstante $C = C(C_P, k, d, \mathcal{B}, \mathcal{L}_{\mathbf{g}}, \mathbf{g}, T, c_0, C_I)$.

Teil b): Unter diesen Voraussetzungen verändern sich wegen (2.65) sowohl die Abschätzungen (3.31) zu

$$
\begin{aligned}
\|\delta_{n+1}\| &\leq C\tau^{d+1} \sup_{0\leq t\leq t_{n+1}} \left\|G^{(d)}(t)\right\|, \\
\|\psi_{n+1}\| &\leq C\tau^{d+2} \sup_{0\leq t\leq t_{n+1}} \left\|G^{(d+1)}(t)\right\|.
\end{aligned} \tag{3.35}
$$

Nach *Teil a)* erfüllt der Fehler die Ungleichung

$$\|e_n\| \leq C \max_{l=1,\dots,k-1} \|e_l\| + \tau C \sum_{l=0}^{n-1} \left(\|e_l\| + \|\delta_{l+1}\| + \frac{\|\psi_{l+1}\|}{\tau} \right) + C \sum_{l=0}^{n-1} \left(\|\mathbf{y}_{l+1} - y_{l+1}\| + \tau \|\bar{\mathbf{y}}_{l+1} - \bar{y}_{l+1}\| \right).$$

Mithilfe der Abschätzungen (3.35) und (3.30) sowie der Voraussetzung 3.5 ergibt sich

$$\|e_n\| \leq C\tau \sum_{l=0}^{n-1} \|e_l\| + C\tau^{d+1}. \tag{3.36}$$

Mit Definition der Folge $(\varepsilon_j)_{j\in\mathbb{N}}$ durch $\varepsilon_j := \|e_j\|$ und der Anwendung des Gronwall-Lemmas 2.17 erhalten wir schließlich die Behauptung. Die Konstanten in Teil b) sind nicht identisch mit den aus Teil *a*), aber die Abhängigkeiten bleiben unverändert. □

Nachdem wir die multiple time stepping Varianten zu allgemeinen exponentiellen Mehrschrittverfahren vorgestellt und deren Konvergenz analysiert haben, möchten wir uns der Wahl von M widmen. Die Wahl von M hängt von der vorliegenden Steifigkeit des Problems ab. Für eine begriffliche Unterscheidung werden wir die Bezeichnungen **algebraische Steifigkeit** und **geometrisch induzierte Steifigkeit** verwenden. Mit der algebraischen Steifigkeit bezeichnen wir die Steifigkeit, die bei Problemen vorliegt, bei denen gleichzeitig „schnelle“ und „langsame“ Komponenten auftreten, wobei die verwendete Geschwindigkeitsangabe das Verhalten der Lösungskomponenten in Abhängigkeit der Zeit beschreibt. Die Steifigkeit wird in dem Fall durch die schnellen Komponenten verursacht, deren zeitliche Auflösung eine feine zeitliche Diskretisierung erfordert. Diese Form der Steifigkeit tritt häufig bei Reaktions-Diffusions-Gleichungen auf. Zur numerischen Integration solcher Probleme werden beispielsweise Multirate-Verfahren verwendet.

Die geometrisch induzierte Steifigkeit tritt hingegen bei Problemen auf, bei denen die räumliche Diskretisierung einer partiellen Differentialgleichung über einem Gebiet Ω eine lokale Verfeinerung aufweist. Dies sind zugleich die Probleme, die uns in dieser Arbeit hauptsächlich interessieren. Für die Bestimmung numerischer Lösungen eignen sich in diesem Fall die sogenannten local time stepping Verfahren. Sowohl die Multirate-Verfahren [19, 38, 48, 50] als auch die local time stepping Verfahren [11, 21, 22] können im Allgemeinen als multiple time stepping Verfahren [23] verstanden werden. In der Regel werden

diese Verfahren nur dann angewendet, wenn wenige schnelle Komponenten (algebraische Steifigkeit) bzw. wenige feine Elemente (geometrisch induzierte Steifigkeit) vorliegen und die Auswertung des nichtsteifen Anteils den größten Aufwand beansprucht. Ein multiple time stepping Verfahren ermöglicht eine Approximation auf verschiedenen Zeitskalen und zwar wird der nichtsteife Anteil auf einer groben und der steife Anteil auf einer feinen Zeitskala approximiert. In den folgenden Abschnitten werden wir speziell diese Verfahren beschreiben und deren Bezug zu den von uns vorgestellten Verfahren herstellen.

3.2.2 Local time stepping

Wie bereits erwähnt sind die local time stepping Verfahren für Probleme mit geometrisch induzierter Steifigkeit geeignet. Für M aus (3.16) sind folgende zwei Optionen naheliegend

$$A := \begin{bmatrix} S & Z \\ Y & B \end{bmatrix}, \quad M := \begin{bmatrix} S & 0 \\ Y & 0 \end{bmatrix}, \quad \Longrightarrow \quad A - M = \begin{bmatrix} 0 & Z \\ 0 & B \end{bmatrix}, \tag{3.37a}$$

$$A := \begin{bmatrix} S & Z \\ Y & B \end{bmatrix}, \quad M := \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}, \quad \Longrightarrow \quad A - M = \begin{bmatrix} 0 & Z \\ Y & B \end{bmatrix}. \tag{3.37b}$$

Betrachten wir ein diskretisiertes Problem der Form (3.16), dann wird die Matrix A durch M in einen steifen und einen nichtsteifen Anteil „gesplittet“. Die Blockmatrix S von A stellt hierbei den Anteil der Diskretisierung zu den wenigen „kleinen“ Elementen dar, wobei wir die „Größe“ eines Elements anhand der Intervalllänge oder des Inkreisradius messen. Die Vereinigung dieser kleinen Elemente entspricht zugleich dem feinen Teil der Diskretisierung bzw. dem Bereich der lokalen Verfeinerung. Dementsprechend stellt die Blockmatrix B den restlichen groben Teil der Diskretisierung mit größeren Elementen dar. Die Kopplung vom groben zum feinen Anteil wird durch die Blockmatrix Z und die vom feinen zum groben Anteil durch die Blockmatrix Y beschrieben. Aufgrund der Voraussetzung, dass die lokale Verfeinerung der Diskretisierung mit wenigen kleinen Elementen repräsentiert werden kann, ist die Größe der Matrix S deutlich kleiner als die von B.

Den Unterschied zwischen dem Splitting in (3.37a) und dem in (3.37b) können wir am Beispiel der dritten Graphik aus Abbildung 1.4 erklären. Hierfür sortieren wir diese Matrix um, so dass die feinen Knoten (blau) den ersten Diagonaleinträgen entsprechen. In Abbildung 3.1 haben wir die umsortierte Form der dritten Graphik aus Abbildung 1.4 mit den Splitting-Typen (3.37a) und (3.37b) visualisiert. Der Zweck ist, dass die unbe-

schränkten Koeffizienten nach dem Splitting in M enthalten sind, da die Auflösung dieser Komponenten eine feine zeitliche Diskretisierung benötigt. Wir können erkennen, dass dies mit den Splittings (3.37a) und (3.37b) realisiert wird.

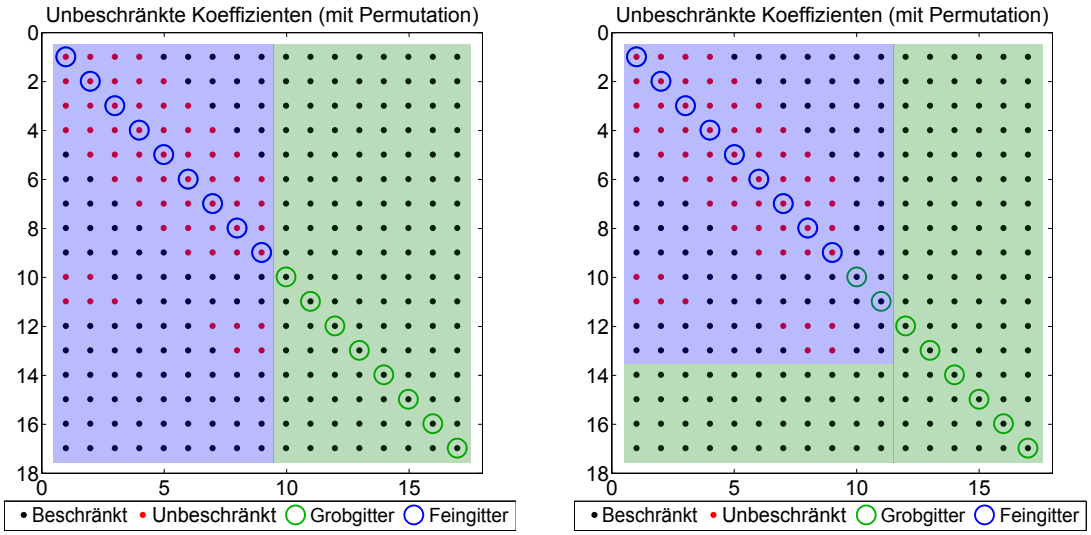

Abbildung 3.1: Visualisierung der zwei Splitting-Typen (3.37a) und (3.37b) für local time stepping Verfahren (Links: (3.37a), Rechts:(3.37b)) zusammen mit den nichtnull-Blöcken von M (blauer Bereich) und $A - M$ (grüner Bereich).

In [33] haben Hochbruck und Ostermann gezeigt, dass die exponentiellen Mehrschrittverfahren allgemein für multiple time stepping geeignet sind und haben speziell für die „Splittings“ (3.37a) und (3.37b) die Verfahrensvorschriften präsentiert. Bei der Konstruktion der local time stepping Verfahren haben Diaz, Grote und Mitkova in ihren Arbeiten [11, 21, 22] ebenfalls diese Splittings verwendet. Wir zeigen, dass die von Grote und Mitkova [22] vorgestellten local time stepping Varianten zu klassischen Adams-Verfahren $\mathrm{LTS} - \mathrm{AB}k(r)$ äquivalent sind zu gewissen multiple time stepping Verfahren exponentieller Mehrschrittverfahren $\mathrm{MTS} - \mathrm{EXPMS}_k^d$, die wir in Abschnitt 3.2 eingeführt haben.

Satz 3.8: *(Beziehung zwischen* $\mathrm{LTS} - \mathrm{AB}k(r)$ *[22] und* $\mathrm{MTS} - \mathrm{EXPMS}_k^k$*)*

Das $\mathrm{LTS} - \mathrm{AB}k(r)$*-Verfahren ist für* $k \geq 1$ *äquivalent zum* $\mathrm{MTS} - \mathrm{EXPMS}_k^k$*-Verfahren, soweit der innere Integrator von* $\mathrm{MTS} - \mathrm{EXPMS}_k^k$ *einem klassischen* k*-Schritt Adams-Verfahren entspricht.*

Beweis: *(Satz 3.8)*
Wir beschränken uns in diesem Beweis auf ein Anfangswertproblem der Form (3.16) mit $g = 0$, da die Verfahren in [22] ebenfalls für lineare Anfangswertprobleme beschrieben wurden. Die Äquivalenz ist jedoch analog für $g \neq 0$ gültig. Notieren wir $B_{\text{fine}} := M$ und $B_{\text{coarse}} := A - M$, so ist das Anfangswertproblem durch

$$y'(t) = My(t) + \mathbf{g}\big(t, y(t)\big) = B_{\text{fine}}y(t) + B_{\text{coarse}}y(t), \quad y(t_0) = y_0 \tag{3.38}$$

gegeben. Wir leiten zunächst eine geeignete Darstellung der Näherung des LTS $-$ AB$k(r)$-Verfahrens her. Für die Lösung von (3.38) gilt

$$\begin{aligned} y(t_n + \tau) &= y(t_n) + \int_{t_n}^{t_n+\tau} B_{\text{fine}}y(t) + B_{\text{coarse}}y(t)\mathrm{d}t \\ &= y(t_n) + \tau \int_0^1 B_{\text{fine}}y(t_n + \theta\tau)\mathrm{d}\theta + \tau \int_0^1 B_{\text{coarse}}y(t_n + \theta\tau)\mathrm{d}\theta. \end{aligned} \tag{3.39}$$

Die Approximation von

$$\begin{aligned} \mathbf{g}\big(t_n + \theta\tau, y(t_n + \theta\tau)\big) &= B_{\text{coarse}}y(t_n + \theta\tau) \\ &\approx p_n(t_n + \theta\tau) = B_{\text{coarse}} \sum_{j=0}^{k-1} (-1)^j \binom{-\theta}{j} \nabla^j y_n, \end{aligned}$$

in (3.39) liefert mit $y_n \approx y(t_n)$ die folgende Näherung zu (3.39)

$$\begin{aligned} y(t_n + \tau) \approx \mathbf{y}(\tau) &= y_n + \tau \int_0^1 B_{\text{fine}}\mathbf{y}(\theta\tau)\mathrm{d}\theta + \tau \int_0^1 p_n(t_n + \theta\tau)\mathrm{d}\theta \\ &= y_n + \int_0^\tau \left(B_{\text{fine}}\mathbf{y}(t) + p_n(t_n + t)\right)\mathrm{d}t. \end{aligned}$$

Diese Approximation entspricht gleichzeitig der exakten Lösung des Anfangswertproblems

$$\mathbf{y}(t)' = B_{\text{fine}}\mathbf{y}(t) + p_n(t_n + t) = M\mathbf{y}(t) + p_n(t_n + t), \quad \mathbf{y}(0) = y_n, \quad t \in [0, \tau]. \tag{3.40}$$

Mit der Anwendung des klassischen k-Schritt Adams-Verfahrens auf (3.40) mit der Schrittweite $\tau_{\text{inner}} := \frac{\tau}{r}$ erhalten wir die Approximation des LTS $-$ AB$k(r)$-Verfahrens. Damit ist bereits die Aussage bewiesen, denn nach der Konstruktion (s. Abschnitt 3.2) der MTS $-$ EXPMS$_k^k$-Verfahren wird $\mathbf{g}$ ebenfalls durch p_n approximiert und die Anwendung der Variation der Konstanten Formel liefert das gleiche Anfangswertproblem (3.40). Nach

Voraussetzung wird auf (3.40) das klassische k-Schritt Adams-Verfahren als innerer Integrator angewendet. Dementsprechend sind diese Verfahren zueinander äquivalent. □

Wie bereits erwähnt haben Grote und Diaz [11] sowie Grote und Mitkova [21, 22] local time stepping Verfahren konstruiert und diese auf lineare hyperbolische Probleme angewendet. Die Stabilität dieser Verfahren haben Sie anhand numerischer Experimente verifiziert. Für simultan diagonalisierbare Matrizen lässt sich dieses Ergebnis mithilfe der Stabilitätsanalyse von Beylkin [5] erklären. Betrachten wir hierfür das Anfangswertproblem (3.16) mit dem Splitting (3.37a) und $g = 0$, so gilt

$$\begin{aligned} & y' = My + (A - M)y, \quad y(t_0) = y_0 \\ \iff \quad & \begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} S & 0 \\ Y & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 0 & Z \\ 0 & B \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \quad \begin{bmatrix} u(t_0) \\ v(t_0) \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}. \end{aligned}$$

Falls M und $A - M$ kommutieren, so sind diese Matrizen simultan diagonalisierbar und das Anfangswertproblem ist äquivalent zu einem entkoppelten System der Form

$$\begin{bmatrix} \widetilde{u}' \\ \widetilde{v}' \end{bmatrix} = \underbrace{\begin{bmatrix} D_S & 0 \\ 0 & 0 \end{bmatrix}}_{=:\widetilde{M}} \begin{bmatrix} \widetilde{u} \\ \widetilde{v} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & D_B \end{bmatrix}}_{=:\widetilde{A}-\widetilde{M}} \begin{bmatrix} \widetilde{u} \\ \widetilde{v} \end{bmatrix}.$$

Mit Hilfe der skalaren Testgleichung (2.54) können wir erkennen, dass die Stabilität in dem Fall unabhängig von der lokalen Verfeinerung ist. Denn durch die Anwendung eines exponentiellen Mehrschrittverfahrens auf dieses entkoppelte System wird der steife Anteil $\widetilde{M}$ exakt bestimmt und der nichtsteife Anteil $\widetilde{A} - \widetilde{M}$ mit dem zugrunde liegenden klassischen Mehrschrittverfahren approximiert. Wir möchten betonen, dass die Stabilitätsanalyse von Beylkin für allgemeinere Fälle nicht gültig ist, da im Allgemeinen M und $A - M$ nicht kommutieren und die skalare Testgleichung zur Stabilitätsanalyse im Allgemeinen ungeeignet ist.

Bei den multiple time stepping Verfahren aus Abschnitt (3.2.1) ist neben der Stabilität des äußeren Integrators auch die Stabilität des inneren Integrators entscheidend. Der innere Integrator ist frei wählbar. Die Schrittweite $\tau_{\text{inner}} = \frac{\tau}{r}$ muss so gewählt werden, dass der innere Integrator für diese stabil und hinreichend genau ist. Dies kann für hinreichend großes r immer erreicht werden. Wir beschreiben mit $\tau_{\text{outer}}^{\max}$ und $\tau_{\text{inner}}^{\max}$ die maximal stabile Schrittweite des äußeren und inneren Integrators zu einem gegebenen Problem. Zudem

definieren wir mit r_s einen Verfeinerungsfaktor, der ein Maß für das Größenverhältnis (Länge, Fläche, Volumen) zwischen den feinen und groben Elementen darstellt. Des Weiteren sei $p(r_s)$ ein Schätzfaktor, der uns angibt mit welchem Faktor die Eigenwerte des gegebenen Problems abhängig von dem Verfeinerungsfaktor r_s wachsen. Demnach können wir mit der Wahl

$$r = \left\lceil p \cdot \frac{\tau_{\text{outer}}^{\max}}{\tau_{\text{inner}}^{\max}} \right\rceil \tag{3.41}$$

die Stabilität des inneren Integrators erwarten. Hierbei passt der Schätzfaktor p die Schrittweite τ auf die feine Zeitskala an und der Quotient aus $\tau_{\text{outer}}^{\max}$ und $\tau_{\text{inner}}^{\max}$ passt die Schrittweite mit dem Stabilitätsverhältnis zwischen dem inneren und äußeren Integrator an. In der Praxis kann die maximale Schrittweite eines Verfahrens anhand der effektiven Optimierungsparameter (s. Abschnitt 2.2.6) abgeschätzt werden. Für die Abschätzung des Parameters p ist die Beziehung zwischen dem Verfeinerungsfaktor und den Eigenwerten entscheidend. Diese besitzen bei hyperbolischen Problemen einen linearen und bei parabolischen Problemen einen quadratischen Zusammenhang. Mit dem nächsten Beispiel verifizieren wir diese Eigenschaften und fahren mit den Multirate-Verfahren fort.

Beispiel 3.9: *(Fortsetzung von Beispiel 2.23)*
Betrachten wir die 1D Wärmeleitungsgleichung aus Beispiel 2.23, wobei die Diskretisierung von Ω folgende lokale Verfeinerung aufweist:

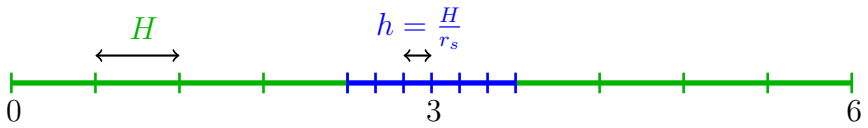

Abbildung 3.2: Skizze einer Diskretisierung von $\Omega = [0, 6]$ mit Gitterweite H. Die lokale Verfeinerung im Bereich $[3 - H, 3 + H]$ wird mit dem Verfeinerungsfaktor $r_s = 3$ erzeugt.

Das lineare Anfangswertproblem (2.76), das aus der räumlichen Diskretisierung der Wärmeleitungsgleichung (2.75) mit zentralen finiten Differenzen resultiert können wir dann mithilfe von (3.37a) bzw. (3.37b) umformulieren zu

$$y' = My + (A - M)y, \quad y(t_0) = y_0$$

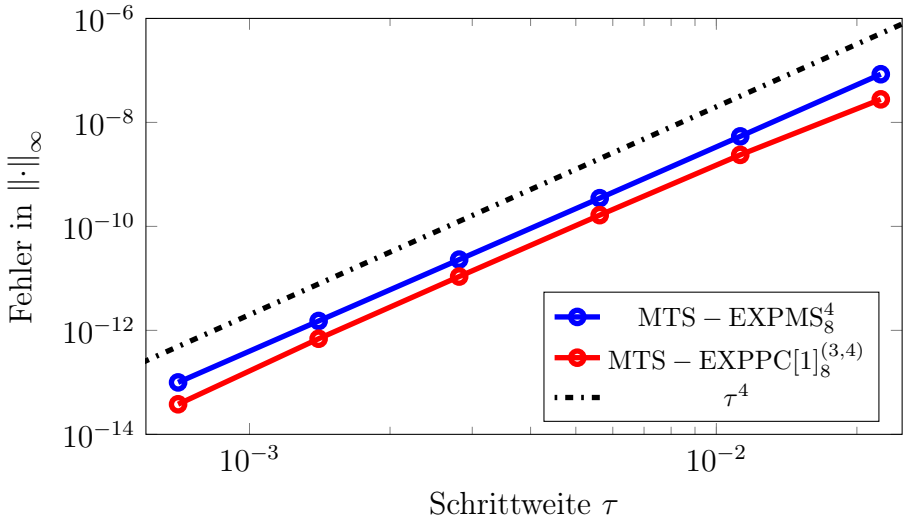

Abbildung 3.3: Fehler (des Endzeitpunkts in der $\|\cdot\|_\infty$-Norm) der Zeitintegratoren MTS – EXPMS$_8^4$ und MTS – EXPPC[1]$_8^{(3,4)}$ in Abhängigkeit der Zeitschrittweite.

Die Konvergenzordnung der multiple time stepping Varianten exponentieller Mehrschrittverfahren prüfen wir anhand der Verfahren MTS – EXPMS$_8^4$ und MTS – EXPPC[1]$_8^{(3,4)}$. Die Fehler der numerischen Lösungen dieser Verfahren auf dem Zeitintervall $T = [0, 10]$ sind in Abbildung 3.3 in Abhängigkeit der Schrittweite τ logarithmisch dargestellt. Als innerer Integrator wurde stets das klassische Runge-Kutta-Verfahren verwendet und der Parameter r wurde entsprechend (3.41) gewählt mit $p = r_s^2$ mit $r_s = 3$. Wir können erkennen, dass auch die multiple time stepping Varianten exponentieller Mehrschrittverfahren die gewünschte Ordnung erhalten. Als nächstes vergleichen wir die Stabilität verschiedener Mehrschrittverfahren der Konsistenzordnung $d = 4$, indem wir die maximal stabilen Schrittweiten dieser Verfahren betrachten. Diese sind in Abbildung 3.4 aufgeführt, wobei die klassischen Verfahren in rot, die exponentiellen in blau und die optimierten in grün dargestellt sind. Wie erwartet erlauben die optimierten Verfahren die Wahl einer größeren Schrittweiten. Des Weiteren ist zu erkennen, dass die exponentiellen Mehrschrittverfahren unabhängig von der Verfeinerung dieselbe maximale Schrittweite zulassen. Bei den klassischen Verfahren können wir eine quadratische Reduktion der maximalen Schrittweite in Abhängigkeit der Verfeinerung erkennen.

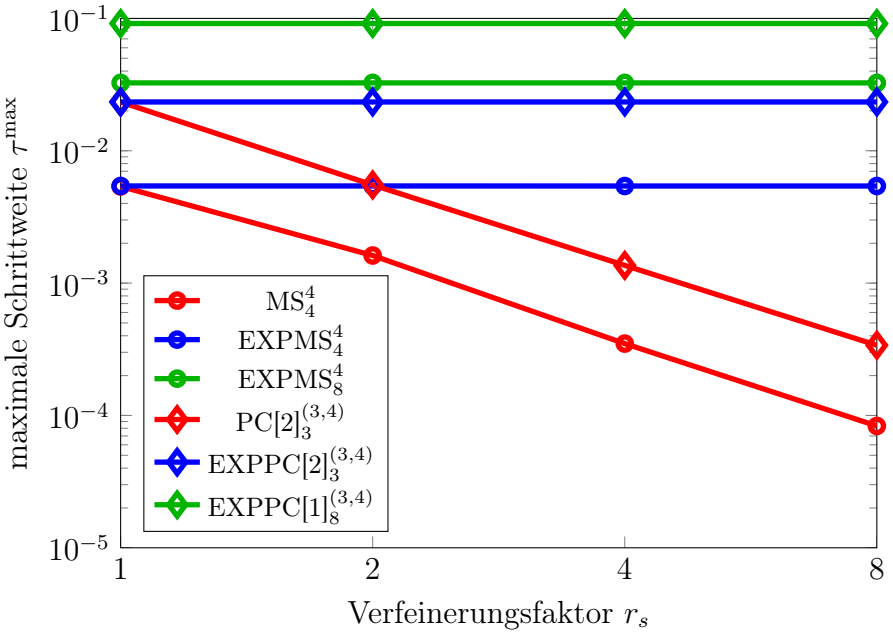

Abbildung 3.4: Maximal stabile Schrittweiten $\tau^{\max}$ von Mehrschrittverfahren der Konsistenzordnung $d = 4$.

In einem weiteren Test haben wir die exponentiellen Mehrschrittverfahren mit deren multiple time stepping Varianten verglichen, wobei wir stets das klassische Runge-Kutta-Verfahren als inneren Integrator gewählt und den Parameter r entsprechend (3.41) bestimmt haben. Das Resultat ist in Abbildung 3.5 dargestellt, wobei wir bei den multiple time stepping Varianten $\tau_{\text{outer}}^{\max}$ als maximale Schrittweite zu verstehen haben, da die äußeren Integratoren den exponentiellen Mehrschrittverfahren entsprechen. Wir sehen, dass die äußeren Integratoren für r aus (3.41) dasselbe Verhalten aufweisen, wie die zugehörigen exponentiellen Integratoren. Nach Konstruktion der multiple time stepping Verfahren ist es offensichtlich, dass r und damit auch der Aufwand des inneren Integrators quadratisch in Abhängigkeit des Verfeinerungsfaktors r_s wächst. Dieser Aufwand wird im Grunde genommen nur im Bereich der lokalen Verfeinerung betrieben, wohingegen die Schrittweitenreduktion bei klassischen Verfahren das ganze System betrifft.

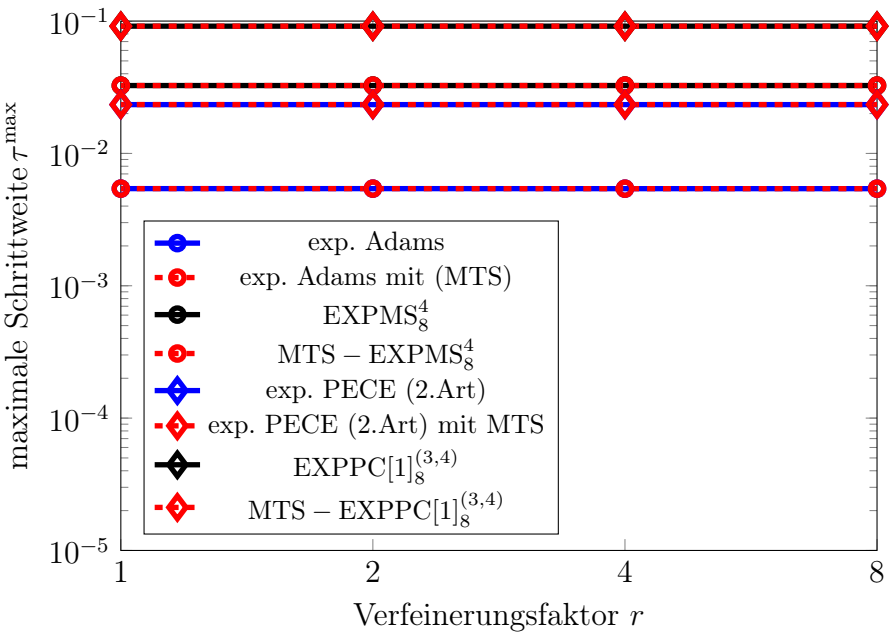

Abbildung 3.5: Maximal stabile Schrittweiten der multiple time stepping Varianten exponentieller Mehrschrittverfahren der Konsistenzordnung $d = 4$.

3.2.3 Multirate-Verfahren

Multirate-Verfahren wurden erstmals von Rice [48] für Runge-Kutta-Verfahren formuliert. Später haben Gear und Wells [19] diese Idee für lineare Mehrschrittverfahren aufgegriffen. Multirate-Verfahren sind für Probleme geeignet bei denen die Lösungsgkomponenten sowohl schnelle als auch langsame Komponenten enthalten. Semilineare Probleme (2.1) dieser Art können mit dem folgenden Splitting

$$A := \begin{bmatrix} S & Z \\ Y & B \end{bmatrix}, \quad M := \begin{bmatrix} S & Z \\ 0 & 0 \end{bmatrix}, \quad \Longrightarrow \quad A - M = \begin{bmatrix} 0 & 0 \\ Y & B \end{bmatrix} \tag{3.42}$$

umformuliert werden zu einem Anfangswertproblem der Form (3.16). Im Vergleich zu den Splittings der local time stepping Verfahren (3.37a) und (3.37b), wo der lineare Anteil spaltenweise unterteilt wurde, wird der lineare Anteil A zeilenweise aufgeteilt. Dies ist der wesentliche Unterschied zwischen Multirate- und local time stepping Verfahren.

Die Anwendung eines Multirate-Verfahrens auf (3.16) ermöglicht die unterschiedliche numerische Behandlung der schnellen und langsamen Komponenten. Die wenigen schnellen

Komponenten werden dadurch in einer feinen Zeitskala und die vielen langsamen Komponenten in einer groben Zeitskala aufgelöst. Wie bei den local time stepping Verfahren wird nur der Anteil mit einer kleinen Schrittweite behandelt, der eine feine zeitliche Diskretisierung erfordern. Mit dem nächsten Satz zeigen wir, dass die multiple time stepping Varianten exponentieller Mehrschrittverfahren auch als Multirate-Verfahren aufgefasst werden können.

Satz 3.10: *(Bezug zu Multirate-Verfahren)*

Sei (3.16) ein Anfangswertproblem mit M aus (3.42). Die Anwendung einer multiple time stepping Variante eines exponentiellen Mehrschrittverfahrens aus Abschnitt 3.2 entspricht einem Multirate-Verfahren, sofern die Konsistenzordnungen des inneren und äußeren Integrators übereinstimmen.

Beweis: *(Satz 3.10)*
Wir werden den Beweis konstruktiv führen und betrachten das semilineare Problem (3.16) mit dem Splitting (3.42)

$$\begin{aligned} & y' = My + \mathbf{g}(t,y), \quad y(t_0) = y_0 \\ \Longleftrightarrow \qquad & \begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} S & Z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \mathbf{g}^{[u]}(t,u,v) \\ \mathbf{g}^{[v]}(t,u,v) \end{bmatrix}, \quad \begin{bmatrix} u(t_0) \\ v(t_0) \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}. \end{aligned}$$

Die polynomielle Approximation p_n von $\mathbf{g}$ liefert uns das Anfangswertproblem

$$\begin{aligned} & \mathbf{y}' = M\mathbf{y} + p_n(t), \quad \mathbf{y}(t_n) = y_n, \\ \Longleftrightarrow \qquad & \begin{bmatrix} \mathbf{u}' \\ \mathbf{v}' \end{bmatrix} = \begin{bmatrix} S & Z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} p_n^{[\mathbf{u}]}(t) \\ p_n^{[\mathbf{v}]}(t) \end{bmatrix}, \quad \begin{bmatrix} \mathbf{u}(t_0) \\ \mathbf{v}(t_0) \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}. \end{aligned} \tag{3.43}$$

mit $\mathbf{y}(\tau) \approx y(t_{n+1})$. Die langsamen Komponenten $\mathbf{v} \approx v$ sind entkoppelt und die polynomielle Approximation p_n entspricht einer Approximation auf der groben Zeitskala der Schrittweite τ. Der bisherige Teil entspricht der äußeren Integration einer multiple time stepping Variante eines exponentiellen Mehrschrittverfahrens.

Die Komponenten $\mathbf{v} \approx v$ können exakt bestimmt werden, da das Polynom $p_n^{[\mathbf{v}]}$ exakt integriert werden kann. Insgesamt ergibt sich die Approximation eines Multirate-Verfahrens [38, 50], indem die Approximation zu den Komponenten $\mathbf{u} \approx u$ (3.43) mit einem Mehrschrittverfahren auf der feinen Zeitskala bestimmt werden. Verwenden wir also ein Mehrschrittverfahren als inneren Integrator, deren Konsistenzordnung nach Voraussetzung mit

der Konsistenzordnung des äußeren Integrators übereinstimmt, so entspricht diese multiple time stepping Variante eines exponentiellen Mehrschrittverfahrens einem Multirate-Verfahren. Notieren wir die Konsistenzordnung des inneren und äußeren Integrators mit d, so sind die Polynome $p_n^{[\mathbf{u}]}$ und $p_n^{[\mathbf{v}]}$ vom Grad $\leq d-1$. Die langsamen Lösungskomponenten $\mathbf{v}$ werden demnach exakt bestimmt und die schnellen Komponenten werden approximativ auf der entsprechenden feinen Zeitskala der Schrittweite $\tau_{\text{inner}} = \frac{\tau}{r}$ bestimmt. □

3.3 Startwertprozedur

Bei Mehrschrittverfahren mit $k \geq 1$ ist die Bestimmung von Startwerten notwendig. Wenn wir zugleich die Konvergenzresultate beachten, so benötigen wir Startwerte mit einer gewünschten Genauigkeit. Mithilfe von Einschrittverfahren können diese Startwerte hinreichend genau ermittelt werden. Eine nützliche Alternative zur Bestimmung der Startwerte wurde von Hochbruck und Ostermann in [33] vorgestellt, die auf der Arbeit von Calvo und Palencia [8] beruht. Wir werden diese Prozedur in Anlehnung an [33] erklären und eine multiple time stepping Variante dieser Startwert-Prozedur beschreiben.

Zur Bestimmung der Startwerte y_j mit $j = 1, \ldots, k-1$ für ein k-Schrittverfahren verwenden wir das Interpolationspolynom

$$p_0(t_0 + \theta\tau) = \sum_{j=0}^{k-1} \binom{\theta}{j} \Delta^j g_0 \tag{3.44}$$

zum Datensatz (t_j, g_j) mit $j = 0, \ldots, k-1$ und $g_j := g(t_j, y_j)$, wobei die Vorwärtsdifferenzen wie folgt definiert sind

$$\Delta^j g_n = \begin{cases} g_n, & j = 0 \\ \Delta^{j-1} g_{n+1} - \Delta^{j-1} g_n, & j \neq 0 \end{cases}.$$

Nach der VdK-Formel gilt für $1 \leq m \leq k-1$ die Darstellung der exakten Lösung

$$y(t_0 + m\tau) = e^{m\tau A} y(t_0) + \tau \int_0^m e^{(1-\theta)\tau A} g\big(t_0 + \theta\tau, y(t_0 + \theta\tau)\big) \mathrm{d}\theta. \tag{3.45}$$

Ersetzen wir g in (3.45) durch das Interpolationspolynom p_0, so ergeben sich die Approximationen

$$y(t_0 + m\tau) \approx y_m = e^{m\tau A} y_0 + \tau \int_0^m e^{(1-\theta)\tau A} p_0(t_0 + \theta\tau) \mathrm{d}\theta \tag{3.46}$$

für $m = 1, \ldots, k-1$. Damit erhalten wir insgesamt das nichtlineare Gleichungssystem

$$\mathbf{Y} = \mathcal{N}(\mathbf{Y}) \quad \text{mit} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{k-1} \end{bmatrix} \quad \text{und} \quad \mathcal{N}(\mathbf{Y}) := \begin{bmatrix} \mathbf{N}_1(\mathbf{Y}) \\ \vdots \\ \mathbf{N}_{k-1}(\mathbf{Y}) \end{bmatrix}, \tag{3.47}$$

wobei

$$\mathbf{N}_m(\mathbf{Y}) := e^{m\tau A} y_0 + \tau \int_0^m e^{(1-\theta)\tau A} p_0(t_0 + \theta\tau) \mathrm{d}\theta$$

gilt. In unseren Konvergenzbeweisen setzen wir stets voraus, dass g eine Lipschitz-Bedingung erfüllt. Mit dieser Voraussetzung entspricht die Abbildung $\mathcal{N}$ für hinreichend kleine Schrittweiten einer kontrahierenden Selbstabbildung. Definieren wir zu (3.47) die Fixpunktiteration

$$\mathbf{Y}^{[n+1]} = \mathcal{N}\big(\mathbf{Y}^{[n]}\big) \quad \text{mit} \quad \mathcal{N}\big(\mathbf{Y}^{[n]}\big) := \begin{bmatrix} \mathbf{N}_1\big(\mathbf{Y}^{[n]}\big) \\ \vdots \\ \mathbf{N}_{k-1}\big(\mathbf{Y}^{[n]}\big) \end{bmatrix}, \tag{3.48}$$

so konvergiert diese Fixpunktiteration nach dem Fixpunktsatz von Banach für hinreichend kleine Schrittweiten τ. Zur Initialisierung von $\mathbf{Y}^{[0]}$ kann das exponentielle Euler-Verfahren verwendet werden. Der Vorteil einer solchen Startwert-Prozedur ist, dass die gewünschte Toleranz bei der Berechnung von Startwerten kontrolliert werden kann. Zudem basiert diese Startwert-Prozedur auf derselben Idee wie die exponentiellen Mehrschrittverfahren, so dass die Implementierung keine große Schwierigkeit darstellt.

Für die multiple time stepping Variante dieser Startwert-Prozedur folgen wir der Idee aus Abschnitt 3.2. Mit der Anwendung der Variation der Konstanten Formel auf (3.46) können wir das Anfangswertproblem zur Fixpunktiteration beschreiben durch

$$\mathbf{y}^{[n+1]'} = A\mathbf{y}^{[n+1]} + p_0(t_0 + t), \quad \mathbf{y}^{[n+1]}(0) = y_0, \quad t \in [0, (k-1)\tau], \tag{3.49}$$

mit

$$\mathbf{y}^{[n+1]}(m\tau) = y_m^{[n+1]} \approx y(t_m) \quad \text{für} \quad 1 \leq m \leq k-1.$$

Anstatt der exakten Bestimmung von $y_1^{[n+1]}, \ldots, y_{k-1}^{[n+1]}$ nach (3.46) können wir einen inneren Integrator der gewünschten Konsistenzordnung mit der Schrittweite $\tau_{\text{inner}} = \frac{\tau}{r}$ verwenden. Diese liefert uns die Approximationen $\hat{y}_1^{[n+1]}, \ldots, \hat{y}_{k-1}^{[n+1]}$ mit

$$\hat{y}_m^{[n+1]} \approx \mathbf{y}^{[n+1]}(m\tau) = y_m^{[n+1]} \approx y(t_m)$$

mit denen wir die Fixpunktiteration ausführen können. Diese Startwert-Prozedur stellt eine Alternative dar, die keine Berechnung von Matrixfunktionen erfordert.

Sollte der innere Integrator bei einem multiple time stepping Verfahren einem l-Schrittverfahren entsprechen, so sind in der Regel auch dafür Startwerte (auf der feinen Zeitskala) zu bestimmen. In diesem Fall sollte der innere Integrator der Startwert-Prozedur zusätzlich die Approximationen zu

$$\mathbf{y}^{[n+1]}(t_{k-1} - m\tau_{\text{inner}}) \quad \text{für} \quad 1 \leq m \leq l-1 \tag{3.50}$$

bestimmen. Diese Approximationen (3.50) können dann als Startwerte für den inneren Integrator verwendet werden, das einem l-Schrittverfahren entspricht. Damit schließen wir die Theorie dieser Verfahren für semilineare Probleme ab und widmen uns im nächsten Kapitel der Implementierung der vorgestellten Mehrschrittverfahren.

4 Implementierung

In diesem Kapitel werden wir die Algorithmen allgemeiner exponentieller Mehrschrittverfahren und deren multiple time stepping Varianten vorstellen. Ebenso werden wir Pseudocodes zu den Startwertprozeduren aufführen. Anschließend werden wir die Anwendungsbeispiele präsentieren, die in Zusammenarbeit mit Hochbruck, Niegemann und Busch [10] entstanden sind.

In Abschnitt 2.2.6 haben wir darauf hingewiesen, dass wir die Optimierungsroutine für die Bestimmung optimierter klassischer Mehrschrittverfahren verwenden können. Die Implementierung optimierter Verfahren ist im klassischen Fall analog zu den bereits bekannten Adams-Verfahren. Aus dem Grund werden wir die Implementierung optimierter klassischer Mehrschrittverfahren nicht beschreiben.

4.1 Codes

Viele Details zur rechnerischen Umsetzung exponentieller Adams-Verfahren wurden in [33] beschrieben. In diesem Kapitel werden wir auf diese Details verweisen und ergänzende Informationen aufführen, die die rechnerische Umsetzung der vorgestellten Verfahren vereinfachen sollen.

4.1.1 Implementierung exponentieller Mehrschrittverfahren

Exponentielle Mehrschrittverfahren stellen für semilineare Probleme (2.1) bzw. (3.16) eine geeignete Klasse von Zeitintegratoren dar. Diese Integratoren können zugleich als multiple time stepping (local time stepping bzw. Multirate-) Verfahren aufgefasst werden, die die „innere Integration“ exakt bestimmen. Wenn die Steifigkeit des linearen Operators zu stark ausgeprägt ist, sodass sich die multiple time stepping Varianten exponentieller Mehrschrittverfahren nicht rentieren, so könnten die optimierten exponentiellen Mehrschrittverfahren eine gute Alternative darstellen. Denn die Steifigkeit im linearen Anteil stellt für exponentielle Integratoren keine Schwierigkeit dar, da kein innerer Integrator benötigt wird. Allerdings erfordern diese Verfahren die Berechnung von Matrixfunktion-Vektor-Produkten (s. Abschnitt 3.1).

In [33] haben Hochbruck und Ostermann die exponentiellen Adams-Verfahren anhand der Darstellung (2.24) analysiert. Diese Darstellung ist insbesondere für die Implementierung dieser Verfahren von Vorteil, denn die Rückwärtsdifferenzern erfüllen für hin-

reichend glatte Lösungen die Beziehung $\|\nabla^j g_n\| = \mathcal{O}(\tau^j)$. Dies hat zur Folge, dass die Krylov-Verfahren für $j > 0$ deutlich schneller konvergieren [33, Fig. 5.1]. Aus dem Grund werden wir am Beispiel eines expliziten allgemeinen exponentiellen Mehrschrittverfahrens erläutern, wie wir aus der Darstellung (2.43) eine Darstellung mit Rückwärtsdifferenzen erhalten. Im impliziten Fall gilt dies analog mit den entsprechenden Rückwärtsdifferenzen.

Ein explizites allgemeines Mehrschrittverfahren der Form (2.43) können wir anhand der Rückwärtsdifferenzen wie folgt repräsentieren

$$y_{n+1} = e^{\tau A} y_n + \tau \sum_{j=0}^{k-1} \beta_j(\tau A) g_{n-k+1+j} = e^{\tau A} y_n + \tau \sum_{j=0}^{k-1} \gamma_j(\tau A) \nabla^j g_n. \tag{4.1}$$

Nach Berechnung ergibt sich für die γ_j-Funktionen die folgende Rekursion

$$\begin{aligned} \gamma_{k-1-l}(x) &= (-1)^{k-1-l} \beta_l(x) - \sum_{i=k-l}^{k-1} \binom{i}{k-1-l} \gamma_i(x) \\ \text{mit} \quad \beta_l(x) &= \sum_{i=0}^{d-1} b_{i,l} \varphi_{i+1}(x) \quad \text{und} \quad l = 0, \ldots, k-1. \end{aligned} \tag{4.2}$$

Die skalaren Funktionen γ_j für $j = 0, \ldots, k-1$ sind unabhängig von der Problemstellung nur einmalig zu bestimmen. Damit wir ein solches Mehrschrittverfahren überhaupt anwenden können benötigen wir hinreichend genaue Startwerte. Aus dem Grund haben wir in Algorithmus 1 eine Prozedur für die mögliche Berechnung der Startwerte entsprechend dem Abschnitt 3.3 beschrieben.

Algorithmus 1 Startwertprozedur für exponentielle Mehrschrittverfahren (kurz: `StartValExp`)

Input: t_0, y_0, τ, `tol`
Definiere $p_0(t) = y_0$ und setze $l = 0$
Berechne mit (3.46) y_m für $m = 1, \ldots, k-1$ und definiere $\mathbf{Y}^{[1]} = \mathbf{Y}$
while `fehler` $>$ `tol` **do**
  Setze $l = l + 1$
  Bestimme das lokale Interpolationspolynom p_0 zu den Punkten $\big(t_j, g(t_j, y_j)\big)_{j=0}^{k-1}$
  Berechne $\mathbf{Y}^{[l+1]} = \mathcal{N}(\mathbf{Y}^{[l]})$ nach (3.48)
  Berechne `fehler` anhand von $\mathbf{Y}^{[l+1]}$ und $\mathbf{Y}^{[l]}$
end while
Output: Startwerte $y_1, \ldots, y_{k-1}$.

Für die Implementierung kann als `fehler` beispielsweise die diskrete L^2-Norm aus der Differenz der Fixpunktiterierten verwendet werden. Mit Algorithmus 1 lassen sich die Algorithmen zu den EXPMS$_k^d$- und EXPPC[$\mathbf{n}$]$_k^{(d_\mathrm{P},d_\mathrm{C})}$-Verfahren wie folgt formulieren:

Algorithmus 2 EXPMS$_k^d$-Verfahren

Input: y_0, tspan $= [t_0, T]$, τ
Initialisiere: $n = 0, \quad t = t_0, \quad y_n = y_0$
if $k \geq 2$ **then**
  Bestimme k Startwerte $y_0, \ldots, y_{k-1}$ mit `StartValExp`
  Berechne $g_0 = g(t_0, y_0), \ldots, g_{k-1} = g(t_{k-1}, y_{k-1})$
  Setze $n = n + k - 1, \quad t = t + (k-1)\tau, \quad y_n = y_{k-1}$
end if
while $(t < T)$ **do**
  Bestimme y_{n+1} aus (2.38) mit Hilfe von Krylov-Verfahren
  Berechne $g_{n+1} = g(t_{n+1}, y_{n+1})$ und setze $n = n + 1, \quad t = t + \tau$
end while
Output: $y_i \approx y(t_i)$ mit $t_i = t_0 + i\tau$ und $i = 1, \ldots, n$

Algorithmus 3 EXPPC[$\mathbf{n}$]$_k^{(d_\mathrm{P},d_\mathrm{C})}$-Verfahren

Input: y_0, tspan $= [t_0, T]$, τ
Initialisiere: $n = 0, \quad t = t_0, \quad y_n = y_0$
if $k \geq 2$ **then**
  Bestimme k Startwerte $y_0, \ldots, y_{k-1}$ mit `StartValExp`
  Berechne $g_0 = g(t_0, y_0), \ldots, g_{k-1} = g(t_{k-1}, y_{k-1})$
  Setze $n = n + k - 1, \quad t = t + (k-1)\tau, \quad y_n = y_{k-1}$
end if
while $(t < T)$ **do**
  Bestimme $\bar{y}_{n+1}$ aus (2.40) mit Hilfe von Krylov-Verfahren
  Berechne $\bar{g}_{n+1} = g(t_{n+1}, \bar{y}_{n+1})$
  Bestimme y_{n+1} aus (2.41) mit Hilfe von Krylov-Verfahren
  Berechne $g_{n+1} = g(t_{n+1}, y_{n+1})$ und setze $n = n + 1, \quad t = t + \tau$
end while
Output: $y_i \approx y(t_i)$ mit $t_i = t_0 + i\tau$ und $i = 1, \ldots, n$

4.1.2 Implementierung der multiple time stepping Varianten exponentieller Mehrschrittverfahren

Bevor wir die Algorithmen der multiple time stepping Varianten exponentieller Mehrschrittverfahren vorstellen, möchten wir einige Notationen vornehmen. Für eine allgemeine Formulierung der Implementierung verwenden wir die Notation `innerInt`(τ), um die Anwendung des inneren Integrators mit der Schrittweite τ zu beschreiben. Zur Beschreibung der Fixpunktiteration werden wir zudem die Notation

$$\widehat{\mathbf{Y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_{k-1} \end{bmatrix}$$

verwenden. Damit lässt sich der Pseudo-Code für die Startwertprozedur wie folgt beschreiben.

Algorithmus 4 Startwertprozedur für exponentielle Mehrschrittverfahren mit multiple time stepping (kurz: `StartValMTS`)

Input: t_0, y_0, τ, r, `tol`, `innerInt`
Definiere $p_0(t) = y_0$, $\tau_{\text{inner}} = \frac{\tau}{r}$ und setze $l = 0$
Wende `innerInt`(τ_{inner}) auf (3.49) an und erhalte $\hat{y}_m$ für $m = 1, \ldots, k-1$.
Definiere $\widehat{\mathbf{Y}}^{[1]} = \widehat{\mathbf{Y}}$
while `fehler` $>$ `tol` **do**
  Setze $l = l + 1$
  Bestimme das lokale Interpolationspolynom p_0 zu den Punkten $\big(t_j, g(t_j, \hat{y}_j)\big)_{j=0}^{k-1}$
  Wende `innerInt`(τ_{inner}) auf (3.49) an und erhalte $\hat{y}_m$ für $m = 1, \ldots, k-1$.
  Definiere $\widehat{\mathbf{Y}}^{[l]} = \widehat{\mathbf{Y}}$
  Berechne `fehler` anhand von $\widehat{\mathbf{Y}}^{[l]}$ und $\widehat{\mathbf{Y}}^{[l-1]}$
end while
Output: Startwerte $\hat{y}_1, \ldots, \hat{y}_{k-1}$.

Die Implementierung der multiple time stepping Varianten exponentieller Mehrschrittverfahren erfolgt in der Umsetzung einfacher als die Implementierung allgemeiner exponentieller Mehrschrittverfahren. Die Beschreibung der Algorithmen unterscheidet sich dagegen kaum. Zur Vollständigkeit haben wir diese aufgeführt.

Algorithmus 5 MTS – EXPMS$_k^d$-Verfahren

Input: y_0, tspan $= [t_0, T]$, τ, r, `innerInt`
Initialisiere: $n = 0, \quad t = t_0, \quad \mathbf{y}_n = y_0, \quad \tau_{\text{inner}} = \frac{\tau}{r}$
if $k \geq 2$ **then**
  Bestimme k Startwerte $\mathbf{y}_0, \ldots, \mathbf{y}_{k-1}$ mit `StartValMTS`
  Berechne $g_0 = g(t_0, \mathbf{y}_0), \ldots, g_{k-1} = g(t_{k-1}, \mathbf{y}_{k-1})$
  Setze $n = n + k - 1, \quad t = t + (k-1)\tau, \quad \mathbf{y}_n = \mathbf{y}_{k-1}$
end if
while $(t < T)$ **do**
  Wende `innerInt`(τ_{inner}) auf (3.18) an und erhalte $\mathbf{y}_{n+1} \approx y(t_n + \tau)$
  Berechne $g_{n+1} = g(t_{n+1}, \mathbf{y}_{n+1})$ und setze $n = n + 1, \quad t = t + \tau$
end while
Output: $\mathbf{y}_i \approx y(t_i)$ mit $t_i = t_0 + i\tau$ und $i = 1, \ldots, n$

Algorithmus 6 MTS – EXPPC$[\mathbf{n}]_k^{(d_{\text{P}}, d_{\text{C}})}$-Verfahren

Input: y_0, tspan $= [t_0, T]$, τ, r, `innerInt`
Initialisiere: $n = 0, \quad t = t_0, \quad \mathbf{y}_n = y_0, \quad \tau_{\text{inner}} = \frac{\tau}{r}$
if $k \geq 2$ **then**
  Bestimme k Startwerte $\mathbf{y}_0, \ldots, \mathbf{y}_{k-1}$ mit `StartValMTS`
  Berechne $g_0 = g(t_0, \mathbf{y}_0), \ldots, g_{k-1} = g(t_{k-1}, \mathbf{y}_{k-1})$
  Setze $n = n + k - 1, \quad t = t + (k-1)\tau, \quad \mathbf{y}_n = \mathbf{y}_{k-1}$
end if
while $(t < T)$ **do**
  Wende `innerInt`(τ_{inner}) auf (3.18) an und erhalte $\bar{\mathbf{y}}_{n+1} \approx y(t_n + \tau)$
  Berechne $\hat{\bar{g}}_{n+1} = g(t_{n+1}, \bar{\mathbf{y}}_{n+1})$
  Wende `innerInt`(τ_{inner}) auf (3.27) an und erhalte $\mathbf{y}_{n+1} \approx y(t_n + \tau)$
  Berechne $g_{n+1} = g(t_{n+1}, \mathbf{y}_{n+1})$ und setze $n = n + 1, \quad t = t + \tau$
end while
Output: $\mathbf{y}_i \approx y(t_i)$ mit $t_i = t_0 + i\tau$ und $i = 1, \ldots, n$

4.2 Anwendungsbeispiele

Die bisher vorgestellten Zeitintegratoren werden wir in diesem Abschnitt in verschiedenen Beispielen anwenden. Das Hauptaugenmerk setzen wir dabei auf das Verhalten des numerischen Zeitintegrators. Die physikalische Bedeutung und Interpretation sowie die räumliche Diskretisierung werden wir aus dem Grund kurz ansprechen und entsprechend

referieren. Wir möchten an dieser Stelle bemerken, dass diese Beispiele aus der Zusammenarbeit mit Busch, Hochbruck und Niegemann [10] stammen.

Wir betrachten die Maxwell-Gleichungen (differentielle Form)

$$\begin{aligned} \frac{\partial}{\partial t}\vec{D}(t,x) - \nabla \times \vec{H}(t,x) &= -\vec{J}(t,x) \\ \frac{\partial}{\partial t}\vec{B}(t,x) + \nabla \times \vec{E}(t,x) &= 0 \\ \nabla \cdot \vec{D}(t,x) &= \rho \\ \nabla \cdot \vec{B}(t,x) &= 0, \end{aligned} \tag{4.3}$$

wobei $x \in \Omega$ mit $x := [x_1, x_2, x_3]^{\mathrm{T}}$ gilt und die Terme entsprechend der folgenden Tabelle zu verstehen sind.

Notation	**Bezeichnung**	**mathematische Beschreibung**
$\vec{D}$	elektrische Flussdichte	$\vec{D} : \mathbb{R}_+ \times \Omega \to \mathbb{R}^3$
$\vec{H}$	magnetische Feldstärke	$\vec{H} : \mathbb{R}_+ \times \Omega \to \mathbb{R}^3$
$\vec{B}$	magnetische Flussdichte	$\vec{B} : \mathbb{R}_+ \times \Omega \to \mathbb{R}^3$
$\vec{E}$	elektrische Feldstärke	$\vec{E} : \mathbb{R}_+ \times \Omega \to \mathbb{R}^3$
$\vec{J}$	Stromdichte	$\vec{J} : \mathbb{R}_+ \times \Omega \to \mathbb{R}^3$
ρ	Ladungsdichte	$\rho : \mathbb{R}_+ \times \Omega \to \mathbb{R}$

Die ersten zwei Gleichungen der Maxwell-Gleichungen (4.3) bezeichnen wir als Rotationsgleichungen und die übrigen zwei Gleichungen als Divergenzgleichungen. Zudem setzen wir die Beziehungen

$$\begin{aligned} \vec{D}(t,x) &= \varepsilon(x)\vec{E}(t,x) \\ \vec{B}(t,x) &= \mu(x)\vec{H}(t,x) \end{aligned} \tag{4.4}$$

voraus, wobei $\varepsilon(x) := \varepsilon_0\varepsilon_r(x)$ die Permitivität und $\mu(x) := \mu_0\mu_r(x)$ die Permeabilität beschreibt. Speziell in unseren Beispielen gelten die physikalischen Vorraussetzengen $\vec{J} = 0$ und $\rho = 0$. Unter diesen Voraussetzungen (4.4) vereinfachen sich die Maxwell-Gleichungen (4.3) zu

$$\begin{aligned}
\varepsilon(x)\frac{\partial}{\partial t}\vec{E}(t,x) &= \nabla \times \vec{H}(t,x) \\
\mu(x)\frac{\partial}{\partial t}\vec{H}(t,x) &= -\nabla \times \vec{E}(t,x) \\
\nabla \cdot \big(\varepsilon(x)\vec{E}(t,x)\big) &= 0 \\
\nabla \cdot \big(\mu(x)\vec{H}(t,x)\big) &= 0.
\end{aligned} \tag{4.5}$$

Man kann zeigen, dass die Divergenzbedingungen für $t > t_0$ erfüllt bleiben, soweit die Anfangswerte diese zur Anfangszeit $t = t_0$ erfüllen [40]. Für die Beschreibung der zeitlichen Entwicklung anhand numerischer Zeitintegratoren beschränken wir uns aus dem Grund auf die Rotationsgleichungen und verweisen für weitere Details auf die Dissertation von Niegemann [40].

4.2.1 Simulation eines Ringresonators in 2D

In diesem Beispiel nehmen wir an, dass die Wellenausbreitung in x_3-Richtung verschwindet, so dass die partiellen Ableitungen ∂_{x_3} den Wert 0 annehmen. Unter dieser Voraussetzung können die Maxwell-Gleichungen (4.5) in zwei Differentialgleichungssysteme entkoppelt werden [40]. Zum einen sind es die TE-Polarisationsgleichungen

$$\begin{aligned}
\partial_t E_1 &= \frac{1}{\varepsilon}\partial_{x_2} H_3 \\
\partial_t E_2 &= -\frac{1}{\varepsilon}\partial_{x_1} H_3 \\
\partial_t H_3 &= \frac{1}{\mu}\left(\partial_{x_2} E_1 - \partial_{x_1} E_2\right)
\end{aligned} \tag{4.6}$$

und zum anderen die TM-Polarisationsgleichungen

$$\begin{aligned}
\partial_t H_1 &= -\frac{1}{\mu}\partial_{x_2} E_3 \\
\partial_t H_2 &= \frac{1}{\mu}\partial_{x_1} E_3 \\
\partial_t E_3 &= \frac{1}{\varepsilon}\left(\partial_{x_1} H_2 - \partial_{x_2} H_1\right).
\end{aligned} \tag{4.7}$$

Zur Simulation eines Ringresonators [43, Abschnitt 4.1], [44, Abschnitt 5] betrachten wir die TE-Gleichungen (4.6) über dem Gebiet Ω (s. Abbildung (4.1)) mit „Perfectly-Matched-Layer“ (PML) Randbedingungen [40, Abschnitt 3.2, Abschnitt 5.2].

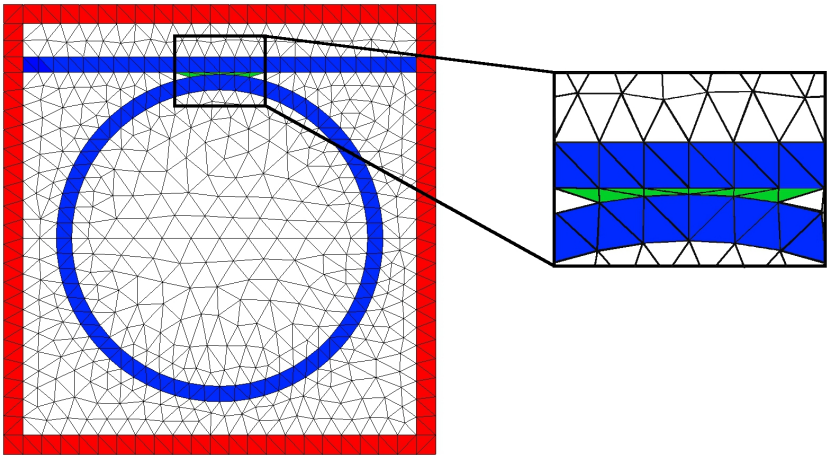

Abbildung 4.1: Triangulierung von Ω. Die Wellenleiter werden durch die blauen, der Bereich der lokalen Verfeinerung wird durch die grünen und die PML anhand der roten Elemente beschrieben.

Aus der räumlichen Diskretisierung mit Discontinuous-Galerkin-Verfahren resultiert eine gewöhnliche Differentialgleichung der Form (2.1) mit einem linearen Anteil, den wir mit A bezeichnen. Für die räumliche Diskretisierung der Gleichungen (4.6) wurde ein Discontinuous-Galerkin-Verfahren der Ordnung 3 verwendet und als numerischen Fluss haben wir einen „Upwind-Flux“ benutzt [27, 28, 43]. Aus der Abbildung 4.1 können wir leicht erkennen, dass die grünen Dreiecke deutlich kleiner sind, als die übrigen Dreiecke.

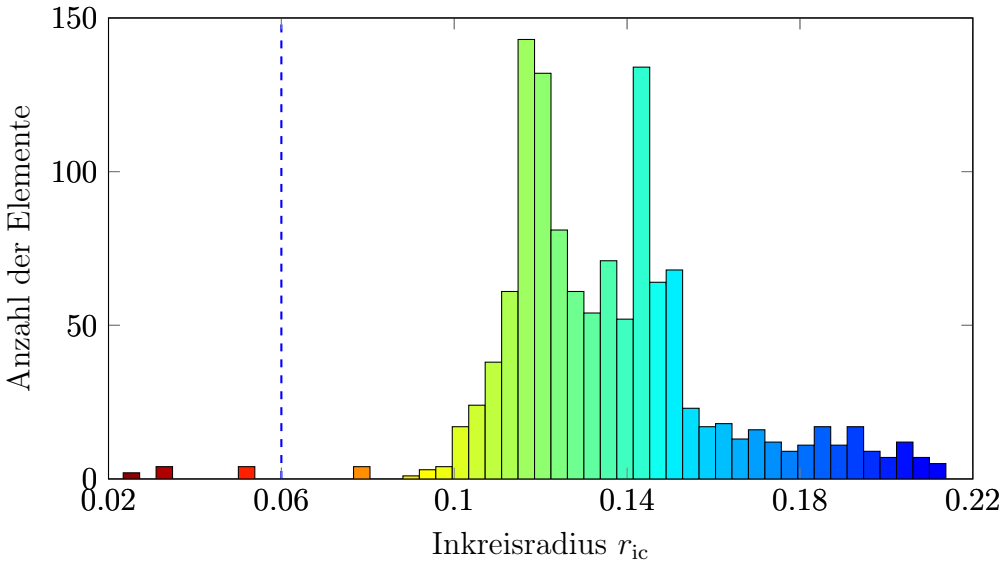

Abbildung 4.2: Inkreisradien der Elemente aus der Diskretisierung (4.1) von Ω.

Auf den linearen Anteil A wenden wir ein Splitting der From (3.37b) an und ordnen die Komponenten der Dreiecke mit $r_{\text{ic}} < 0.06$ zur Matrix S zu. Mit der gestrichelten blauen Linie in Abbildung 4.2 haben wir dieses geometrische Splitting gekennzeichnet. Die Anzahl der Freiheitsgrade in unserem Testbeispiel beträgt insgesamt $N = 38760$ und nur 300 von diesen Komponenten wurden als „steif“ identifiziert. In Abbildung 4.3 sind die 200 größten Eigenwerte von A und $A - M$ dargestellte, wobei M entsprechend dem Splitting (3.37b) definiert ist. Wir können erkennen, dass die Eigenwerte von $A - M$ um einen Faktor vier kleiner sind, als die Eigenwerte von A. Gemäß der verwendeten Diskretisierung beobachten wir eine Eigenwert-Verteilung, wie es in [27, Kap. 8, Abb. 8.8, Abb. 8.13, Abb. 8.23] bzw. [42] beschrieben wird. Zur Bestimmung optimierter exponentieller Mehrschrittverfahren eignet sich demnach (2.74) (s. Beispiel 2.22) als Zielgebiet.

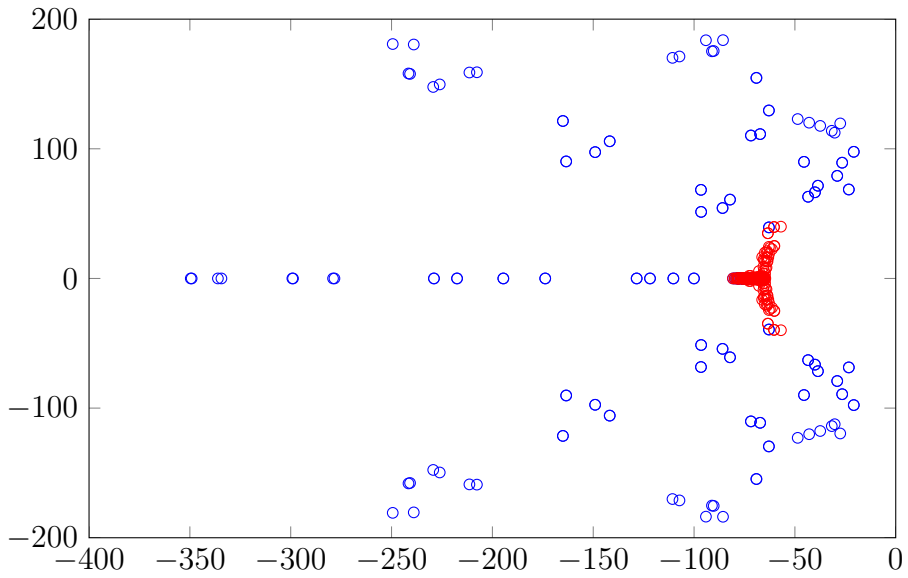

Abbildung 4.3: Eigenwerte von A (blau) und $A - M$ (rot).

Als Zeitintegrator für die Simulation des Ringresonators auf $[t_0, T]$ verwenden wir ein MTS $-$ EXPMS$_6^4$-Verfahren mit $r = 2$. Das klassische Runge-Kutta-Verfahren haben wir in diesem Fall als inneren Integrator verwendet. In Abbildung 4.4 ist der Fehler des MTS $-$ EXPMS$_6^4$-Verfahrens in Abhängigkeit der Schrittweite τ logarithmisch dargestellt, wobei der Fehler zur Endzeit $T = t_0 + 1$ in der $\|\cdot\|_\infty$-Norm gemessen wurde. Entsprechend unseren theoretischen Resultaten können wir erkennen, dass das Verfahren die gewünschte Konvergenzordnung vier besitzt.

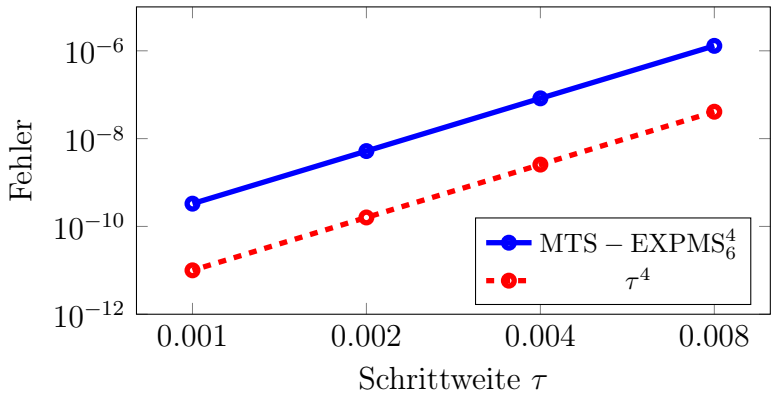

Abbildung 4.4: Fehler des MTS – EXPMS$_6^4$-Verfahrens in Abhängigkeit der Schrittweite τ in logarithmischer Darstellung.

4.2.2 3D Benchmark-Test

Im zweiten Beispiel betrachten wir die Rotationsgleichungen (4.5) zusammen mit „Perfectly Electric Conductor“ (kurz: PEC) Randbedingungen, d.h. für $x \in \partial\Omega$ fordern wir

$$n \times \vec{E} = 0 \quad \text{und} \quad n \cdot (\mu \vec{H}) = 0. \tag{4.8}$$

Für den Benchmark-Test werden wir in Abhängigkeit der Zeit eine Eigenwelle (engl. eigenmode) in einem leeren Würfel Ω betrachten [3, Kapitel 8.3]. In Abbildung 4.5 ist die Zerlegung von Ω mit 320 Tetraedern dargestellt.

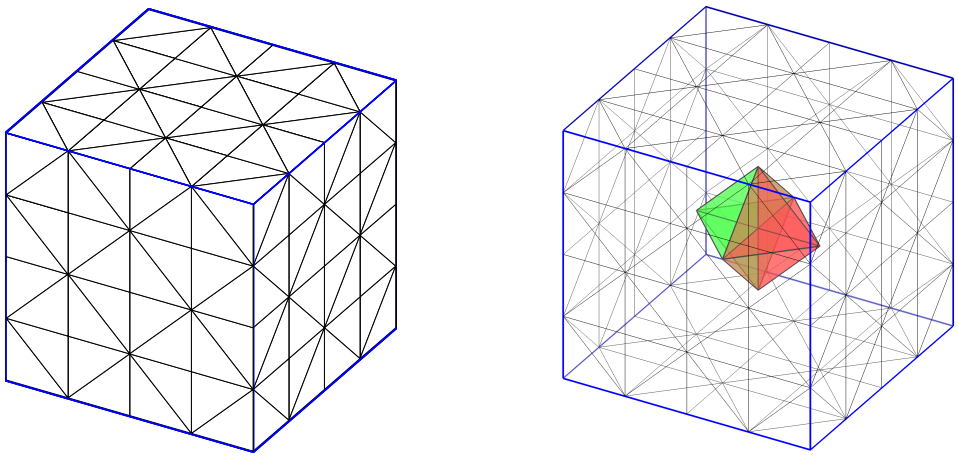

Abbildung 4.5: Diskretisierung von Ω mit Hilfe von Tetraedern. Links: Regelmäßige Zerlegung von Ω, Rechts: Die inneren 8 Tetraeder, die die gitterinduzierte Steifheit hervorrufen.

Eine geometrisch induzierte Steifigkeit ergibt sich, wenn der Mittelpunkt des Würfels (s. Abbildung 4.5) entlang einer Koordinatenachse verschoben wird und die Kanten der Elemente beibehalten werden. Die Umformung bewirkt, dass die inneren vier Tetraeder jeweils einen kleineren Inkugelradius besitzen als die übrigen Elemente, wobei wir den Inkugelradius als Maß für die „Gitterweite“ verwenden.

Sei r_s der Verfeinerungsfaktor, der die Skalierung der Inkugelradien der vier Tetraeder beschreibt. In unserem Vergleich verwenden wir sechs verschiedene Gitter mit $r_s = 2^j$ für $j = 0, \ldots, 5$ und für deren Diskretisierung verwenden wir Discontinuous-Galerkin-Verfahren der Ordnungen $p_{\mathrm{DG}} \in \{3, 4, 5, 6\}$. Hierbei vergleichen wir das Verhalten des optimierten low-storage Runge-Kutta-Verfahrens der Ordnung 4 mit 14 Stufen [41] (kurz: LSRK$(14, 4)$) mit dem Verhalten des MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahrens. Als innerer Integrator wurde beim MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahren ebenfalls das LSRK$(14, 4)$-Verfahren verwendet. Zudem wird das MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahren mit einem Splitting der Form (3.37a) ausgeführt, wobei nur die Komponenten der vier Tetraeder im Inneren als „steif“ identifiziert wurden, die die „kleinsten“ Inkugelradien besitzen.

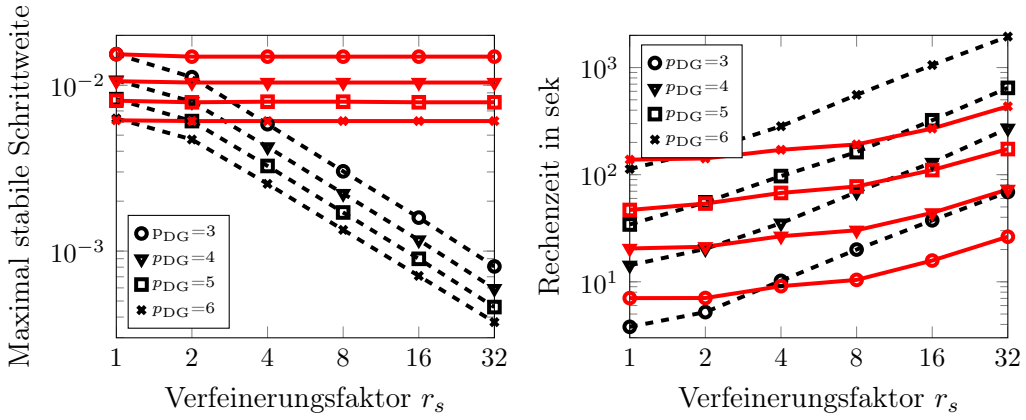

Abbildung 4.6: Vergleich zwischen dem LSRK$(14, 4)$-Verfahren (schwarz) und dem MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahren (rot) für DG-Diskretisierungen mit $p_{\mathrm{DG}} \in \{3, \ldots, 6\}$. Links: Maximale Schrittweite in Abhängigkeit des Verfeinerungsfaktors r_s, Rechts: Rechenzeit (in Sekunden) in Abhängigkeit des Verfeinerungsfaktors r_s unter Verwendung der maximal stabilen Zeitschrittweite.

Aus der Abbildung 4.6 erkennen wir, dass der äußere Integrator des MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahrens ungefähr dieselbe Stabilität aufweist für $r_s = 1$, denn beide Verfahren wurden bezüglich der Stabilität mit demselben Zielgebiet optimiert und enthalten das Zielgebiet mit der entsprechenden Skalierung. Des Weiteren stellen wir fest, dass beim MTS$-$

EXPPC$[1]_8^{(3,4)}$-Verfahren die Schrittweite des äußeren Integrators unabhängig von der Verfeinerung gewählt werden kann. Beim LSRK$(14, 4)$-Verfahren beobachten wir hingegen, dass sich die maximale Schrittweite linear in Abhängigkeit der Verfeinerung verkleinert. Der Vergleich der Rechenzeiten zeigt, dass sich MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahren mit wachsender Ordnung p_{DG} immer mehr rentieren. Wir können auch erkennen, dass die Verwendung von MTS $-$ EXPPC$[1]_8^{(3,4)}$-Verfahren sich bei stärkerer Verfeinerung noch mehr auszahlt, da der Aufwand deutlich schwächer in Abhängigkeit der Verfeierung r_s wächst als bei LSRK$(14, 4)$.

4.2.3 Ein realistisches Anwendungsbeispiel aus der Nanophotonik

Im letzten Beispiel betrachten wir ein realistisches Problem [35], das zur Simulation von Nanoantennen benutzt wird. Hierzu wird ein Gold-Dimer verwendet, das durch zwei Goldsphären vom Radius 80nm beschrieben wird. Die Goldsphären werden durch eine kleine Lücke der Breite 1nm voneinander getrennt.

Erneut betrachten wir die Rotationsgleichungen mit PML-Randbedingungen [43]. Des Weiteren wird die Dispersion vom Goldmaterial mit dem entsprechendem Drude-Lorentz-Model [40] in das System integriert. Abhängig von der Anregung ergibt die räumliche Diskretisierung mit Discontinuous-Galerkin-Verfahren ein inhomogenes Anfangswertproblem der Form (2.1) mit einem linearen Anteil A.

Aufgrund intrinsischer Symmetrien [35] kann das Gebiet Ω um einen Faktor vier reduziert werden. Deswegen verwenden wir für das Gebiet Ω die in Abbildung 4.7 beschriebene Zerlegung. Wir können erkennen, dass die kleine Lücke zwischen den Goldsphären (s. gezoomten Bereich in Abbildung 4.7) die Verwendung sehr kleiner Tetraeder erfordert.

In Abbildung 4.8 haben wir eine kumulative Verteilung der Elemente dargestellt. Hierfür wurden die Elemente entsprechend ihrer Inkugelradien r_{ic} relativ zum Inkugelradius $r_{\mathrm{ic}}^{\mathrm{min}}$ des kleinsten Elementes sortiert. Für einen relativen Inkugelradius $r_{\mathrm{ic}}^{\mathrm{rel}} = \frac{r_{\mathrm{ic}}}{r_{\mathrm{ic}}^{\mathrm{min}}}$ können wir aus der Verteilung entnehmen, welchen prozentualen Anteil der Zerlegung die Elemente ausmachen, deren Inkugelradius kleiner oder gleich $r_{\mathrm{ic}}^{\mathrm{rel}}$ ist. Aus den Abbildungen 4.7 und 4.8 geht hervor, dass bei dem vorliegenden Problem eine gitterinduzierte Steifigkeit vorliegt.

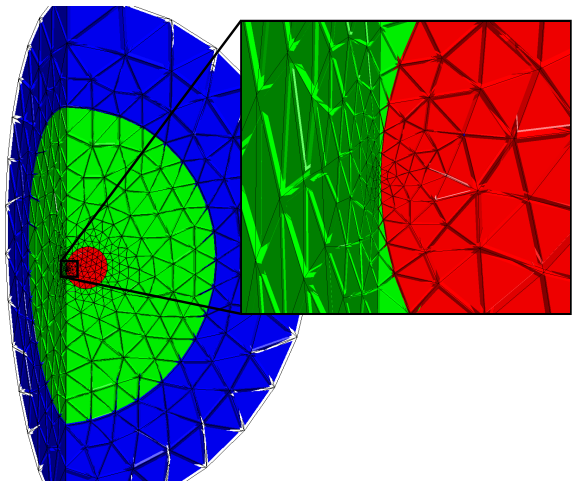

Abbildung 4.7: Diskretisierung von Ω. Die roten Elemente beschreiben die Goldsphäre und die blauen Elemente den verwendeten PML. Der Zoom veranschaulicht die feinen Tetraeder zwischen den Goldsphären.

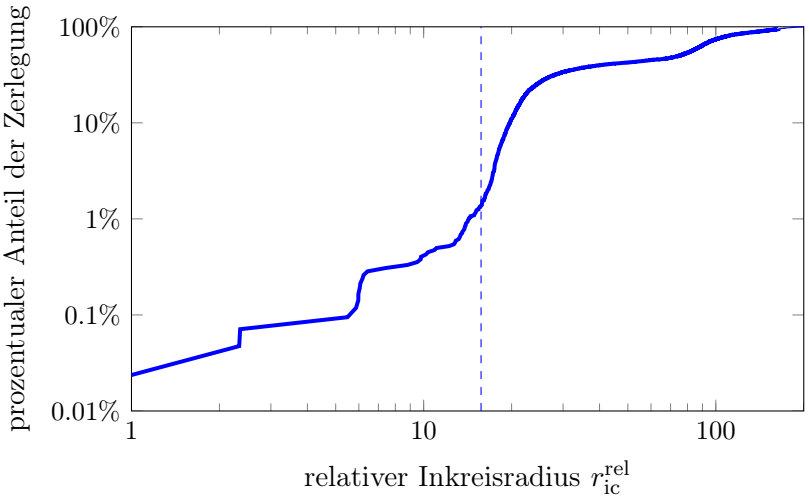

Abbildung 4.8: Kumulative Verteilung der relativen Inkugelradien $r_{\text{ic}}^{\text{rel}}$. Die gestrichelte Linie stellt die Schranke dar, die wir für das Splitting verwenden.

Wir verwenden auch diesmal ein Splitting der Form (3.37a). In Abbildung 4.8 ist eine Schranke als senkrechte Linie gekennzeichnet. Diese Schranke verdeutlicht, dass die Komponenten der Elemente, deren Inkugelradius kleiner oder gleich dieser Schranke sind, als steif identifiziert werden. Mit diesem Splitting besteht der feine Anteil aus 58 Elementen und der grobe Anteil aus 4164 Elementen.

Wie in Abschnitt 4.2.2 vergleichen wir die MTS − EXPPC$[1]_k^{(3,4)}$-Verfahren für $k \in \{4, 6, 8\}$ mit dem LSRK$(14, 4)$-Verfahren. Als inneren Integrator haben wir jeweils das LSRK$(14, 4)$-Verfahren verwendet. Die Simulation wurde mit beiden Verfahren ausgeführt, wobei jeweils die maximal stabile Schrittweite gewählt wurde. In Abbildung 4.9 ist zu erkennen, wieviel Rechenzeit die MTS − EXPPC$[1]_k^{(3,4)}$-Verfahren mit $k \in \{4, 6, 8\}$ im Vergleich zum LSRK$(14, 4)$-Verfahren sparen. Entsprechend den Resultaten aus Abschnitt 4.2.2 können wir auch hier erkennen, dass die multiple time stepping Varianten exponentieller Mehrschrittverfahren für höhere Ordnungen sich stärker rentieren. Insgesamt lässt sich eine signifikante Beschleunigung mindestens vom Faktor 1.7 erkennen.

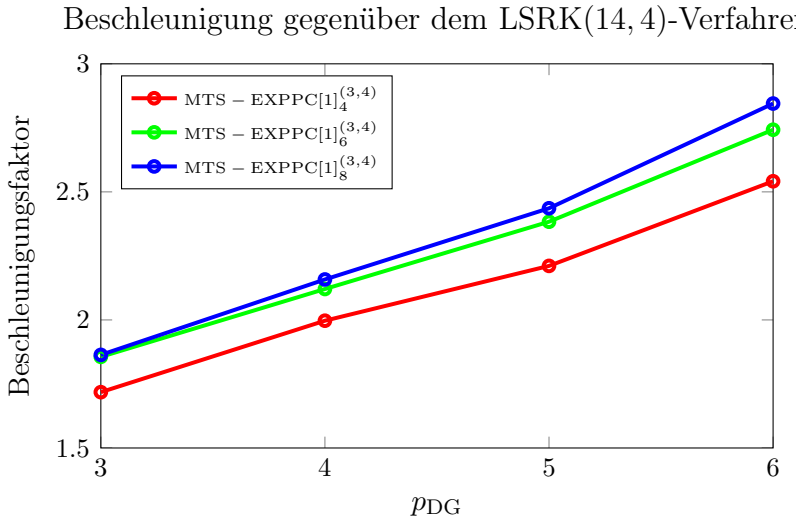

Abbildung 4.9: Rechenzeit der MTS − EXPPC$[1]_k^{(3,4)}$-Verfahren für $k \in \{4, 6, 8\}$ gegenüber der Rechenzeit des LSRK$(14, 4)$-Verfahrens (jeweils unter Verwendung der maximal stabilen Schrittweite) bei der Simulation einer Nanoantenne.

5 Nichtlinearer Fall

Im letzten Kapitel dieser Arbeit werden wir uns mit nichtlinearen Anfangswertproblemen beschäftigen. Wir setzen dabei die Existenz und Eindeutigkeit einer Lösung als gegeben voraus. Wir werden die Idee der multiple time stepping Varianten exponentieller Mehrschrittverfahren für semilineare Probleme auf nichtlineare Anfangswertprobleme verallgemeinern.

5.1 Multiple time stepping Verfahren für nichtlineare Probleme

Betrachten wir ein nichtlineares Anfangswertproblem der Form (1.5), wobei wir die Funktionen kurz mit

$$\mathbf{f} = f_{\text{fine}} \quad \text{und} \quad \mathbf{g} = f_{\text{coarse}}$$

notieren, so entspricht $\mathbf{f}$ dem steifen und $\mathbf{g}$ dem nichtsteifen Teil der Differentialgleichung. Des Weiteren setzen wir voraus, dass die Funktionen $\mathbf{f}$ und $\mathbf{g}$ einer einseitigen Lipschitz-Bedingung

$$\begin{aligned} \operatorname{Re}\langle \mathbf{f}(t,y) - \mathbf{f}(t,z), y-z\rangle &\leq l_{\mathbf{f}} \|y-z\|^2, \quad l_{\mathbf{f}} \leq 0 \\ \operatorname{Re}\langle \mathbf{g}(t,y) - \mathbf{g}(t,z), y-z\rangle &\leq l_{\mathbf{g}} \|y-z\|^2, \quad l_{\mathbf{g}} \leq 0 \end{aligned} \tag{5.1}$$

und die Funktion $\mathbf{g}$ zusätzlich einer Lipschitz-Bedingung

$$\|\mathbf{g}(t,y) - \mathbf{g}(t,z)\| \leq \mathcal{L}_{\mathbf{g}} \|y-z\|, \quad \mathcal{L}_{\mathbf{g}} \geq 0. \tag{5.2}$$

genügt. Folgen wir der Idee der multiple time stepping Verfahren aus Abschnitt 3.2, so wird der nichtsteife Anteil $\mathbf{g}$ des nichtlinearen Anfangswertproblems

$$y' = \mathbf{f}(t,y) + \mathbf{g}(t,y), \quad y(t_0) = y_0$$

auf $[t_{j-1}, t_j]$ für $0 \leq j \leq n$ mit $0 < t_n \leq T$ und $t_j = j\tau$ durch ein Polynom p_j der Form (2.36) approximiert, wobei die Koeffizienten des Polynoms die Ordnungsbedingungen aus Satz 2.13 für ein $d \geq 1$ erfüllen. Demzufolge ist in jedem Zeitschritt ein Anfangswertproblem der nachstehenden Form zu lösen

$$\mathbf{y}_n' = \mathbf{f}(t_n + t, \mathbf{y}_n) + p_n(t_n + t), \quad \mathbf{y}_n(0) = y_n, \quad t \in [0, \tau]. \tag{5.3}$$

Das Verfahren, das die Lösung von (5.3) exakt bestimmt, bezeichnen wir als das nichtlineare Mehrschrittverfahren zum Polynom p_n. Dieses Verfahren entspricht zugleich dem äußeren Integrator des multiple time stepping Verfahrens. Zur Bestimmung der Approximation $\mathbf{y}_{n+1}$ des multiple time stepping Verfahrens mit

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n(\tau) \approx y(t_{n+1}) \tag{5.4}$$

wird schließlich ein innerer Integrator mit der Schrittweite $\tau_{\text{inner}} = \frac{\tau}{r}$ auf (5.3) angewendet, wobei der Parameter r entsprechend (3.41) gewählt werden kann.

Mit NLMTS $-$ MS$_k^d$ notieren wir ein multiple time stepping Verfahren, das ein nichtlineares Mehrschrittverfahren zum Polynom p_n (2.36) als äußeren Integrator besitzt, wobei die Koeffizienten von p_n die Ordnungsbedingungen von Satz 2.13 für $d \geq 1$ erfüllen. In Anlehnung an die NLMTS $-$ MS$_k^d$-Verfahren lassen sich analog Predictor-Corrector-Verfahren konstruieren.

Prediction: (P)
Die Verwendung eines NLMTS $-$ MS$_k^d$-Verfahrens liefert nach (5.3) und (5.4) die Approximation

$$y(t_{n+1}) \approx \bar{\mathbf{y}}_{n+1}. \tag{5.5}$$

Evaluation: (E)
Aus der Auswertung der Funktion $\mathbf{g}$ mit (5.5) ergibt sich die Approximation

$$\mathbf{g}\big(t_{n+1}, y(t_{n+1})\big) \approx \mathbf{g}(t_{n+1}, \bar{\mathbf{y}}_{n+1}) =: \bar{\mathbf{g}}_{n+1}.$$

Correction: (C)
Im Correction-Prozess wird ein nichtlineares Mehrschrittverfahren zum Polynom q_n als äußerer Integrator gewählt, wobei das Polynom q_n durch

$$\begin{aligned} q_n(t_n + \theta\tau) &= \sum_{i=0}^{d_{\mathrm{C}}-1} \sum_{j=1}^{k} b_{i,j}^{\mathrm{iM}} \frac{\theta^i}{i!} \mathbf{g}^*_{n-k+1+j}, \\ \text{mit} \quad \mathbf{g}^*_{n-k+1+j} &= \begin{cases} \mathbf{g}(t_{n-k+1+j}, \mathbf{y}_{n-k+1+j}), & j \neq k \\ \mathbf{g}(t_{n-k+1+j}, \bar{\mathbf{y}}_{n-k+1+j}), & j = k \end{cases} \end{aligned} \tag{5.6}$$

gegeben ist. Das zugehörige Anfangswertproblem ist gegeben durch

$$\begin{aligned} \mathbf{y}_n' &= \mathbf{f}(t_n + t, \mathbf{y}) + q_n(t_n + t), \;\; \mathbf{y}_n(0) = y_n \\ \text{mit } \; t &\in [0, \tau] \;\text{ und }\; \mathbf{y}_n(\tau) \approx y(t_n + \tau). \end{aligned} \tag{5.7}$$

Die anschließende Verwendung eines inneren Integrators liefert uns die Approximation dieses multiple time stepping Verfahrens mit

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n(\tau) \approx y(t_n + \tau). \tag{5.8}$$

Für Predictor-Corrector-Verfahren der zweiten Art wird im Correction-Prozess anstatt (5.6) das folgende Polynom benutzt

$$\begin{aligned} q_n(t_n + \theta\tau) &= \sum_{i=0}^{d_\mathrm{C}-1} \sum_{j=0}^{k} b_{i,j}^{\mathrm{iM}} \frac{\theta^i}{i!} \mathbf{g}_{n-k+1+j}^*, \\ \text{mit} \quad \mathbf{g}_{n-k+1+j}^* &= \begin{cases} \mathbf{g}(t_{n-k+1+j}, \mathbf{y}_{n-k+1+j}), & j \neq k \\ \mathbf{g}(t_{n-k+1+j}, \bar{\mathbf{y}}_{n-k+1+j}), & j = k \end{cases}. \end{aligned} \tag{5.9}$$

Wir notieren mit $\mathrm{NLMTS} - \mathrm{PC}[\mathbf{n}]_k^{(d_\mathrm{P}, d_\mathrm{C})}$ ein multiple time stepping Verfahren bei dem der äußere Integrator des Predictor-Verfahrens einem nichtlinearen Mehrschrittverfahren zum Polynom p_n (2.36) entspricht. Für $\mathbf{n} = 1$ ist der äußere Integrator des Corrector-Verfahrens ein nichtlineares Mehrschrittverfahren zum Polynom q_n aus (5.6). Im Fall $\mathbf{n} = 2$ wird das Polynom q_n aus (5.9) gewählt. Die Koeffizienten von p_n und q_n erfüllen zudem die Ordnungsbedingungen aus Satz 2.13 für $d = d_\mathrm{P}$ und $d = d_\mathrm{C}$. Der zusätzliche Parameter $\mathbf{n}$ mit $\mathbf{n} = 1, 2$ gibt an, ob das zugrunde liegende Predictor-Corrector-Verfahren von der ersten oder zweiten Art ist.

Nachdem wir die multiple time stepping Verfahren für nichtlineare Anfangswertprobleme präsentiert haben werden wir im nächsten Abschnitt die Konvergenz solcher Verfahren analysieren.

5.2 Fehleranalyse

Die Fehleranalyse der multiple time stepping Verfahren für nichtlineare Anfangswertprobleme können wir nicht wie in Abschnitt 3.2.1 über eine Fehlerrekursion führen. Deswegen formulieren wir vorerst Abschätzungen, die für diese Fehleranalyse eine wesentliche Bedeutung darstellen und konzentrieren uns anschließend auf die Fehleranalyse der NLMTS $-$ MS_k^d-Verfahren.

Mit der Verwendung eines multiple time stepping Verfahrens wird mit dem äußeren Integrator in jedem Zeitschritt ein Anfangswertproblem der Form (5.3) beschrieben. Der äußere Integrator der NLMTS $-$ MS_k^d-Verfahren approximiert hierfür den nichtsteifen Anteil $\mathbf{g}$ im n-ten Schritt durch ein Polynom p_n der Form (2.36). Definieren wir das entsprechende Polynom zu den exakten Daten $\big(t_j, y(t_j)\big)$ mit $n-k+1 \leq j \leq n$ durch $\tilde{p}_n$, so folgt aus Lemma 2.18 die folgende Abschätzung

$$\big\|\tilde{p}_n(t) - \mathbf{g}\big(t, y(t)\big)\big\| \leq \Theta_n(\tau) \quad \text{für} \quad t \in [t_n, t_{n+1}] \quad \text{mit} \quad \Theta_n(\tau) \leq C_\Theta \tau^d \tag{5.10}$$

mit einer Konstante $C_\Theta = C_\Theta(C_{\tilde{P}}, \mathbf{g})$. Die Differenz der Polynome $p_n(t) - \tilde{p}_n(t)$ können wir für $t \in [t_n, t_{n+1}]$ wie folgt abschätzen

$$\|p_n(t) - \tilde{p}_n(t)\| \leq \kappa_n \quad \text{mit} \quad \kappa_n := C_\kappa \sum_{j=0}^{k-1} \|\mathbf{y}_{n-k+1+j} - y(t_{n-k+1+j})\|\,, \tag{5.11}$$

wobei für die Konstante $C_\kappa = C_\kappa(k, d, \mathcal{B}, \mathcal{L}_\mathbf{g})$ gilt. Wie im semilinearen Fall benötigen wir eine Voraussetzung für den inneren Integrator (vgl. Voraussetzung 3.5).

Voraussetzung 5.1: *(Innerer Integrator mit Schrittweite τ_{inner})*

> *Für die Approximation $\mathbf{y}_j \approx y_j = \mathbf{y}_{j-1}(\tau)$ des inneren Integrators zum Anfangswertproblem (5.3) gilt*
>
> $$\|\mathbf{y}_j - y_j\| \leq C_I \tau^{d+1} \quad \textit{mit} \quad C_I = C_I(\mathbf{f}, r, d, T), \tag{5.12}$$
>
> *sofern p_n die Ordnungsbedingungen (Satz 2.13) für $l = 0, \ldots, d-1$ erfüllt.*

Neben diesen Abschätzungen spielt das folgende Gronwall-Lemma in differentieller Form eine wichtige Rolle in unserer Fehleranalyse, das wir an dieser Stelle angeben möchten.

Lemma 5.2: *(Gronwall-Lemma in differentieller Form [16, Prop. 2.2])*

Sei $I = [t_0, t_1]$ und seien die Abbildungen $a : I \to \mathbb{R}$ und $b : I \to \mathbb{R}$ stetig. Die Abbildung $y : I \to \mathbb{R}$ sei zudem stetig differenzierbar auf I und erfülle die folgende Beziehung

$$y'(t) \leq a(t)y(t) + b(t) \quad \textit{für} \quad t \in I \quad \textit{und} \quad y(t_0) = y_0.$$

Unter diesen Voraussetzungen gilt die Ungleichung

$$y(t) \leq e^{A(t)}y_0 + \int_{t_0}^{t} e^{A(t)-A(s)}b(s)\mathrm{d}s \quad \textit{mit} \quad A(x) := \int_{t_0}^{x} a(r)\mathrm{d}r.$$

Für den Konvergenzbeweis der NLMTS $-$ MS$_k^d$-Verfahren benötigen wir zudem eine Beziehung zwischen den exakten Lösungen von (5.3) und (1.5). Mithilfe der Abschätzungen und dem Gronwall-Lemma 5.2 lässt sich hierfür das folgende Lemma beweisen.

Lemma 5.3: *(Fehler des äußeren Integrators)*

Sei p ein Polynom, das die Abschätzungen (5.10) mit $\Theta(\tau)$ und (5.11) mit κ erfüllt. Zudem sei vorausgesetzt, dass (5.1) und (5.2) gilt. Betrachten wir unter diesen Voraussetzungen die gewöhnlichen Differentialgleichungen

$$\begin{aligned} y' &= \mathbf{f}(t, y) + \mathbf{g}(t, y), \quad y(t_0) = y_0 \\ z' &= \mathbf{f}(t, z) + p(t), \qquad z(t_0) = z_0, \end{aligned}$$

so gilt für die Differenz $y(t) - z(t)$ mit $c := l_{\mathbf{f}} + l_{\mathbf{g}}$ die folgende Abschätzung

$$\|y(t) - z(t)\| \leq e^{c(t-t_0)} \|y(t_0) - z(t_0)\| + \int_{t_0}^{t} e^{c(t-s)} \big(\Theta(\tau) + \kappa\big) \,\mathrm{d}s.$$

Beweis: *(Lemma 5.3)*
Mithilfe der Voraussetzungen lässt sich die folgende Abschätzung bestimmen

$$\begin{aligned} 2\|y(t) - z(t)\| \frac{\mathrm{d}}{\mathrm{d}t}\|y(t) - z(t)\| &= \frac{\mathrm{d}}{\mathrm{d}t}\|y(t) - z(t)\|^2 \\ &= 2\mathrm{Re}\,\langle y'(t) - z'(t), y(t) - z(t)\rangle \\ &= 2\mathrm{Re}\,\langle \mathbf{f}(t, y) + \mathbf{g}(t, y) - \mathbf{f}(t, z) - p(t), y(t) - z(t)\rangle \\ &\leq 2c\|y(t) - z(t)\|^2 + 2\big(\Theta(\tau) + \kappa\big)\|y(t) - z(t)\|. \end{aligned}$$

Diese Ungleichung ist äquivalent zu

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\|y(t)-z(t)\| \le c\,\|y(t)-z(t)\| + \Theta(\tau) + \kappa. \tag{5.13}$$

Die Anwendung des Gronwall-Lemmas 5.2 liefert uns die Abschätzung

$$\|y(t)-z(t)\| \le e^{c(t-t_0)}\,\|y(t_0)-z(t_0)\| + \int_{t_0}^{t} e^{c(t-s)}\big(\Theta(\tau)+\kappa\big)\,\mathrm{d}s,$$

womit die Behauptung bewiesen ist. □

Mit dieser Vorarbeit können wir die Konvergenz der NLMTS – MS_k^d-Verfahren beweisen.

Satz 5.4: *(Konvergenz von* NLMTS – MS_k^d*-Verfahren)*

Die Voraussetzungen (5.1) und (5.2) seien erfüllt. Zudem sei die Funktion G mit $G(t) := \mathbf{g}\big(t, y(t)\big)$ hinreichend glatt und die Schrittweite τ sei beschränkt durch eine hinreichend kleine Konstante $\mathcal{T}$ mit $0 \le \tau \le \mathcal{T}$. Des Weiteren sei vorausgesetzt, dass der innere Integrator der Voraussetzung 5.1 genügt. Betrachten wir für hinreichend genaue Startwerte

$$\|y(t_j) - \mathbf{y}_j\| \le c_0\tau^d \quad \textit{für} \quad j = 1,\dots,k-1, \tag{5.14}$$

die numerische Approximation des NLMTS – MS_k^d*-Verfahrens, so ist der Fehler für $0 \le n\tau \le T$ gleichmäßig beschränkt durch*

$$\|y(t_n) - \mathbf{y}_n\| \le C\tau^d, \quad C = C(C_I, k, d, \mathcal{B}, T, c_0, \mathcal{L}_{\mathbf{g}}, \mathbf{g})$$

mit einer Konstante C die von T abhängt, aber unabhängig ist von n und τ.

Beweis: *(Satz 5.4)*
Definieren wir den Fehler des NLMTS – MS_k^d-Verfahrens durch $e_n := y(t_n) - \mathbf{y}_n$ mit $e_0 = 0$, so ergibt sich mit Lemma 5.3, $c = l_{\mathbf{f}} + l_{\mathbf{g}}$ und $y_j := \mathbf{y}_{j-1}(\tau)$ die nachfolgende Abschätzung

$$\begin{aligned}\|e_n\| &\le \|y(t_n) - y_n\| + \|y_n - \mathbf{y}_n\| \\ &\overset{(5.12)}{\le} e^{c\tau}\,\|y(t_{n-1}) - \mathbf{y}_{n-1}\| + \tau\int_0^1 e^{c(1-\theta)\tau}\big(\Theta_{n-1}(\tau) + \kappa_{n-1}\big)\,\mathrm{d}s + C_I\tau^{d+1}\end{aligned}$$

$$
\begin{aligned}
&\leq \tau \sum_{l=0}^{n-1} e^{(n-l-1)c\tau} \left(\int_0^1 e^{c(1-\theta)\tau} \big(\Theta_l(\tau) + \kappa_l\big)\, \mathrm{d}s + C_I \tau^d \right) \\
&\overset{c\leq 0}{\leq} \tau \sum_{l=0}^{n-1} \big(\Theta_l(\tau) + \kappa_l + C_I \tau^d\big) \\
&\leq C \max_{1\leq j\leq k-1} \|e_j\| + C\tau \sum_{l=0}^{n-1} \|e_l\| + C\tau^d,
\end{aligned}
$$

wobei für die Konstanten $C = C(C_\kappa, k)$ und $C := C(C_I, C_\Theta, c_0, \mathbf{g}, T)$ gilt. Die Behauptung folgt schließlich aus der Anwendung des diskreten Gronwall-Lemmas (2.17) □

Den Konvergenzbeweis zu NLMTS – PC$[\mathbf{n}]_k^{(d_\mathrm{P},d_\mathrm{C})}$-Verfahren können wir auf eine ähnliche Weise führen.

Satz 5.5: *(Konvergenz von* NLMTS – PC$[\mathbf{n}]_k^{(d_\mathrm{P},d_\mathrm{C})}$*-Verfahren)*

a) *Die Voraussetzungen (5.1) und (5.2) seien erfüllt. Zudem sei die Funktion G mit $G(t) := \mathbf{g}\big(t, y(t)\big)$ hinreichend glatt und die Schrittweite τ sei beschränkt durch eine hinreichend kleine Konstante $\mathcal{T}$ mit $0 \leq \tau \leq \mathcal{T}$. Des Weiteren sei vorausgesetzt, dass die inneren Integratoren der Voraussetzung 5.1 genügen. Betrachten wir für hinreichend genaue Startwerte*

$$\|y(t_j) - \mathbf{y}_j\| \leq c_0 \tau^d \quad \textit{für} \quad j = 1, \ldots, k-1 \tag{5.15}$$

die numerische Approximation des NLMTS – PC$[\mathbf{n}]_k^{(d,d)}$*-Verfahrens mit $\mathbf{n} = 1, 2$, so ist der Fehler für $0 \leq n\tau \leq T$ gleichmäßig beschränkt durch*

$$\|y(t_n) - \mathbf{y}_{n+1}\| \leq C\tau^d, \quad C = C(C_I, k, d, \mathcal{B}, T, c_0, \mathcal{L}_\mathbf{g}, \mathbf{g})$$

mit einer Konstante C die von T abhängt, aber unabhängig ist von n und τ.

b) *Es gelten die Voraussetzungen aus Teil a). Für die Startwerte gelte zudem die Abschätzung*

$$\|y(t_j) - \mathbf{y}_j\| \leq c_0 \tau^{d+1} \quad \textit{für} \quad 1 \leq j \leq k-1. \tag{5.16}$$

Dann ist der Fehler der numerischen Approximation des NLMTS – PC$[\mathbf{n}]_k^{(d,d+1)}$-

Verfahrens mit $\mathbf{n} = 1, 2$ *für* $0 \le n\tau \le T$ *gleichmäßig beschränkt durch*

$$\|y(t_n) - \mathbf{y}_n\| \le C\tau^{d+1}, \quad C = C(C_I, k, d, \mathcal{B}, T, c_0, \mathcal{L}_\mathbf{g}, \mathbf{g})$$

mit einer Konstante C *die von* T *abhängt, aber unabhängig ist von* n *und* τ*.*

Beweis: *(Satz 5.5)*
Wir beweisen diese Aussage für NLMTS $-$ PC$[\mathbf{n}]_k^{(d,d)}$-Verfahren mit $\mathbf{n} = 2$. Im Fall $\mathbf{n} = 1$ gilt der Beweis analog.

Teil a): Wir definieren den Fehler des NLMTS $-$ PC$_k^{(d,d)}$-Verfahrens durch $e_n := y(t_n) - \mathbf{y}_n$ mit $e_0 = 0$. Entsprechend der Voraussetzungen gelten für den Prediction-Prozess die Abschätzungen

$$\begin{aligned}
&\left\|\tilde{p}_n(t) - \mathbf{g}\big(t, y(t)\big)\right\| \le \Theta_n^{\mathrm{P}}(\tau) \ \text{ für } \ t \in [t_n, t_{n+1}] \ \text{ mit } \ \Theta_n^{\mathrm{P}}(\tau) \le C_{\Theta^{\mathrm{P}}}\tau^d, \\
&\|p_n(t) - \tilde{p}_n(t)\| \le \kappa_n^{\mathrm{P}} \quad \text{mit} \quad \kappa_n^{\mathrm{P}} := C_{\kappa^{\mathrm{P}}} \sum_{j=0}^{k-1} \|e_{n-k+1+j}\|, \\
&\|\bar{y}_{n+1} - \bar{\mathbf{y}}_{n+1}\| \le C_I\tau^{d+1} \quad \text{mit} \quad \bar{y}_{n+1} := \bar{\mathbf{y}}_n(\tau).
\end{aligned} \tag{5.17}$$

Die Abschätzungen zum Correction-Prozess notieren wir durch

$$\begin{aligned}
&\left\|\tilde{q}_n(t) - \mathbf{g}\big(t, y(t)\big)\right\| \le \Theta_n^{\mathrm{C}}(\tau) \ \text{ für } \ t \in [t_n, t_{n+1}] \ \text{ mit } \ \Theta_n^{\mathrm{C}}(\tau) \le C_{\Theta^{\mathrm{C}}}\tau^d, \\
&\|q_n(t) - \tilde{q}_n(t)\| \le \kappa_n^{\mathrm{C}} \quad \text{mit} \quad \kappa_n^{\mathrm{C}} := C_{\kappa^{\mathrm{C}}} \sum_{j=0}^{k} \left\|\mathbf{y}^*_{n-k+1+j} - y(t_{n-k+1+j})\right\|, \\
&\|y_{n+1} - \mathbf{y}_{n+1}\| \le C_I\tau^{d+1} \quad \text{mit} \quad y_{n+1} := \mathbf{y}_n(\tau),
\end{aligned} \tag{5.18}$$

wobei $\mathbf{y}^*_{n-k+1+j}$ wie folgt definiert ist

$$\mathbf{y}^*_{n-k+1+j} = \begin{cases} \mathbf{y}_{n-k+1+j}, & j \ne k \\ \bar{\mathbf{y}}_{n-k+1+j}, & j = k \end{cases}.$$

Nach Lemma 5.3 ergibt sich mit $c := l_\mathbf{f} + l_\mathbf{g}$ die Fehlerabschätzung

$$\begin{aligned}
\|e_n\| &\le \|y(t_n) - y_n\| + \|y_n - \mathbf{y}_n\| \\
&\le e^{c\tau}\|y(t_{n-1}) - \mathbf{y}_{n-1}\| + \tau \int_0^1 e^{c(1-\theta)\tau}\left(\kappa_{n-1}^{\mathrm{C}} + \Theta_{n-1}^{\mathrm{C}}(\tau)\right) \mathrm{d}\theta + C_I\tau^{d+1}
\end{aligned}$$

$$
\begin{aligned}
&\leq \tau \sum_{l=0}^{n-1} e^{(n-l-1)c\tau} \left(\int_0^1 e^{c(1-\theta)\tau} \left(\kappa_l^{\mathrm{C}} + \Theta_l^{\mathrm{C}} \right) \mathrm{d}\theta + C_I \tau^d \right) \\
&\overset{c\leq 0}{\leq} \tau C \sum_{l=0}^{n-1} \left(\left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\| \right) + \|y(t_{l+1}) - \bar{\mathbf{y}}_{l+1}\| + \Theta_l^{\mathrm{C}}(\tau) \right) + C\tau^d \\
&\leq \tau C \sum_{l=0}^{n-1} \left(\left(\sum_{j=0}^{k-1} \|e_{l-k+1+j}\| \right) + \|y(t_{l+1}) - \bar{y}_{l+1}\| + C_I \tau^{d+1} + \Theta_l^{\mathrm{C}}(\tau) \right) + C\tau^d \\
&\leq \tau C \sum_{l=0}^{n-1} \left(\left(\sum_{j=l-k+1}^{l} \|e_j\| \right) + e^{c\tau} \|e_l\| + \tau \int_0^1 e^{c(1-\theta)\tau} \left(\Theta_l^{\mathrm{P}}(\tau) + \kappa_l^{\mathrm{P}} \right) \mathrm{d}\theta + \Theta_l^{\mathrm{C}}(\tau) \right) \\
&\quad + C\tau^d \\
&\leq \tau C \sum_{l=0}^{n-1} \left(\|e_l\| + \tau \Theta_l^{\mathrm{P}}(\tau) + \Theta_l^{\mathrm{C}}(\tau) \right) + C\tau^d \\
&\leq C \max_{1\leq j\leq k-1} \|e_j\| + \tau C \sum_{l=0}^{n-1} \|e_l\| + C\tau^d.
\end{aligned}
$$

Die Konstanten sind nicht identisch mit den Konstanten im Beweis von Satz 5.4, doch die Abhängigkeiten der Konstanten stimmen überein. Die Anwendung des diskreten Gronwall-Lemmas liefert uns letztlich die Behauptung.

Teil b): Unter diesen Voraussetzungen gelten anstatt von (5.18) die nachstehenden Abschätzungen

$$
\begin{aligned}
&\left\| \tilde{q}_n(t) - \mathbf{g}\big(t, y(t)\big) \right\| \leq \Theta_n^{\mathrm{C}}(\tau) \ \text{ für } \ t \in [t_n, t_{n+1}] \ \text{ mit } \ \Theta_n^{\mathrm{C}}(\tau) \leq C_{\Theta^{\mathrm{C}}} \tau^{d+1}, \\
&\|q_n(t) - \tilde{q}_n(t)\| \leq \kappa_n^{\mathrm{C}} \quad \text{mit} \quad \kappa_n^{\mathrm{C}} := C_{\kappa^{\mathrm{C}}} \sum_{j=0}^{k} \left\| \hat{y}_{n-k+1+j}^* - y(t_{n-k+1+j}) \right\|, \qquad (5.19) \\
&\|y_{n+1} - \hat{y}_{n+1}\| \leq C_I \tau^{d+2}.
\end{aligned}
$$

Anschließend folgt die Behauptung vollkommen analog zu *Teil a).* □

Abschließend betrachten wir im Folgenden ein wissenschaftliches Testproblem zur Verifizierung der Konvergenzordnung der vorgestellten multiple time stepping Verfahren.

Beispiel 5.6: *(Verifizierung der Konvergenzordnung)*
Wir betrachten das nichtlineare gewöhnliche Anfangswertproblem

$$\begin{aligned} \frac{d}{dt}u(t) &= \frac{1}{u(t)} - v\frac{e^{t^2}}{t^2} - t, \\ \frac{d}{dt}v(t) &= \frac{1}{v(t)} - e^{t^2} - 2te^{-t^2} \\ \text{mit} \quad u(1) = 1, &\quad v(1) = e^{-1} \quad \text{für} \quad t \geq 1. \end{aligned} \tag{5.20}$$

Die exakte Lösung des Anfangswertproblems (5.20) ist gegeben durch

$$u(t) = \frac{1}{t} \quad \text{und} \quad v(t) = e^{-t^2}.$$

Auf dieses Anfangswertproblem wenden wir zum einen NLMTS $-$ MS$_k^d$-Verfahren mit $k \in \{3, 4\}$ und $d = k$ und zum anderen NLMTS $-$ PC$_k^{(d_\mathrm{P}, d_\mathrm{C})}$-Verfahren mit $k = 6$ und $(d_\mathrm{P}, d_\mathrm{C}) = (2, 3)$ sowie $k = 8$ und $(d_\mathrm{P}, d_\mathrm{C}) = (3, 4)$ an. Das Konvergenzverhalten dieser Verfahren untersuchen wir mit zwei unterschiedlichen „nichtlinearen Splittings“

$$f_1 = \left(\frac{1}{u},\, \frac{1}{v}\right)^{\mathrm{T}}, \qquad g_1 = \left(-v\frac{e^{t^2}}{t^2} - t,\, -e^{t^2} - 2te^{-t^2}\right)^{\mathrm{T}}, \tag{5.21}$$

$$f_2 = \left(\frac{1}{u} - v\frac{e^{t^2}}{t^2} - t,\, 0\right)^{\mathrm{T}}, \quad g_2 = \left(0,\, \frac{1}{v} - e^{t^2} - 2te^{-t^2}\right)^{\mathrm{T}}. \tag{5.22}$$

In den Abbildungen 5.1 und 5.2 wird jeweils der Fehler der jeweiligen numerischen Verfahren in Abhängigkeit der Schrittweite τ logarithmisch dargestellt. Als inneren Integrator haben wir bei allen Verfahren das eingebettete Runge-Kutta-Verfahren mit Schrittweitensteuerung verwendet, das in MATLAB als `ode45` implementiert ist. Das Anfangswertproblem wurde hierbei auf dem Zeitintervall $[1, 1.4]$ numerisch approximiert. Den numerischen Fehler e_n zur Zeit $t_n = 1.4$ haben wir mit dem folgenden Ausdruck gemessen

$$e_n = |u_n - u(t_n)| + |v_n - v(t_n)|\,. \tag{5.23}$$

In beiden Fällen beobachten wir, dass die Ordnung der multiple time stepping Verfahren bei einem nichtlinearen Anfangswertproblem der Form (1.5) erhalten bleibt.

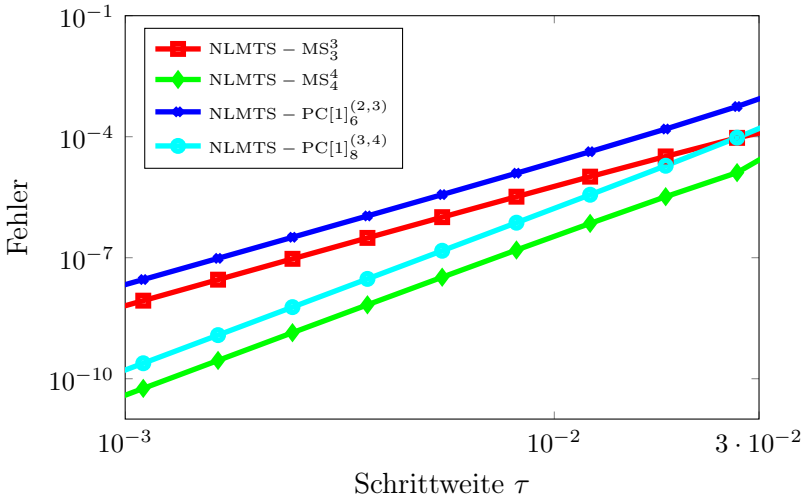

Abbildung 5.1: Logarithmische Darstellung des Fehlers (5.23) in Abhängigkeit der Schrittweite τ bei einem Splitting der Form (5.21).

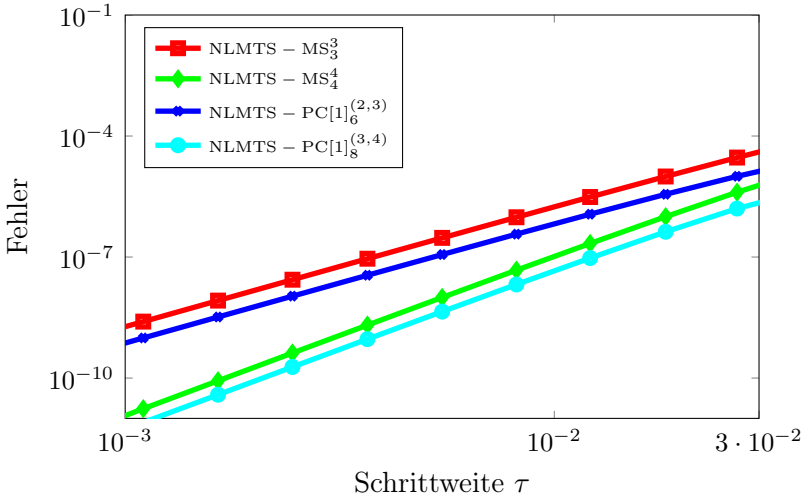

Abbildung 5.2: Logarithmische Darstellung des Fehlers (5.23) in Abhängigkeit der Schrittweite τ bei einem Splitting der Form (5.22).

Literatur

[1] A. H. Al-Mohy, N. J. Higham,
Computing Matrix Functions,
Acta Numerica, vol. 19, 159-208 (2010).

[2] A. H. Al-Mohy, N. J. Higham,
Computing the Action of the Matrix Exponential with an Application to Exponential Integrators,
SIAM Review, vol. 33(2), 488-511 (2011).

[3] C. A. Balanis,
Advanced Engineering Electromagnetics,
John Wiley & Sons, (1989).

[4] L. Bergamaschi, M. Vianello,
Efficient Computation of the Exponential Operator for Large, Sparse, Symmetric Matrices,
Numerical Linear Algebra with Applications, vol. 7, 27-45 (2000).

[5] G. Beylkin, J.M. Keiser, L. Vozovoi,
A New Class of Time Discretization Schemes for the Solution of Nonlinear PDEs, Journal of Computational Physics vol. 147, 362-387 (1998).

[6] M. Caliari,
Accurate evaluation of divided differences for polynomial interpolation of exponential propagators,
Computing, vol. 80, 189-201 (2007).

[7] M. Caliari, M. Vianello, L. Bergamaschi,
Interpolating, discrete advection-diffusion propagators at Leja sequences,
Journal of Computational and Applied Mathematics, vol. 172, 79-99 (2004).

[8] M. P. Calvo, C. Palencia,
A class of explicit multistep exponential integrators for semilinear problems,
Numerische Mathematik, vol. 102, 367-381 (2006).

[9] J. Certaine,
The solution of ordinary differential equations with large time constants,
Mathematical Methods for Digital Computers, Wiley, New York, 1960, pp. 128-132.

[10] A. Demirel, J. Niegemann, K. Busch, M. Hochbruck,
Efficient Multiple Time-Stepping Algorithms of Higher Order,
Preprint, http://na.math.kit.edu/download/papers/jcp-mts-noSplitting.pdf, (2014).

[11] J. Diaz, M. Grote,
Energy conserving explicit local time-stepping for second order wave equations,
SIAM Journal on Scientific Computing, vol. 31, 1985-2014 (2009).

[12] V. L. Druskin, L. A. Knizhnerman
Error bounds in the simple Lanczos procedure for computing functions of symmetric matrices and eigenvalues,
USSR Computational Mathematics and Mathematical Physics, vol. 31, 20-30 (1991).

[13] V. L. Druskin, L. A. Knizhnerman
On application of the Lanczos method to solution of some partial differential equations,
Journal of Computational and Applied Mathematics, vol. 50, 255-262 (1994).

[14] V. L. Druskin, L. A. Knizhnerman
Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic,
Numerical Linear Algebra with Applications, vol. 2, 205-217 (1995).

[15] M. Eiermann, O. G. Ernst,
A restarted Krylov subspace method for the evaluation of matrix functions,
SIAM Journal on Numerical Analysis, vol. 44, 2481-2504 (2006).

[16] E. Emmerich,
Discrete versions of Gronwall's lemma and their application to the numerical analysis of parabolic problems, Preprint No. 637, Fachbereich Mathematik, TU Berlin, (1999).

[17] K.J. Engel, R. Nagel,
One-Parameter Semigroups for Linear Evolution Equations,
Springer, New York, Graduate Texts in Mathematics vol. 194, 2000.

[18] E. Gallopoulos, Y. Saad,
Efficient solution of parabolic equations by Krylov approximation methods,
SIAM Journal on Scientific and Statistical Computing, vol. 13, 1236-1264 (1992).

[19] C. W. Gear, D. R. Wells,
Multirate linear multistep methods,
BIT Numerical Mathematics, vol. 24, 484-502 (1984).

[20] V. Grimm, M. Hochbruck,
Rational approximation to trigonometric operators, BIT Numerical Mathematics, vol. 48, 215-229 (2008).

[21] M. Grote, T. Mitkova,
Explicit local time-stepping for Maxwell's equations,
Journal of Computational and Applied Mathematics, vol. 234, 3283-3302 (2010).

[22] M. Grote, T. Mitkova,
High-order explicit local time-stepping methods for damped wave equations,
Journal of Computational and Applied Mathematics, vol. 239, 270-289 (2013).

[23] E. Hairer, C. Lubich, G. Wanner,
Geometric Numerical Integration,
Springer, Berlin, second edition, Springer Series in Computational Mathematics vol. 31, 2006.

[24] E. Hairer, S.P. Nørsett, G. Wanner,
Solving Ordinary Differential Equations I. Nonstiff Problems,
Springer, Berlin, 2nd version, Springer Series in Computational Mathematics vol. 8, 1993.

[25] E. Hairer, G. Wanner,
Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems,
Springer, Berlin, 2nd version, Springer Series in Computational Mathematics vol. 14, 1996.

[26] D. Henry,
Geometric Theory of Semilinear Parabolic Equations,
Springer, Berlin, Lecture Notes in Mathematics vol. 840, 1981.

[27] J.S. Hesthaven, T. Warburton,
Nodal Discontinuous Galerkin Methods,
Springer, Berlin, Texts in Applied Mathematics vol. 54, 2008.

[28] J.S. Hesthaven, T. Warburton,
Nodal High-Order Methods on Unstructured Grids: Time-Domain Solution of Maxwell's Equations,
Journal of Computational Physics vol. 181, 186-221 (2002).

[29] N. J. Higham,
The Scaling and Squaring Method for the Matrix Exponential Revisited,
SIAM Review, vol. 51(4), 747-764 (2009).

[30] N. J. Higham,
Functions of Matrices: Theory and Computation,
SIAM, xx-425 (2008).

[31] M. Hochbruck,
Vorlesungsskript zu Numerik, 2010.

[32] M. Hochbruck, C. Lubich,
On Krylov Subspace Approximations to the Matrix Exponential Operator, SIAM Journal on Numerical Analysis, vol. 34, 1911-1925 (1997).

[33] M. Hochbruck, A. Ostermann,
Exponential multistep methods of Adams-type,
BIT, vol. 51, 889-908 (2011).

[34] M. Hochbruck, A. Ostermann,
Exponential integrators.
Acta Numerica, vol. 19, 209-286 (2010).

[35] J. Hoffmann, C. Hafner, P. Leidenberger, J. Hesselbarth, S. Burger,
Comparison of electromagnetic field solvers for the 3d analysis of plasmonic nanoantennas,
Proceedings SPIE vol. 7390, Modeling Aspects in Optical Metrology II, 73900J-73900J-11 (2009).

[36] W. Hundsdorfer, J. Verwer,
Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations,
Springer, Berlin, Springer Series in Computational Mathematics vol. 33, 2003.

[37] G.E. Karniadakis, M. Israeli, S.A. Orszag,
High order splitting methods for the incompressible Navier-Stokes-equations,
Journal of Computational Physics vol. 97, 414-443 (1991).

[38] A. Klöckner,
High-Performance and High-Order Simulation of Wave and Plasma Phenomena,
PhD thesis, Brown University USA (2010).

[39] C. B. Moler, C. F. Van Loan,
Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later,
SIAM Review, vol. 45(1), 3-49 (2003).

[40] J. Niegemann, Higher Order Methods for Solving Maxwell's Equations in the Time Domain,
PhD thesis, Universität Karlsuhe (KIT) (2009).

[41] J. Niegemann, R. Diehl, K. Busch
Efficient low-storage Runge-Kutta schemes with optimized stability regions,
Journal of Computational Physics, vol. 231, 364-372 (2012).

[42] J. Niegemann, R. Diehl, K. Busch
Comparison of low-storage Runge-Kutta schemes for discontinuous Galerkin time domain simulations of Maxwell's equations,
Journal of Computational and Theoretical Nanoscience vol. 7, 1572-1580 (2010).

[43] J. Niegemann, M. König, K. Busch,
Discontinuous Galerkin methods in nanophotonics, Laser & Photonics Reviews vol. 5, 773-809 (2011).

[44] J. Niegemann, M. König, K. Busch,
Simulation of optical resonators using DGTD and FDTD, Journal of Optics A: Pure and Applied Optics vol. 11, 114015 (2009).

[45] J. Niehoff,
Projektionsverfahren zur Approximation von Matrixfunktionen mit Anwendungen auf die Implementierung exponentieller Integratoren,
Heinrich-Heine-Universität Düsseldorf, Dissertation (2006).

[46] J. Niesen, W. M. Wright,
Algorithm 919: A Krylov Subspace Algorithm for Evaluating the φ-Functions Appearing in Exponential Integrators,
ACM Transactions on Mathematical Software, vol. 38, 1-19 (2012).

[47] A. Pazy,
Semigroups of Linear Operators and Applications to Partial Differential Equations,
Springer, Berlin, 2nd version, Applied Mathematical Sciences vol. 44, 1992.

[48] J. R. Rice,
Split Runge-Kutta method for simultaneous equations,
Journal of Research of the National Bureau of Standards, vol. 64B, 151-170 (1960).

[49] Y. Saad,
Analysis of some Krylov subspace approximations to the matrix exponential operator,
SIAM Journal on Numerical Analysis, vol. 29, 209-228 (1992).

[50] A. Stock,
Development and application of a multirate multistep AB method to a discontinuous Galerkin method based particle-in-cell scheme,
Technical Report 2009-34, Scientific Computing Group, Brown University USA, (2009).

[51] J. Van den Eshof, M. Hochbruck,
Preconditioning Lanczos approximations to the matrix exponential, SIAM Journal on Scientific Computing, vol. 27, 1438-1457 (2006).

Erklärung

Hiermit versichere ich, dass ich diese Dissertation selbstständig und ohne unerlaubte Hilfsmittel verfasst habe. Ich habe hierfür keine anderen als die angegebenen Quellen und Hilfsmittel verwendet. Ich habe diese Dissertation bei keiner anderen Institution eingereicht und habe bisher auch keine erfolglosen Promotionsversuche unternommen.

Karlsruhe, den 18.März

Abdullah Demirel