

Universität Stuttgart

Masterarbeit

Zeitparallele Mehrgittermethoden für die Simulation des neuromuskulären Systems

eingereicht von: Marius Nitzsche
Matrikelnummer: 2901806
eingereicht am : 14.09.2020
Betreuer: Prof. Dominik Göddeke

Inhaltsverzeichnis

1	Einleitung	5
1.1	Motivation	5
1.2	Aufbau der Arbeit	7
1.3	Notation	7
2	Grundlagen	9
2.1	Lösungsverfahren im Ort	10
2.1.1	Stationäre Defektkorrekturverfahren	11
2.1.2	Geometrische Mehrgitterverfahren	13
2.1.3	Full Approximation Scheme	17
2.1.4	Mehrgitter Reduktionsverfahren	18
2.2	Lösungsverfahren in der Zeit	21
2.2.1	Numerische Methoden zur Lösung von Anfangs-Randwertproblemen	22
2.2.2	Übersicht über zeitparallele Verfahren	24
2.2.3	Das Parareal Verfahren	31
2.3	Die Simulation des neuromuskulären Systems	35
2.3.1	Das OpenDiHu-Simualtionspaket	36
2.3.2	Referenzproblem: Diffusionsgleichung	37
2.3.3	Das Modell des neuromuskulären Systems: das Multidomain Problem	38
3	Multigrid Reduction in Time	45
3.1	Das MGRIT Verfahren	45
3.1.1	Die Komponenten des Verfahrens	45
3.1.2	Das lineare Zwei-Level MGRIT Verfahren	48
3.1.3	Das nichtlineare Multi-Level MGRIT Verfahren	48
3.1.4	Eigenschaften	49
3.1.5	Das XBraid-Softwarepaket	51
3.2	Theorie	52
3.2.1	Fehlerfortpflanzungsoperator	53
3.2.2	Fehlerabschätzung	57
3.2.3	Differentialgleichungssysteme	60
3.2.4	Nichtlinearität und Zeitabhängigkeit	61
4	Numerische Tests	63
4.1	Umsetzung	64
4.1.1	Modellparameter der Testproblemstellungen	64
4.1.2	Anwendung MGRITs auf das Multidomain Problem	65
4.1.3	Technische Aspekte	67
4.2	Parameterstudien	68
4.2.1	Einfluss des Vergrößerungsfaktors	68

Inhaltsverzeichnis

4.2.2	Einfluss der Relaxationsmethode	70
4.2.3	Einfluss der Zyklen	72
4.3	Skalierung und Laufzeit	73
4.3.1	Analyse	73
4.3.2	Laufzeitsenkung gegenüber des sequentiellen Verfahrens	75
5	Zusammenfassung und Ausblick	77
	Literaturverzeichnis	79

1 Einleitung

Dieses Kapitel skizziert den roten Faden der Arbeit. In Abschnitt 1.1 wird die Anwendung zeitparalleler Mehrgitterverfahren auf die Simulation des neuromuskulären Systems motiviert. Einen expliziten Überblick über den Inhalt der einzelnen Kapitel wird in Abschnitt 1.2 gegeben. Zu guter Letzt wird in Abschnitt 1.3 die verwendete Notation erläutert.

1.1 Motivation

Mit Hilfe partieller Differentialgleichungen (PDEs) können viele verschiedene naturwissenschaftliche Phänomene und Prozesse mathematisch beschrieben werden. Beispiele wie Bewegungsgleichungen in der klassischen Newtonschen Mechanik, Wettervorhersagen in der Meteorologie, Börsenkursvorhersagen in der Ökonomie und viele mehr zeugen von der Vielfalt sowie der Wichtigkeit von PDEs für das wissenschaftliche Verständnis der Welt. Ein weiterer großer Anwendungsbereich ist die Medizintechnik. Die Erforschung des menschlichen Körpers wird immer weiter vorangetrieben, um Krankheiten zu heilen und Leiden zu reduzieren oder sogar zu verhindern. Ein wichtiger Bestandteil des Körpers ist das neuromuskuläre System. Dieses beschreibt das Zusammenspiel zwischen Muskeln und Nervensystem. Die gesamte Skelettmuskulatur wird mit Hilfe von elektrischen Aktivitäten durch das zentrale Nervensystem gesteuert um Bewegung und Kraft zu erzeugen. Die Elektromyographie (EMG) misst Veränderungen elektrischer Potentiale an der Muskelfasermembran, welche durch Entladungen von Nervenzellen ausgelöst werden, und ermöglicht dadurch die Untersuchung des neuromuskulären Systems (Merletti R, 2004; Pfeifer und Vogt, 2004). Die Messungen werden mit Hilfe nicht invasiver Elektroden an der Oberfläche der Muskeln durchgeführt und ermöglichen eine dreidimensionale Visualisierung des Muskels. Die Methode hilft Muskel- und Nervenerkrankungen zu diagnostizieren und genau zu bestimmen. Elektromyographie ist mittlerweile eine gängige Diagnostik in Krankenhäusern.

Die experimentell bestimmten Daten der EMG sind jedoch sehr schwer zu deuten, da das neuromuskuläre System hochkomplex ist. Außerdem ist eine realistische Visualisierung eines dreidimensionalen Muskels mit Hilfe zweidimensionaler Messungen auf der Oberfläche des Muskels sehr schwierig. Ein Beispielproblem, welches bei der Interpretation der Messungen aufkommt, sind Interferenzsignale der Elektromyographie (Holobar et al., 2010). Das Bestimmen der unterschiedlichen Einflüsse verschiedener Untereinheiten des Muskels auf die Signale führt zu einem schwer lösbaren und aufwendigen Problem. Eine realistische Simulation eines Muskels und der elektromyographischen Signale kann bei der Analyse der Daten erheblich helfen. Die Simulation ermöglicht ein besseres Verständnis des neuromuskulären Systems. Vor allem der mögliche Vergleich der gemessenen und simulierten Daten macht eine Validierung der benutzten Dekodierungsalgorithmen und Klassifizierungsstrategien möglich, mit denen Ärzte Probleme diagnostizieren und Menschen helfen (Klotz et al., 2019).

Der menschliche Körper ist jedoch ein sehr großes, hochkomplexes System und damit ist eine realistische Simulation sehr aufwendig. Allein ein Muskel besteht aus mehreren motorischen Einheiten, welche die kleinste individuell steuerbare Einheit sind und aus einer Nervenzelle und mehreren Muskelfasern bestehen. Letztendlich sind bis zu eine Million Muskelfasern in einem Muskel enthalten (MacIntosh et al., 2006). Eine einfache Überschlagsrechnung berechnet die Anzahl der benötigten Differentialgleichungsevaluierungen, um eine einsekundige Kontraktion eines großen Muskels zu simulieren (O. et al., 2020). Jede der angenommenen eine Million Muskelfasern wird mit 300 linearen Lagrange Finiten Elementen diskretisiert. Das Erregungs-Kontraktions-Modell führt nun zu 58 gekoppelten, steifen, gewöhnlichen Differentialgleichungen pro Gitterpunkt. Unter Verwendung einer direkten Methode wie dem expliziten Euler-Verfahren ist eine Zeitschrittweite von 0.0001 ms vonnöten beziehungsweise zehn Millionen Zeitschritte. Zusammengerechnet müssen $1.74 \cdot 10^{17}$ Differentialgleichungsevaluierungen durchgeführt werden. Diese Rechnung zeigt, dass für eine realistische Muskelsimulation mit angemessener Laufzeit zum einen effiziente numerische Verfahren benötigt werden und zum anderen die Möglichkeit, durch Parallelisierung das Potential von Hochleistungsrechnern zu nutzen.

Eine der effizientesten Klassen numerischer Methoden zur Lösung von linearen Gleichungssystemen, welche durch die Diskretisierung partieller Differentialgleichungen mit zum Beispiel der Finiten Elemente Methode entstehen, sind Mehrgitterverfahren. Diese nutzen eine Hierarchie an unterschiedlich feinen Gittern und den glättenden Effekt einfacher Verfahren aus, um eine Lösung mit asymptotisch optimaler Konvergenz zu bestimmen: Die Konvergenz ist unabhängig von der Schrittweite des Gitters und damit unabhängig von der Problemgröße; außerdem ist die Anzahl der benötigten Rechnungen von der Menge der Freiheitsgrade linear abhängig. Dies ist gegenüber der vollen Gauß-Elimination mit kubischem Aufwand ein entscheidender Vorteil. Dadurch sind diese Löser sehr gut für große Probleme geeignet.

Um möglichst schnell große und genaue Modelle zu simulieren benötigt es neben effizienten Verfahren viel Rechenkraft. Fortschritte in der Technik führten zu immer schnelleren Recheneinheiten, doch die Weiterentwicklung der Prozessoren stößt an physikalische Grenzen. Dadurch ist es heutzutage unmöglich, dass ein Prozessor genug Rechenoperationen pro Sekunde durchführen kann, um die erwünschten aufwendigen Kalkulationen in angemessener Zeit durchzuführen. Daher bestehen moderne Computer aus mehr als einem Prozessor; Hochleistungsrechner sind mit tausenden Prozessoren ausgerüstet. Wenn es möglich ist, das Problem auf mehrere Rechenkerne aufzuteilen, sodass diese gleichzeitig arbeiten und am Ende die Lösung zusammentragen, kann erhebliche Laufzeit eingespart werden. Dieser Ansatz wurde vor allem für Probleme im Ort erforscht. Es gibt sehr effiziente Verfahren zur Zerlegung des diskretisierten Gebiets in mehrere Teilgebiete, auf welchen jeweils ein Prozessor rechnen kann. Eine Übersicht über solche Gebietszerlegungsverfahren zeigt Saad (2003).

Doch auch hier gibt es Grenzen. Parallele Verfahren verbrauchen Rechenkraft und Zeit für Kommunikation zwischen Prozessoren und für redundante Rechnungen. Daher gibt es für ein Gebiet mit einer festen Anzahl an Gitterpunkten ein Limit an Prozessoren, mit welchen das Programm effizient rechnen kann. Ab einer bestimmten Anzahl an Prozessoren dominiert die benötigte Zeit der Kommunikation zwischen den Prozessoren. Wird dieses Limit erreicht, lohnt es nicht, noch mehr Prozessoren zu nutzen, auch wenn diese zur Verfügung stehen. Ein möglicher Weg, die zusätzlich verfügbaren Prozessoren effektiv zu nutzen, ist die Parallelisierung der Zeitdimension. Traditionelle Zeitintegratoren wie das implizite Euler-Verfahren benötigen Informationen aus dem vorherigen Zeitschritt, um den aktuellen zu berechnen. Wird die Überschlagsrechnung weiter oben betrachtet,

fällt auf, dass dadurch ein großer sequentieller Block bestehend aus zehn Millionen Zeitschritten vorhanden ist. Dies führt dazu, dass aufwendige, zeitabhängige Simulationen wie die Simulation des neuromuskulären Systems nicht die gesamte Rechenkapazität eines Hochleistungsrechners effizient ausnutzen können.

An diesem Problem wird seit einigen Jahren geforscht. Mittlerweile wurden iterative Algorithmen entwickelt, welche die Parallelisierung der Zeitdimension ermöglichen. Ein vielversprechender Ansatz ist der „Multigrid Reduction in Time“ (MGRIT) Algorithmus, welcher von Friedhoff et al. (2013) vorgestellt wurde. Dieser ist ein Mehrgitter Reduktionsverfahren in der Zeit und zeichnet sich vor allem dadurch aus, dass das vorhandene, sequentielle Zeitschrittverfahren kaum angepasst werden muss. In der Dissertation von Friedhoff (2014) sowie in dem Paper von Dobrev et al. (2017) wurde untersucht und zum Teil auch gezeigt, dass MGRIT für lineare parabolische partielle Differentialgleichungen effizient konvergieren kann. Doch für komplexere nichtlineare partielle Differentialgleichungssysteme ist dies nicht garantiert (Falgout et al., 2017). In dieser Arbeit wird daher untersucht, ob und unter welchen Umständen die Simulation des neuromuskulären Systems durch das MGRIT-Verfahren profitieren kann.

1.2 Aufbau der Arbeit

Ziel dieser Arbeit ist die Untersuchung des zeitparallelen Mehrgitter Reduktionsverfahrens MGRIT angewandt auf die Simulation des neuromuskulären Systems. Im Grundlagenkapitel 2 wird die Basis des MGRIT Algorithmus gelegt. Das zeitparallele Verfahren basiert auf speziellen Mehrgitterverfahren im Ort sowie Parareal, eine Methode, welche Zeitparallelität ermöglicht. In Abschnitt 2.1 wird in grundlegende Lösungsverfahren im Ort, wie das geometrische Mehrgitterverfahren, eingeführt und in Abschnitt 2.2 werden Lösungsverfahren zeitabhängiger Differentialgleichungen thematisiert. Dabei wird eine Übersicht über zeitparallele Methoden gegeben und explizit Parareal näher betrachtet. Am Ende des Grundlagenkapitels werden in Abschnitt 2.3 die untersuchten Modellprobleme des neuromuskulären Systems vorgestellt und deren Diskretisierung beschrieben. Schließlich wird in Kapitel 3 der MGRIT Algorithmus detailliert dargestellt. In Abschnitt 3.1 wird der MGRIT Algorithmus und dessen Eigenschaften erläutert. Eine Übersicht über die existierende Theorie des Verfahrens wird in Abschnitt 3.2 gegeben. In Kapitel 4 werden numerische Tests der Anwendung MGRITs auf das neuromuskuläre System durchgeführt. Nach der Beschreibung des Testvorgangs wird eine Parameterstudie präsentiert. Die Ergebnisse werden verglichen mit dem Verhalten MGRITs angewendet auf die Wärmeleitungsgleichung. Zusätzlich wird die Skalierung des Verfahrens untersucht und zum Schluss werden in Kapitel 5 Ergebnisse zusammengefasst und ein Ausblick gegeben.

1.3 Notation

Für das bessere Leseverständnis werden an dieser Stelle einige Bemerkungen zu der verwendeten Notation gemacht. Die Bedeutung jeder verwendeten Variable beziehungsweise jedes verwendeten Parameters wird bei erstem Vorkommen des jeweiligen Symbols beschrieben. Es werden vier verschiedene Sub- beziehungsweise Superskripte verwendet:

$$(\cdot)_{c,(d)}^{a,(b)}$$

Das Superskript $(\cdot)^a$ entspricht dem klassischen mathematischen Exponenten. Das geklammerte Superskript $(\cdot)^{(b)}$ beschreibt im Regelfall das Gitterlevel, auf welchem die Variable agiert. Das Subskript $(\cdot)_c$ stellt den Gitterpunkt dar, der der Variable entspricht. Das geklammerte Subskript $(\cdot)_{(d)}$ gibt die Iteration an.

2 Grundlagen

Die durch Diskretisierung partieller Differentialgleichungen entstehenden linearen Gleichungssysteme beinhalten typischerweise sehr große, dünnbesetzte und meist strukturierte Systemmatrizen. Für die Bestimmung der numerischen Lösung solcher Systeme sind zwei verschiedene Klassen von Lösern relevant: direkte Methoden und iterative Verfahren.

Iterative Verfahren verbessern schrittweise eine Initiallösung mit Hilfe einer Serie an Aktualisierungsschritten bis zur Erfüllung eines Abbruchkriteriums. Im 19. Jahrhundert entstanden erste solche Löser durch Jacobi (1845) und später durch Seidel (1873). Doch erst mit dem besseren Nutzen von Rechenmaschinen beziehungsweise Computern in den 1940er Jahren stieg das Interesse an iterativen Verfahren. Wichtige Meilensteine waren das SOR-Verfahren (unabhängig entwickelt von Frankel (1950) und Young (1954)) und das CG-Verfahren (Hestenes und Stiefel, 1952). Erste Veröffentlichungen zu Mehrgitterverfahren stammen von Fedorenko (1962) und Bakhvalov (1966) aus den 1960er Jahren. Wirklich bekannt wurde dieser Ansatz aber erst durch Hackbusch (1978) und Brandt (1977). Brandt stellte auch Erweiterungen wie „Full Multigrid“ (FMG) oder „Fast Approximation Scheme“ (FAS) für nichtlineare Systeme vor.

Über die Jahre wurden aber auch schnelle direkte Verfahren entwickelt. Direkte Methoden berechnen innerhalb einer endlichen Anzahl an Schritten die exakte Lösung (bis auf Maschinengenauigkeit) eines linearen Gleichungssystems. Klassische direkte Methoden, wie zum Beispiel die Gauß-Elimination, sind jedoch für sehr große Systeme unpraktikabel, da sie asymptotisch kubische Rechenzeit sowie sehr viel Speicherplatz benötigen. Durch die Ausnutzung bestimmter Eigenschaften des Systems kann dieses Problem umgangen werden. Für spezielle, dünnbesetzte, strukturierte Systemmatrizen wurden mehrere effiziente direkte Löser entwickelt, von denen einige auf dem „Block Cyclic Reduction“ Verfahren basieren. Dieses wurde von Hockney (1965) vorgestellt. Wichtige Beispiele sind der Algorithmus von Buneman (1969) und die partielle bzw. totale Reduktion von Schröder et al. (1978). Neuere Algorithmen sind UMFPACK von Davis (2004) und SuperLU von Li (2005), welche jeweils auf einer LR-Zerlegung der Systemmatrix basieren und sehr effiziente Löser für große, dünnbesetzte und nicht symmetrische lineare Gleichungssysteme darstellen. Tabelle 2.1 zeigt eine Übersicht der erwähnten Verfahren und deren Rechenkomplexität für das zweidimensionale Poissonproblem; für iterative Löser ist als Abbruchkriterium die Diskretisierungsgenauigkeit gewählt. Dies ist erkennbar an dem Term $\log \epsilon$.

In dieser Arbeit ist eine Kombination der beiden Löserklassen von Interesse, die Klasse der Mehrgitter Reduktionsverfahren. Um diese zu beschreiben werden in Abschnitt 2.1 Lösungsverfahren für Ortsprobleme erläutert. Aufbauend auf den in Abschnitt 2.1.1 beschriebenen stationären Defektkorrekturverfahren, welche eine grundlegende Komponente der Mehrgitterverfahren darstellen, wird in Abschnitt 2.1.2 das klassische, geometrische Mehrgitterverfahren vorgestellt. In Abschnitt 2.1.3 wird auf das „Full Approximation Scheme“ näher eingegangen, welches die Anwendung von Mehrgitterverfahren auf nichtlineare Probleme ermöglicht. Schließlich wird in Abschnitt 2.1.4 die Idee der Mehrgitterverfahren mit einer direkten Methode, der totalen Reduktion, kombiniert. Abschnitt 2.2

Löser	Klasse	Rechenkomplexität
Gaußelimination	Direkt	$O(n^3)$
Jacobi-Verfahren	Iterativ	$O(n^2 \log \epsilon)$
Gauß-Seidel-Verfahren	Iterativ	$O(n^2 \log \epsilon)$
SOR	Iterativ	$O(n^{\frac{3}{2}} \log \epsilon)$
CG	Iterativ	$O(n^{\frac{3}{2}} \log \epsilon)$
Buneman	Direkt	$O(n \log n)$
Totale Reduktion	Direkt	$O(n)$
Mehrgitterverfahren	Iterativ	$O(n \log \epsilon)$
Mehrgitterverfahren (FMG)		$O(n)$

Tabelle 2.1: Rechenkomplexität für verschiedene Löser des zweidimensionalen Poissonproblems mit n Freiheitsgraden. Werte übernommen (Trottenberg et al., 2000).

behandelt Lösungsverfahren für Systeme, die zeitabhängig sind. In Abschnitt 2.2.1 wird kurz in die Verfahren der Zeitintegratoren eingeführt und in Abschnitt 2.2.2 wird ein Überblick über verschiedene Ansätze der Forschung der parallelen Zeitintegration gegeben. Dabei liegt ein Fokus auf dem Parareal Algorithmus, welcher in Abschnitt 2.2.3 genauer betrachtet wird. Zu guter Letzt wird in Abschnitt 2.3 das „Open Digital Human“ Projekt (OpenDiHu) vorgestellt sowie die Modellprobleme des neuromuskulären Systems und deren Diskretisierung beschrieben.

2.1 Lösungsverfahren im Ort

Partielle Differentialgleichungen können bekannterweise in drei verschiedene Grundtypen klassifiziert werden: Elliptische, parabolische und hyperbolische partielle Differentialgleichungen. elliptische Gleichungen sind im Regelfall stationäre Probleme, das heißt sie hängen nicht von der Zeit ab. Um eine Lösung dieser Klasse von Gleichungen zu bestimmen, werden Randwerte benötigt. Sowohl parabolische als auch hyperbolische partielle Differentialgleichungen sind zeitabhängig und benötigen daher neben den örtlichen Randbedingungen zeitliche Anfangswerte. Das in dieser Arbeit betrachtete „Multigrid Reduction in Time“ Verfahren dient, wie der Name verrät, als Löser für zeitabhängige Probleme. Mehrere Ansätze und Ideen der Methode, wie zum Beispiel die Verwendung einer Gitterhierarchie, stammen jedoch ursprünglich von Lösungsverfahren im Ort ab. Daher werden in diesem Unterkapitel zunächst Verfahren zur Lösung stationärer Probleme betrachtet, bevor in Abschnitt 2.2 auf die numerische Behandlung von Anfangs-Randwertproblemen näher eingegangen wird.

Um die Lösung einer stationären Differentialgleichung numerisch berechnen zu können, muss das kontinuierliche Problem im Ort diskretisiert werden. Typische Diskretisierungsverfahren sind die Methoden der Finiten Differenzen und der Finiten Elemente. Dazu wird der Raum in ein Gitter eingeteilt und die entsprechenden Ortsableitungen an den n Gitterpunkten beziehungsweise an den n Gitterelementen approximiert. Es resultiert das zu lösende lineare Gleichungssystem

$$Au = f \tag{2.1}$$

mit der Systemmatrix $A \in \mathbb{R}^{n \times n}$, der diskreten Lösung $u \in \mathbb{R}^n$ und der rechten Seite $f \in \mathbb{R}^n$. Methoden zur Lösung dieses Gleichungssystems werden in den nächsten Kapiteln präsentiert.

2.1.1 Stationäre Defektkorrekturverfahren

Dieser Abschnitt skizziert eine erste Klasse von iterativen Verfahren, die stationären Defektkorrekturverfahren. Ihre Glättungseigenschaft bildet eine Grundlage für das Mehrgitterverfahren. Inhaltlich basiert der Abschnitt auf dem Buch von Hackbusch (1994) und der Dissertation von Friedhoff (2014).

Es wird das lineare Gleichungssystem $Au = f$ aus Gleichung (2.1) mit der Systemmatrix A , der exakten Lösung u und der rechten Seite f betrachtet. Sei v eine beliebige approximative Lösung des Problems. Iterative Verfahren verbessern eine Initiallösung $v_{(0)}$ mit Hilfe einer Serie an Aktualisierungsschritten $v_{(m+1)} = \Phi(v_{(m)}, b, A)$ bis zur Erfüllung eines Abbruchkriteriums. Der Fehler der Näherungslösung v ist folgendermaßen definiert:

Definition 2.1.1 (Fehler)

Sei $u \in \mathbb{R}^n$ die exakte, diskrete Lösung der Gleichung (2.1) und sei $v \in \mathbb{R}^n$ eine Approximation dieser Lösung. Dann ist der Fehler $e \in \mathbb{R}^n$ der Näherungslösung v folgendermaßen definiert:

$$e = u - v$$

Allerdings ist die exakte Lösung im Allgemeinen unbekannt, da sie durch ein numerisches Verfahren bestimmt werden soll. Damit ist der Fehler nicht berechenbar. Ein berechenbares Maß der Güte der Näherungslösung v ist das Residuum, welches folgendermaßen definiert ist.

Definition 2.1.2 (Residuum)

Betrachtet wird Gleichung (2.1). Sei $v \in \mathbb{R}^n$ eine Approximation der exakten, diskreten Lösung $u \in \mathbb{R}^n$. Dann ist das Residuum $r \in \mathbb{R}^n$ der Näherungslösung v folgendermaßen definiert:

$$r = f - Av$$

Es gibt an, wie gut v das LGS erfüllt. Die Kombination der Definition 2.1.1 und Definition 2.1.2 liefert die Residuumsleichung.

$$Au = f \Leftrightarrow A(v + e) = f \Leftrightarrow Ae = f - Av = r \quad (2.2)$$

Gleichung (2.2) zeigt, dass mit Hilfe des Residuums und der approximativen Lösung die exakte Lösung durch eine Aktualisierung bestimmt werden kann:

$$u = v + e \Leftrightarrow u = v + A^{-1}r \quad (2.3)$$

Die Bestimmung der Inversen der Matrix A ist im Regelfall aufwändig, daher wird bei Defektkorrekturverfahren eine approximative Inverse $B \approx A^{-1}$ gewählt. Wird B in die Gleichung (2.3) eingesetzt, wird folgender allgemeine Aktualisierungsschritt erhalten.

$$v \leftarrow v + Br = (I - BA)v + Bf$$

$P := (I - BA)$ wird als Iterationsmatrix definiert. Damit kann eine allgemeine Form eines Defekt-korrekturverfahrens definiert werden:

$$v_{(k+1)} = Pv_{(k)} + Bf, \quad k = 0, 1, \dots \quad (2.4)$$

Es gibt viele verschiedene Möglichkeiten für die Wahl der Iterationsmatrix P . Mit der Zerlegung der Systemmatrix $A = D - L - U$ in den Diagonal-, den unteren und den oberen Dreiecksteil kann das (gedämpfte) Jacobi-Verfahren beschrieben werden. Das gedämpfte Jacobi-Verfahren approximiert die Inverse der Matrix A mit der Inversen des Diagonalteils und dem Dämpfungsparameter $\omega \in \mathbb{R}$, d.h. $B_{\omega J} = \omega D^{-1}$. Für $\omega = 1$ wird das ungedämpfte Jacobi-Verfahren erhalten. Eingesetzt in Gleichung (2.4) liefert die Iterationsmatrix des gedämpften Jacobi-Verfahrens $P_{\omega J} = (1 - \omega)I + \omega D^{-1}(L + U)$ und somit das Verfahren selbst:

$$\begin{aligned} v_{(k+1)} &= [(1 - \omega)I + \omega D^{-1}(L + U)]v_{(k)} + \omega D^{-1}f \\ \Leftrightarrow v_{(k+1)} &= P_{\omega J}v_{(k)} + B_{\omega J}f \end{aligned}$$

Weitere gängige Defektkorrekturverfahren sind das Gauß-Seidel-Verfahren mit $B = (D - L)^{-1}$ und der Iterationsmatrix $P_{GS} = (D - L)^{-1}U$ sowie das SOR-Verfahren, welches das Gauß-Seidel-Verfahren durch Einführung eines Relaxationsparameters verallgemeinert.

Es muss jedoch festgestellt werden, dass die hier vorgestellten stationären Defektkorrekturverfahren sehr langsam konvergieren. Daher sind sie in den meisten Fällen nicht dazu geeignet, große Probleme selbstständig zu lösen. Mit Hilfe einer umfangreichen Fourier-Analyse kann das Konvergenzverhalten theoretisch untersucht werden; in dem Buch von Wienands und Joppich (2004) kann dies detailliert nachgelesen werden. Der Spektralradius der Iterationsmatrix einer iterativen Methode

$$\rho(P) = \max\{|\lambda| \mid \lambda \text{ ist Eigenwert von } P\}$$

beschreibt die asymptotische Konvergenzeigenschaft der Methode und wird daher auch asymptotischer Konvergenzfaktor genannt. Das Verfahren konvergiert genau dann, wenn $\rho(P)$ kleiner als Eins ist. Wird zum Beispiel das Jacobi-Verfahren auf die zweidimensionale Poisson-Gleichung $-\Delta u = f$ angewendet, welche mit der zentralen Finiten Differenzen Methode diskretisiert wurde, ergibt sich der Spektralradius $\rho(P_J) = \cos(\pi h)$ (Wienands und Joppich, 2004). Tabelle 2.2 zeigt einige Spektralradien für verschiedene Gitterweiten h .

h	1/16	1/32	1/64	1/128
$\rho(P_J)$	0.98079	0.99518	0.99880	0.99970

Tabelle 2.2: Asymptotischer Konvergenzfaktor $\rho(P_J) = \cos(\pi h)$ für das Jacobi-Verfahren angewandt auf die Poisson-Gleichung mit verschiedenen Gitterweiten h

Die präsentierten Werte zeigen, dass das Jacobi-Verfahren angewandt auf das Poisson-Problem konvergiert. Jedoch ist der asymptotische Konvergenzfaktor sehr nahe an eins, dies entspricht einer langsamen Konvergenz. Außerdem ist der Faktor $\rho(P_J) = \cos(\pi h)$ abhängig von h ; je kleiner die Gitterweite desto schlechter der Konvergenzfaktor. Dies wird als h -abhängige Konvergenz bezeichnet.

Es stellt sich jedoch heraus, dass hochfrequente Teile des Fehlers von stationären Defektkorrekturverfahren schnell und effizient eliminiert werden. Das bedeutet, dass der Fehler nicht unbedingt klein wird, aber er wird glatt. Abbildung 2.1 verbildlicht dies. Dazu wurde als einfaches Beispiel

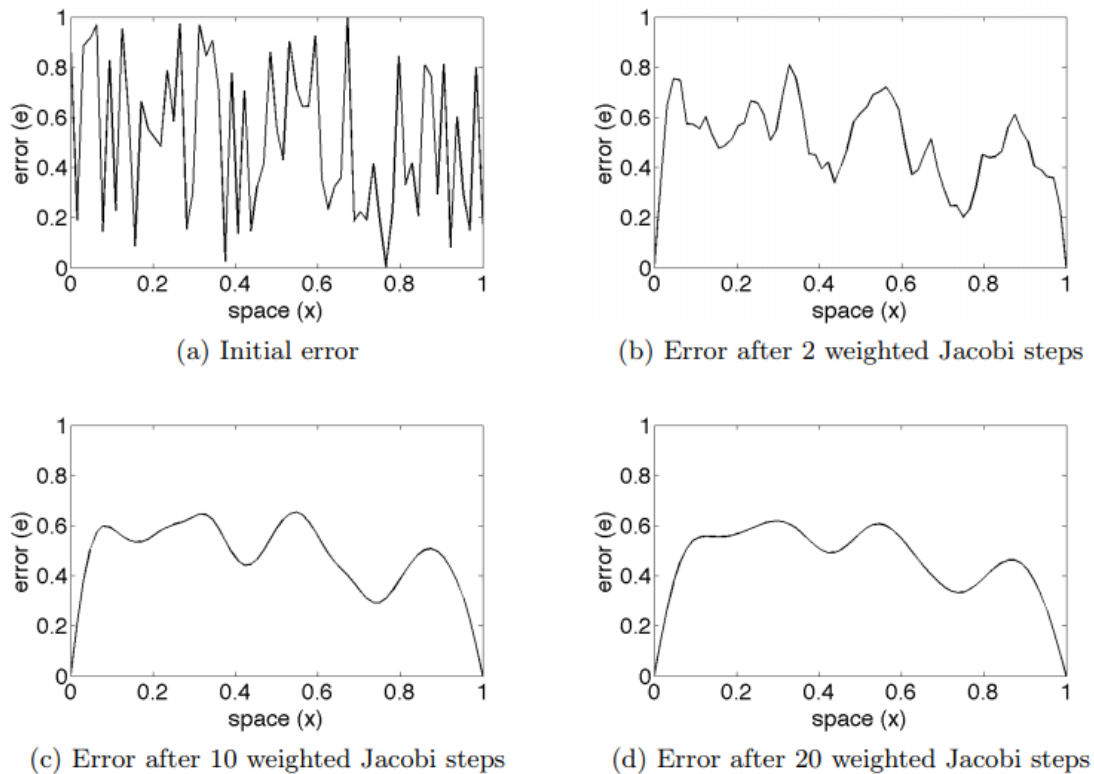


Abbildung 2.1: Fehlerreduktion eines zufälligen Fehlers durch das gedämpfte ($\omega = 2/3$) Jacobi-Verfahren angewandt auf die Laplace-Gleichung. Bild übernommen (Friedhoff, 2014).

die eindimensionale Laplace-Gleichung $-\Delta u = 0$ mit homogenen Dirichlet-Randbedingungen auf dem Einheitsintervall gewählt, welche mit zentralen Finiten Differenzen diskretisiert wurde. Abbildung 2.1 zeigt die Fehlerreduktion des gedämpften Jacobi-Verfahrens mit $\omega = 2/3$ auf einem Gitter mit $n = 64$ Punkten nach 0, 2, 10 und 20 gedämpften Jacobi Schritten.

Dieses Phänomen stammt von der lokalen Wirkungsweise der vorgestellten Methoden: Die Verfahren bilden in bestimmter Reihenfolge den Mittelwert zwischen benachbarten Werten. Dadurch werden große Unterschiede zwischen benachbarten Werten schnell geglättet; ist der Fehler jedoch glatt, werden viele Iterationen benötigt, um eine Korrektur über das ganze Gitter zu verbreiten. Dies ist die sogenannte **Glättungseigenschaft** und ist ein wichtiges Prinzip des Mehrgitterverfahrens, siehe Abschnitt 2.1.2. Werden Defektkorrekturverfahren zur Glättung eines Fehler genutzt, werden sie häufig als Relaxationsverfahren bezeichnet.

2.1.2 Geometrische Mehrgitterverfahren

Mehrgitterverfahren sind mit die effizientesten iterativen Verfahren zur Lösung eines linearen Gleichungssystems, siehe Tabelle 2.1. In dieser Arbeit wird das geometrische Mehrgitterverfahren beschrieben, bei welchem stationäre Defektkorrekturverfahren auf einer Hierarchie von größer

werdenden Gittern angewendet werden. Nach einer Motivation werden die benötigten Komponente beschrieben, welche am Ende zu einem Algorithmus zusammengefasst werden. Abschnitt 2.1.2 basiert auf den Werken von Trottenberg et al. (2000), Briggs et al. (2000) und Göttsche (2010).

Motivation

In Abschnitt 2.1.1 wird deutlich, dass stationäre Defektkorrekturverfahren zwar langsam konvergieren aber schnell hochfrequente Fehlerteile beseitigen. Das bedeutet, dass nach wenigen Iterationen des Verfahrens ein glatter Defekt vorhanden ist. Dieser kann ohne essentiellen Verlust an Informationen auf einem größeren Gitter approximiert werden. Dies bringt mehrere Vorteile mit sich: Zum einen benötigt das Lösen der Residuums Gleichung (2.2) weniger Rechenaufwand, da das System kleiner ist, und zum anderen erscheint der glatte Fehler auf einem größeren Gitter hochfrequenter. Wird das Relaxationsverfahren erneut angewendet, wird der Fehler auf dem größeren Gitter effizient geglättet. Durch wiederholte Anwendung des Prinzips wird das Gitter größer, sowie das zu lösende System und der Fehler kleiner. Ist das Problem klein genug, kann es mit Hilfe eines direkten Verfahrens schnell gelöst werden.

Nach der Berechnung des Korrekturterms auf dem größten Gitter muss dieser zum ursprünglichen, feinen Gitter transportiert werden, um die Approximationslösung zu aktualisieren. Zusätzlich muss auf jedem Gitter nochmals geglättet werden, um hochfrequente Teile des Fehlers, welche durch Transferoperationen eingeführt wurden, zu beseitigen. All diese Schritte zusammen bilden einen sogenannten V-Zyklus eines Mehrgitterverfahrens; Abbildung 2.2 veranschaulicht dies. Mehrgitterverfahren sind iterative Verfahren; ein Zyklus wird wiederholt, bis ein Konvergenzkriterium erfüllt ist.

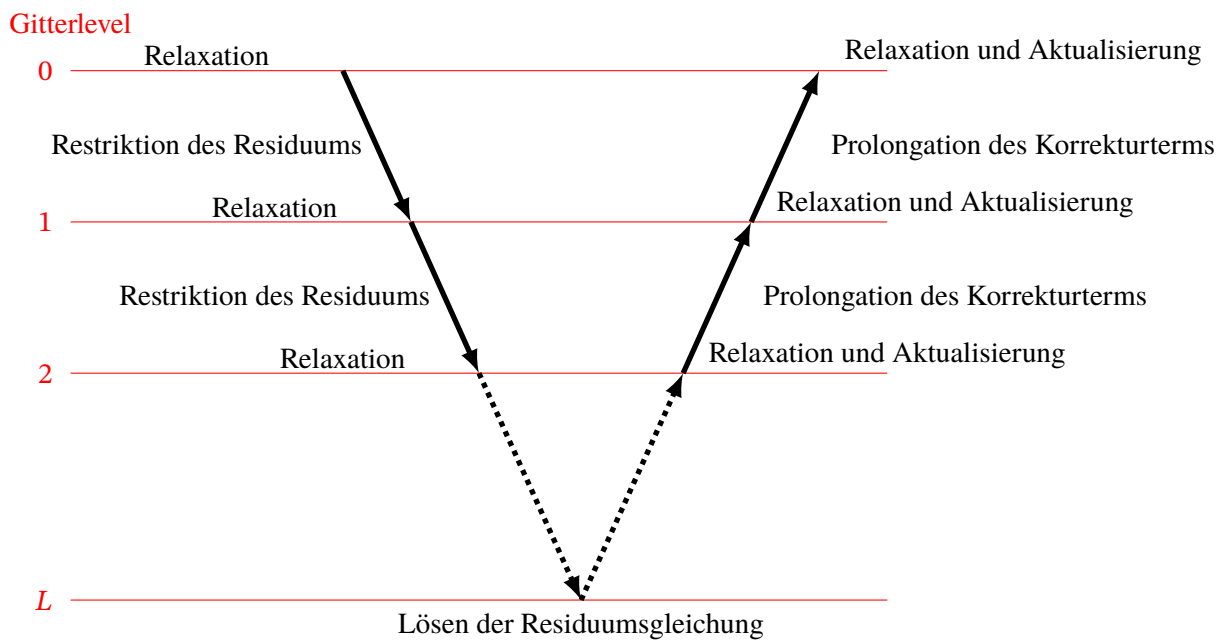


Abbildung 2.2: V-Zyklus des geometrischen Mehrgitterverfahrens

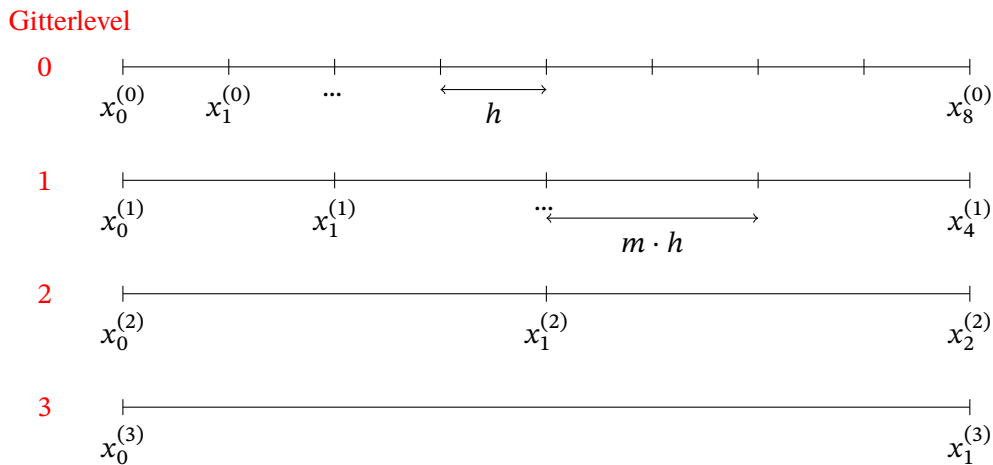


Abbildung 2.3: Beispiel einer eindimensionalen Gitterhierarchie mit $m = 2$

Komponenten

Für die genaue Definition des heuristisch erklärten Algorithmus des Mehrgitterverfahrens müssen zunächst einige Komponenten der Methode genauer betrachtet werden.

Gitter Das geometrische Mehrgitterverfahren agiert auf einer Gitterhierarchie. Das Modellproblem aus Abschnitt 2.3.2 wird mit Hilfe der Finiten Elemente Methode im Ort diskretisiert. In diesem einfachen, eindimensionalen Fall wird das Ortsgebiet in mehrere äquidistante Strecken mit Länge h eingeteilt; diese bilden die Triangulierung beziehungsweise das Gitter Ω_h . Dieses Gitter ist das feinste Gitter der Hierarchie und stellt Gitterlevel 0 dar. Wird die Gitterweite h mit einem Vergrößerungsfaktor $m > 1$ multipliziert, werden gröbere Gitter erhalten; das heißt, wenn die Gitterweite mit jedem Level verdoppelt wird, halbiert sich die Größe des zu lösenden linearen Gleichungssystems. Abbildung 2.3 zeigt ein Beispiel einer eindimensionalen Gitterhierarchie. Für mehr Dimensionen oder kompliziertere geometrische Elemente können analog Gitterhierarchien gebildet werden.

Transfer Es gibt zwei Richtungen für Transferoperationen von Daten auf der Gitterhierarchie: vom feineren Gitter zum gröberen und andersherum. Operatoren, welche benachbarte Werte des feinen Gitters mit bestimmter Gewichtung in Werten des groben Gitters zusammenfassen, werden *Restriktion* genannt. Die Injektion ist der offensichtlichste Restriktionsoperator. Dabei werden die Daten am Gitterpunkt des groben Gitters direkt von dem dazu passenden Gitterpunkt des feinen Gitters übernommen. Dies würde für das Beispiel aus Abbildung 2.3 bedeuten, dass die Daten $v_j^{(l+1)}$ an Gitterpunkt $x_j^{(l+1)}$ folgendermaßen übertragen werden:

$$v_j^{(l+1)} = v_{2j}^{(l)} \quad (2.5)$$

Eine Alternative stellt die gewichtete Restriktion dar. Diese verwendet einen gewichteten Durchschnitt benachbarter Punkte, um den Wert auf dem gröberen Gitter zu bestimmen. Analog zu Gleichung (2.5) kann dies folgendermaßen durchgeführt werden:

$$v_j^{(l+1)} = \frac{1}{4}(v_{2j-1}^{(l)} + 2v_{2j}^{(l)} + v_{2j+1}^{(l)})$$

Falls eine Finite Elemente Diskretisierung betrachtet wird, wird die L^2 -Projektion als Restriktionsoperator verwendet. Diese stellt die Bestapproximation im kleineren Raum dar.

Transferoperationen, welche Daten vom gröberen zum feineren Gitter transportieren, werden *Prolongation* beziehungsweise Interpolation genannt. Viele verschiedene Prolongationsverfahren können dafür benutzt werden. Als Beispiel wird hier die lineare Interpolation verwendet, welche folgendermaßen definiert ist:

$$v_{2j}^{(l)} = v_j^{(l+1)} \tag{2.6}$$

$$v_{2j+1}^{(l)} = \frac{1}{2}(v_j^{(l+1)} + v_{j+1}^{(l+1)}) \tag{2.7}$$

An Gitterpunkten, die auf beiden Gittern vorhanden sind, werden die Daten direkt übernommen; an den Gitterpunkten, welche nur auf dem feinen Gitter existieren, wird der Durchschnitt der benachbarten Punkte berechnet. Bei Mehrgitterverfahren mit Finiten Elementen Diskretisierungen ist die natürliche Einbettung in den größeren Raum eine übliche Wahl der Prolongation.

Relaxation Eine weitere wichtige Komponente des geometrischen Mehrgitterverfahrens sind die Relaxationsmethoden, welche den Fehler glätten und dadurch den Transfer der Daten zwischen den Gittern ohne wesentlichen Informationsverlust ermöglichen. In Abschnitt 2.1.1 wurden bereits die stationären Defektkorrekturverfahren beschrieben, welche dank ihrer Glättungseigenschaft gut dafür geeignet sind. Nur wenige Iterationen des Jacobi- oder Gauß-Seidel-Verfahrens werden zum Glätten benötigt.

Löser Auf dem größten Gitter wird das transferierte Problem gelöst. Dafür benötigt es einen effizienten Löser. Da die Größe des linearen Gleichungssystems auf dem größten Gitter klein ist, bieten sich direkte Methoden an, aber auch Krylov-Unterraum Methoden sind eine beliebte Wahl.

Algorithmus

Aus all diesen Komponenten kann nun der Algorithmus zusammengesetzt werden. Betrachtet wird eine Gitterhierarchie mit $L + 1$ Gitterlevel; Level 0 ist das feinste Gitter. Sei $A^{(0)}u = f^{(0)}$ das zu lösende diskrete Problem, $v^{(0)}$ die Initiallösung und sei $A^{(l)}, f^{(l)}$ und $r^{(l)}$ die Systemmatrix sowie der Vektor der rechten Seite und der Residuumsvektor auf Level $l = 0, \dots, L$. Algorithmus 2.1 stellt einen V-Zyklus des geometrischen Mehrgitterverfahren dar; das heißt, der Algorithmus bildet einen Iterationsschritt und wird wiederholt, bis die Residuumsnorm klein genug ist oder die maximale Anzahl an Iterationen erreicht wird.

Durch minimale Veränderungen können andere Zyklen erzeugt werden. In dieser Arbeit werden ausschließlich der V-Zyklus sowie der F-Zyklus betrachtet. Der oben beschriebene V-Zyklus durchquert genau einmal die gesamte Gitterhierarchie von fein zu grob und zurück zu fein. Der F-Zyklus

Algorithmus 2.1 Ein V-Zyklus des geometrischen Mehrgitterverfahren

```

procedure MG( $l, v^{(l)}, f^{(l)}$ )
  if  $l=L$  then
    Löse  $A^{(L)}v^{(L)} = f^{(L)}$  mit gewähltem Lösungsverfahren
  else
    Glätte  $A^{(l)}v^{(l)} = f^{(l)}$  mit  $v_0^{(l)}$  als Initiallösung und  $v_G^{(l)}$  als Resultat
    Berechne das Residuum:  $r^{(l)} = f^{(l)} - A^{(l)}v_G^{(l)}$ 
    Restringiere das Residuum  $r^{(l)} \mapsto r^{(l+1)}$  und setze  $f^{(l+1)} = r^{(l+1)}$ 
    Berechne den Korrekturterm  $v^{(l+1)} = \text{MG}(l+1, 0, f^{(l+1)})$  // Rekursiv
    Prolongiere  $v^{(l+1)} \mapsto v_I^{(l)}$ 
    Addiere den Korrekturterm  $v_K^{(l)} = v_G^{(l)} + v_I^{(l)}$ 
    Glätte  $A^{(l)}v^{(l)} = f^{(l)}$  mit  $v_K^{(l)}$  als Initiallösung und  $v^{(l)}$  als Resultat
  end if
end procedure

```

geht auf dem Weg zum größten Gitter analog vor; jedoch werden auf dem Weg nach oben größere Gitter wiederholt besucht. Abbildung 2.4 zeigt beide Zyklusarten. Der F-Zyklus benötigt verglichen mit dem V-Zyklus mehr Rechenarbeit aber konvergiert dafür im Regelfall in weniger Iterationen.

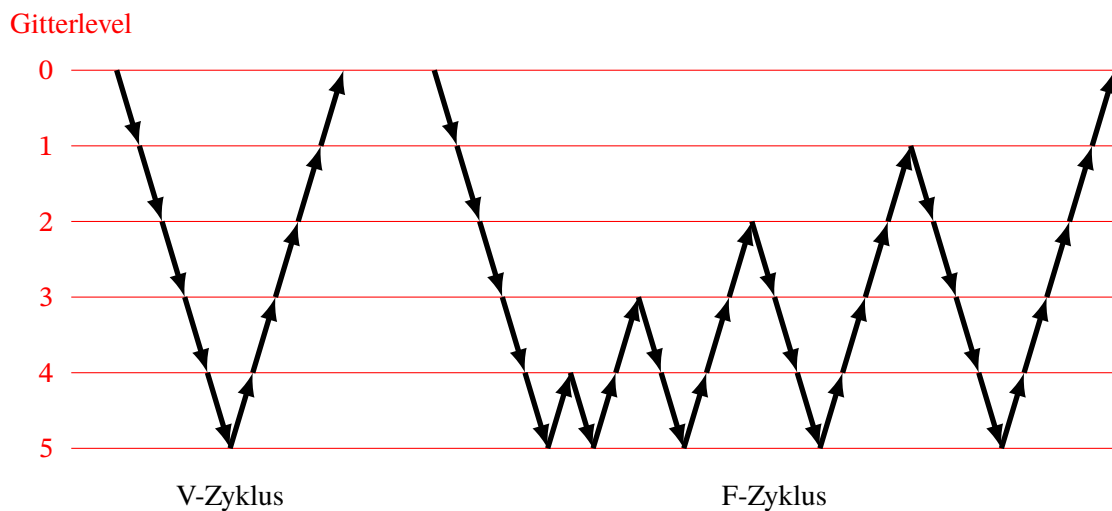


Abbildung 2.4: V-Zyklus und F-Zyklus für sechs Gitterlevel

2.1.3 Full Approximation Scheme

Das in Abschnitt 2.1.2 vorgestellte geometrische Mehrgitterverfahren kann auf lineare Probleme angewandt werden. Damit das Verfahren auch für nichtlineare Probleme konvergiert, muss dieses angepasst werden. Eine Möglichkeit stellt das von Brandt (Brandt, 1977) vorgestellte „Full Approximation Scheme“ (FAS) dar. Eine gut verständliche Einleitung dazu liefert Briggs in Briggs et al., 2000; darauf basiert dieser Abschnitt.

Zuerst wird geklärt, warum der Algorithmus 2.1 nur für lineare Aufgabenstellungen geeignet ist. In Gleichung (2.2) wird die Residuumsgleichung $r = Ae$ hergeleitet mit dem Residuumsvektor r , dem Lösungsvektor u , der approximativen Lösung v , der rechten Seite f , dem Fehler e und dem linearen Operator A . Dabei wurde jedoch die Linearität des Operators A ausgenutzt:

$$A(v + e) = f \Leftrightarrow Av + Ae = f \Leftrightarrow Ae = f - Av = r$$

Das bedeutet, für einen nichtlinearen Operator A kann $r = Ae$ nicht gefolgert werden.

Anstatt der einfachen linearen Residuumsgleichung muss daher mit einer Alternative vorgegangen werden. Dank der Definitionen des Residuums $r = f - Av$, der rechten Seite $f = Au$ und des Fehlers $e = u - v$ gelten folgende Umformungen:

$$r = f - Av \Leftrightarrow r = Au - Av \Leftrightarrow r = A(v + e) - Av \tag{2.8}$$

Betrachtet wird eine Gitterhierarchie mit zwei Gittern um das originale Problem $A^{(0)}u^{(0)} = f^{(0)}$ zu lösen. Das Ziel ist, den Fehler des feinen Gitters $e^{(0)}$ der Näherungslösung $v^{(0)}$ zu approximieren und dadurch mit jeder Iteration $v^{(0)}$ zu verbessern. Dazu wird Gleichung (2.8) auf dem groben Gitter näher untersucht.

$$r^{(1)} = A^{(1)}(v^{(1)} + e^{(1)}) - A^{(1)}v^{(1)}$$

$A^{(1)}$ ist die Systemmatrix auf dem groben Gitter und $r^{(1)}$ kann wie in Algorithmus 2.1 durch Anwendung des Restriktionsoperators auf $r^{(0)}$ bestimmt werden. Im Gegensatz zum linearen Fall wird zusätzlich $v^{(1)}$ benötigt, um $e^{(1)}$ zu berechnen. Durch die Anwendung desselben Restriktionsoperators, welcher das Residuum restringiert, ist dies möglich. Letztendlich entsteht dadurch folgendes nichtlineare Gleichungssystem, welches nach $u^{(1)} = v^{(1)} + e^{(1)}$ aufgelöst werden kann:

$$A^{(1)}u^{(1)} = A^{(1)}v^{(1)} + r^{(1)}$$

Durch Interpolation des Terms $e^{(1)} = u^{(1)} - v^{(1)}$ wird der Fehler $e^{(0)}$ approximiert. Damit ist der Aktualisierungsschritt $v^{(0)} \leftarrow v^{(0)} + e^{(0)}$ des Full Approximation Scheme durchführbar. Der Name des Verfahrens leitet sich von dem Fakt ab, dass auf dem größten Gitter das Problem nicht nach dem Fehler $e^{(1)}$, sondern nach der vollen Approximation $u^{(1)}$ gelöst wird.

Der Hauptunterschied des Full Approximation Scheme zum linearen Mehrgitterverfahren ist, dass nicht nur das Residuum sondern auch die Näherungslösung restringiert werden muss. Aus den vorhergehenden Überlegungen folgt der Algorithmus 2.2; es muss jedoch beachtet werden, dass nichtlineare Relaxationsmethoden verwendet werden müssen. Es können dieselben Transferoperationen wie im linearen Fall angewandt werden. Wird das Verfahren genau untersucht, fällt auf, dass im linearen Fall das Verfahren äquivalent zum Algorithmus 2.1 ist (Trottenberg et al., 2000).

2.1.4 Mehrgitter Reduktionsverfahren

Mehrgitterverfahren sind mit die effizientesten iterativen Methoden zur Lösung eines Gleichungssystems. Tabelle 2.1 zeigt dies. Es gibt jedoch auch direkte Löser, welche spezielle Probleme sehr schnell lösen können. Einer dieser Löser ist die Totale Reduktion von Schröder et al. (1978)). In diesem Abschnitt wird näher auf Reduktionsmethoden und deren Kombination mit Mehrgitterverfahren eingegangen. Die resultierenden Methoden werden Mehrgitter Reduktionsverfahren (MGR)

Algorithmus 2.2 Ein V-Zyklus des Full Approximation Scheme

```

procedure FASMG( $l, v^{(l)}, f^{(l)}$ )
  if  $l=L$  then
    Löse  $A^{(L)}u^{(L)} = A^{(L)}v^{(L)} + f^{(L)}$  mit gewähltem Lösungsverfahren
    Berechne den approximativen Fehler  $e^{(L)} = u^{(L)} - v^{(L)}$ 
  else
    Glätte  $A^{(l)}v^{(l)} = f^{(l)}$  mit  $v_0^{(l)}$  als Initiallösung und  $v_G^{(l)}$  als Resultat
    Berechne das Residuum:  $r^{(l)} = f^{(l)} - A^{(l)}v_G^{(l)}$ 
    Restringiere das Residuum  $r^{(l)} \mapsto r^{(l+1)}$  und setze  $f^{(l+1)} = r^{(l+1)}$ 
    Restringiere die Näherungslösung  $v_G^{(l)} \mapsto v^{(l+1)}$ 
    Berechne den approximativen Fehler  $e^{(l+1)} = \text{FASMG}(l+1, v^{(l+1)}, f^{(l+1)})$  // Rekursiv
    Prolongiere  $e^{(l+1)} \mapsto e_I^{(l)}$ 
    Addiere den Korrekturterm  $v_K^{(l)} = v_G^{(l)} + e_I^{(l)}$ 
    Glätte  $A^{(l)}v^{(l)} = f^{(l)}$  mit  $v_K^{(l)}$  als Initiallösung und  $v^{(l)}$  als Resultat
  end if
end procedure

```

genannt und wurden unter anderen von Foerster et al. (1981) und Ries et al. (1983) beschrieben. Einen schönen Überblick über das Thema gibt Friedhoff (2014), auf deren Dissertation dieses Kapitel basiert.

Eine wichtige Eigenschaft der Reduktionsverfahren ist ihre mögliche effiziente Parallelisierung. Göddeke und Strzodka (2011) nutzen diesen Ansatz zum Beispiel aus, um tridiagonale System parallel zu lösen. Dadurch wird eine starke Relaxationsmethode entwickelt, mit deren Hilfe sehr schlecht konditionierte Gleichungssysteme mit Mehrgittermethoden auf GPUs gelöst werden können. Reduktionsverfahren ermöglichen also fortgeschrittene, effizient parallelisierte Methoden.

Zyklische Reduktionsverfahren Reduktionsmethoden nutzen Strukturen bestimmter Systemmatrizen aus, wie zum Beispiel die Systemmatrix des diskretisierten Poisson-Problems. Zyklische Reduktionsverfahren basieren auf der Idee, die Hälfte der Unbekannten zu eliminieren, die verbliebenen Gleichungen zu sortieren und den Vorgang zu wiederholen. Dadurch haben sie eine große Ähnlichkeit zu Mehrgitterverfahren; tatsächlich können Reduktionsmethoden als Mehrgitterverfahren mit speziellen Transferoperatoren, Grobgitteroperatoren und Gitterhierarchien angesehen werden. Die entscheidenden Unterschiede sind jedoch, dass keine Relaxation stattfindet und dass das Problem auf dem groben Gitter so gewählt wird, dass es für die verbliebenen Unbekannten äquivalent zu dem Problem auf dem feinen Gitter ist. Dadurch wird nur ein Zyklus gebraucht, um das Problem zu lösen, und das Verfahren ist damit ein direkter Löser.

Das Prinzip der zyklischen Reduktionsverfahren ist im eindimensionalen Fall gut ersichtlich. Betrachtet wird ein tridiagonales System $Au = f$ mit $A \in \mathbb{R}^{n \times n}$ und $u, f \in \mathbb{R}^n$. Vereinfachend wird $n = 2^k + 1$ mit beliebiger natürlicher Zahl k angenommen. Das System wird nun in zwei verschiedene Arten von Unbekannten eingeteilt: solche, die eliminiert werden und deshalb nur auf dem feinen Gitter existieren und die verbleibenden, welche auch auf dem reduzierten Gitter vorhanden sind. Erstere werden F-Punkte genannt und letztere C-Punkte. Im eindimensionalen Fall ist eine mögliche Einteilung die Einteilung in Unbekannte mit ungeraden (F-Punkte) und geraden

büßen jedoch dafür ihre Eigenschaft als direkte Löser ein. Das bedeutet, diese Methoden benutzen approximierte Grobgitter-Operatoren. Der dadurch injizierte Fehler wird durch Relaxationsschritte eingedämmt.

In dieser Arbeit wird näher auf Zwei-Level MGR-CH(v, v') Verfahren (Foerster et al., 1981) eingegangen; durch das rekursive Anwenden der Methoden entstehen Multi-Level Verfahren. Alternative Mehrgitter Reduktionsverfahren wie MGR-CH(v) (Ries et al., 1983) oder das Verfahren von Braess (1984) gehen sehr ähnlich vor. Betrachtet wird eine in Abbildung 2.5 dargestellte Gitterhierarchie. Ziel ist das Lösen des Gleichungssystems $A^{(0)}u^{(0)} = f^{(0)}$. Operatoren oder Daten auf dem mittleren

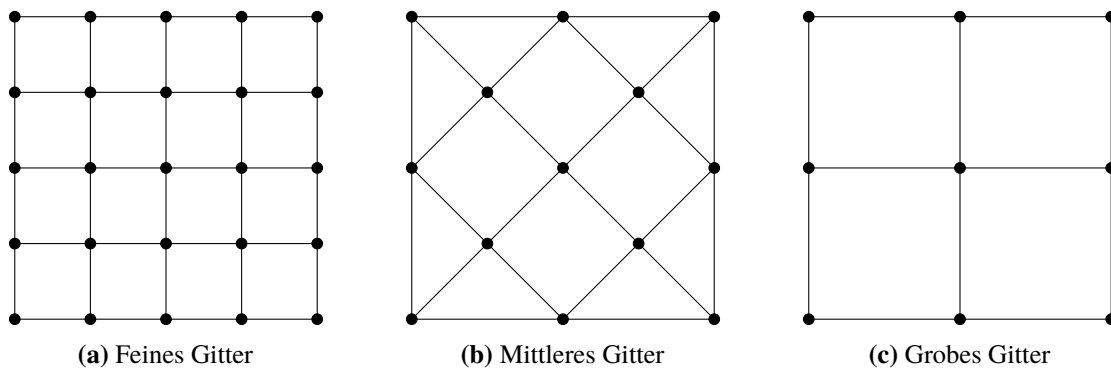


Abbildung 2.5: Gitterhierarchie einer Zwei-Level Mehrgitter Reduktion

Gitter sind mit dem Index m und auf dem groben Gitter (Gitterlevel Eins) mit dem Index Eins gekennzeichnet. Das „checkered“ beziehungsweise RB Gauß-Seidel Verfahren ist das verwendete Relaxationsverfahren und ist eine Variation des Gauß-Seidel Verfahren, welches wie Totale Reduktion auf einem wie ein Schachbrett eingeteilten Gitter arbeitet um Parallelisierung zu ermöglichen. Der Name MGR-CH stammt von dieser Wahl des Relaxationsverfahren ab. Algorithmus 2.3 beschreibt einen Zyklus des Zwei-Level MGR-CH(v, v') Verfahrens und stellt damit eine Iteration eines iterativen Verfahrens dar, welche bis eine Konvergenzbedingung erfüllt ist wiederholt wird.

2.2 Lösungsverfahren in der Zeit

Abschnitt 2.1 beschreibt Lösungsverfahren für Probleme im Ort. Doch diese reichen für viele reale Anwendungen beziehungsweise für viele partielle Differentialgleichungen, wie auch die in Abschnitt 2.3 beschriebenen, nicht aus. Zusätzlich muss die Zeitdimension beachtet werden. Die Simulation des neuromuskulären Systems beschreibt zum Beispiel eine Muskelkontraktion, also das Zusammenziehen eines Muskels. Dieser Vorgang ist kein Zustand, sondern eine Aktion über viele Millisekunden. Um die entsprechenden Anfangs-Randwertprobleme zu lösen, benötigt es eigene Diskretisierungs- und Lösungsmethoden, welche in Abschnitt 2.2.1 vorgestellt werden. In Abschnitt 2.2.2 wird näher auf die Forschung von Verfahren, welche Parallelität in der Zeit ausnutzen, eingegangen. Abschließend wird in Abschnitt 2.2.3 die Parareal Methode erläutert, welche in Kombination mit den Mehrgitter Reduktionsverfahren die Grundlage des MGRIT Verfahrens darstellt.

Algorithmus 2.3 Zwei-Level MGR-CH(v, v')

```

while Residuumsnorm > Toleranz do
    // Glättung
    Wende  $v$  RB Gauß-Seidel Schritte auf  $A^{(0)}v^{(0)} = f^{(0)}$  an mit  $v^{(0)}$  als Initiallösung und  $v_G^{(0)}$ 
    als geglättetes Resultat
    Berechne das Residuum:  $r^{(0)} = f^{(0)} - A^{(0)}v^{(0)}$ 
    // Restriktion
    Restringiere das Residuum  $r^{(l)} \mapsto r^{(m)}$  und setze  $f^{(m)} = r^{(m)}$ 
    Wende  $v'$  RB Gauß-Seidel Schritte auf  $A^{(m)}v^{(m)} = f^{(m)}$  an mit Null-Vektor als Initiallösung
    und  $v_m$  als geglättetes Resultat
    Berechne das Residuum:  $r^{(m)} = f^{(m)} - A^{(m)}v^{(m)}$ 
    Restringiere das Residuum  $r^{(m)} \mapsto r^{(1)}$  und setze  $f^{(1)} = r^{(1)}$ 
    // Lösen des reduzierten Problems

    Löse das Grobgitter Problem  $A^{(1)}v^{(1)} = f^{(1)}$  um den Korrekturterm  $e^{(1)}$  zu bestimmen
    // Interpolation
    Injiziere die Werte an den entsprechenden Gitterpunkten des mittleren Gitters und löse
     $A^{(m)}v^{(m)} = f^{(m)}$  um den Korrekturterm  $e^{(m)}$  zu bestimmen
    Injiziere die Werte an den entsprechenden Gitterpunkten des feinen Gitters und löse  $A^{(0)}v^{(0)} =$ 
     $f^{(0)}$  um den Korrekturterm  $e^{(0)}$  zu bestimmen
    // Korrektur

    Addiere den Korrekturterm  $v^{(0)} = v_G^{(0)} + e^{(0)}$ 
end while

```

2.2.1 Numerische Methoden zur Lösung von Anfangs-Randwertproblemen

Zur Lösung zeitabhängiger partieller Differentialgleichungen benötigt es neben den Methoden im Ort ebenso Methoden zur Diskretisierung und Lösung in der Zeit. Es gibt drei verschiedene Ansätze, solche Gleichungen zu diskretisieren. **Monolitische Verfahren** diskretisieren gleichzeitig in Ort und Zeit. Dadurch, dass die Unbekannten über das gesamte Raum-Zeit Gebiet gekoppelt sind, entstehen oft sehr rechenaufwendige Verfahren. Daher wird dieser Ansatz selten in der Praxis benutzt und spielt in dieser Arbeit keine Rolle; er kann jedoch für spezielle Anwendungen wie Raum-Zeit Mehrgitterverfahren sehr wirkungsvoll sein. Die **Rothe-Methode** diskretisiert das Anfangs-Randwertproblem zunächst in der Zeit, sodass eine Folge im Ort semi-diskreter Randwertaufgaben entsteht. Diese können nacheinander mit Hilfe der üblichen Ortsdiskretisierungen wie Finite Differenzen oder Finite Elemente und Lösungsverfahren im Ort gelöst werden. Genau umgekehrt, das heißt zuerst Diskretisierung im Ort und danach in der Zeit, geht die **Linienmethode** vor. Es ist situationsabhängig, ob die Rothe-Methode oder die Linienmethode die geeignetere Wahl ist; in manchen Fällen sind sie sogar äquivalent. Die Rothe-Methode ermöglicht adaptive Gitterweiten im Ort; die verwendete Zeitschrittweite muss jedoch stabil bezüglich jeder Zeitschrittweite sein. Da der Fokus dieser Arbeit auf Verfahren in der Zeit liegt und die Linienmethode eine übersichtlichere Herleitung des MGRIT Verfahrens ermöglicht, wird diese in dieser Arbeit verwendet.

Als Erstes wird die Anfangs-Randwertaufgabe mit Hilfe der Finite Differenzen Methode oder der Finiten Elemente Methode im Ort diskretisiert. Daraus entsteht ein n -dimensionales System gewöhnlicher Differentialgleichungen, welche nur noch zeitabhängig sind, der Form:

Definition 2.2.1 (Anfangswertproblem)

Sei $[a, b]$, $a, b \in \mathbb{R}$ ein reelles Intervall, $g : [a, b] \times \mathbb{R}^n \mapsto \mathbb{R}^n$ eine stetige Funktion und $u_a \in \mathbb{R}^n$. Das Anfangswertproblem ist die Bestimmung der Lösung $u : [a, b] \mapsto \mathbb{R}^n$ des folgenden n -dimensionalen Systems gewöhnlicher Differentialgleichungen mit der Bedingung $u(a) = u_a$:

$$u'(t) = g(t, u(t)), \quad u(a) = u_a, \quad t \in [a, b]$$

Hierbei wird vorausgesetzt, dass u existiert und eindeutig ist auf dem Intervall $[a, b]$.

Für die Diskretisierung des Systems in der Zeit wird ein Zeitgitter benötigt.

Definition 2.2.2 (Zeitgitter)

Für ein $N \in \mathbb{N}$ ist die Menge

$$\mathcal{T} := \{t_0, t_1, \dots, t_N\}, \quad a = t_0 < t_1 < \dots < t_N = b$$

ein Zeitgitter des Intervalls $[a, b]$ beziehungsweise eine Zerlegung in Subintervalle. Der Abstand

$$\delta t_i = t_{i+1} - t_i, \quad i = 0, \dots, n-1$$

wird als Gitterweite oder Schrittweite bezeichnet. Ist diese konstant, liegt ein äquidistantes Zeitgitter mit der Gitterweite $\delta t = \frac{b-a}{N}$ vor.

Ein Zeitschrittverfahren ist ein numerisches Verfahren zur Approximation der Lösung u eines Anfangswertproblems auf einem gegebenen Zeitgitter. Es resultiert die Gitterfunktion $u_{\mathcal{T}} : \mathcal{T} \mapsto \mathbb{R}^n$, welche einer diskreten Menge an Lösungen $\{u_0, u_1, \dots, u_N\}$ entspricht, für die gilt:

$$u_i := u_{\mathcal{T}}(t_i) \approx u(t_i), \quad \forall i = 0, 1, \dots, N \quad (2.9)$$

Eine gängige Wahl sind Einschrittverfahren, welche im Gegensatz zu Mehrschrittverfahren ausschließlich die Approximation aus dem letzten Zeitschritt nutzen. Ein generelles Einschrittverfahren wird folgendermaßen definiert:

Definition 2.2.3 (Einschrittverfahren)

Sei $\Phi : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$ eine stetige Funktion, \mathcal{T} ein beliebiges Zeitgitter des Intervalls $[a, b]$ und u_a ein beliebiger Anfangswert. Dann ist ein Einschrittverfahren folgende rekursiv definierte Gitterfunktion mit Verfahrensfunktion Φ und Vektor $f_i \in \mathbb{R}^n$, welcher alle Terme, die unabhängig von der Lösung sind, zum Zeitpunkt t_i beinhaltet:

$$\begin{aligned} u_0 &= u_a \\ u_i &= \Phi(t_{i-1}, u_{i-1}, \delta t_{i-1}) + f_i, \quad i = 1, 2, \dots, N \end{aligned}$$

Einfachheitshalber werden in dieser Arbeit äquidistante Zeitgitter sowie zeitunabhängige Zeitdiskretisierungen, das heißt die Verfahrensfunktion ist für jeden Zeitpunkt t_i gleich, betrachtet. Explizite Einschrittverfahren hängen ausschließlich von der Lösung des vorausgehenden Zeitschritts u_{i-1} ab, während implizite Verfahren zusätzlich von u_i abhängen. Ist die Funktion g des Anfangswertproblems linear, so entspricht die Anwendung der Verfahrensfunktion Φ im expliziten Fall einer

Matrix-Vektor-Multiplikation Φu_{i-1} und im impliziten Fall muss ein Gleichungssystem gelöst werden. Damit sind implizite Methoden sehr viel aufwendiger zu rechnen, sie sind jedoch im Regelfall A-stabil. Das bedeutet, diese Methoden konvergieren mit jeglicher Gitterweite im Gegensatz zu expliziten Verfahren, welche relativ kleine Schrittweiten benötigen, um stabil zu sein.

Eine der wichtigsten Klassen der Einschrittverfahren sind die Runge-Kutta-Verfahren. Diese Klasse beinhaltet Methoden verschiedener Konvergenzordnung wie das explizite und implizite Eulerverfahren der Ordnung Eins oder das Heun und das Crank-Nicolson Verfahren der Ordnung Zwei. Für komplexe partielle Differentialgleichungssysteme, wie das System zur Simulation des neuromuskulären Systems, eignen sich Splitting-Verfahren. In Abschnitt 2.3 werden die in dieser Arbeit verwendeten Verfahren beschrieben. Details zu Zeitschrittverfahren wie Runge-Kutta-Verfahren und Splitting-Verfahren können in dem Buch von Hairer et al. (2006) nachgelesen werden.

Traditionelle Zeitschrittverfahren, wie zum Beispiel die hier beschriebenen Einschrittverfahren, benötigen die Approximation des vorherigen Schrittes zur Berechnung der neuen Approximation. Daher sind diese Methoden sequentiell und nicht trivial parallelisierbar. Im nächsten Abschnitt wird ein Überblick über die Erforschung neuer, zeitparalleler Verfahren gegeben.

2.2.2 Übersicht über zeitparallele Verfahren

Hochleistungsrechner werden mit der Zeit immer schneller. Die Website „top500.org“ (Prometheus, o.D.) vergleicht seit 1993 die schnellsten Supercomputer auf der Welt und fasst diese jedes Jahr in einer Liste zusammen. Im Juni 1993 war der Computer CM-5/1024 im Los Alamos National Laboratory, USA, die rechenstärkste Maschine mit maximal erreichten $59,7 \cdot 10^9$ Gleitkommaoperationen pro Sekunde (FLOPS) auf 1024 Rechenkern. 17 Jahre später, im Juni 2020, haben sich die Rechenleistungen deutlich verbessert. Der Supercomputer Fugaku im RIKEN Center for Computational Science, Japan, hat eine maximal gemessene Leistung von $415,530 \cdot 10^{12}$ auf 7.299.072 Rechenkern. Werden diese Zahlen näher untersucht fällt folgendes auf: Die FLOPS pro Kern sind um etwas weniger als das tausendfache gewachsen; die Anzahl der Rechenkern ist jedoch um mehr als das 7000-fache angestiegen. Dieser Trend wird sich weiter fortsetzen, da die Leistungssteigerung der Rechenkern an physikalische Grenzen stößt, aber die Anzahl weiter erhöht werden kann.

Dies zeigt, dass numerische Verfahren, welche Parallelität ausnutzen können, immer wichtiger werden. Es gibt viele effiziente Verfahren, welche parallel im Ort sind. Das Lösen eines Problems kann im Regelfall durch paralleles Rechnen im Ort beschleunigt werden, aber es lohnt nicht, das Gebiet zu klein einzuteilen, da dann zu viel Kommunikation zwischen den Rechenkern benötigt wird. Für zeitabhängige Probleme können mehr Rechenkern trotzdem nützlich sein, wenn auch eine Parallelisierung in der Zeit möglich ist. Aus der traditionellen Perspektive ist die Zeitdimension sequentiell; Zeitintegratoren benötigen Informationen des vorhergehenden Zeitschrittes, um den neuen zu berechnen. Mittlerweile gibt es jedoch viele verschiedene Verfahren, um die Parallelisierung der Zeitdimension zu ermöglichen. In diesem Abschnitt wird eine Übersicht über die Entwicklung und den derzeitigen Stand zeitparalleler Methoden gegeben mit Fokus auf zeitparallele Mehrgitterverfahren. M. Gander (2015) gibt einen schönen Überblick, worauf dieses Kapitel basiert.

Die Anfänge zeitparalleler Methoden gehen auf Nievergelt (1964) zurück. Dieser präsentierte 1964 ein Verfahren, welches für die parallele Lösung zeitabhängiger Probleme eine Unterteilung der Zeitdimension ausnutzt. In seiner Veröffentlichung trifft er einige visionäre Aussagen. So fiel ihm schon damals die Wichtigkeit des parallelen Rechnens auf:

Today, as it appears that one is getting closer to the maximal speed of electronic components, emphasis is put on allowing operations to be performed in parallel.

Außerdem beschreibt er, dass seine Methode nur ein Prototyp ist und andere, verbesserte Methoden darauf basieren werden.

Nievergelts Verfahren ist eine parallele numerische Integration gewöhnlicher Differentialgleichungen, welche zum Beispiel durch die Linienmethode zur Zeitdiskretisierung partieller Differentialgleichungen entstehen. Das Zeitintervall $[a, b]$ wird dafür in mehrere Subintervalle eingeteilt. Auf jedem der resultierenden Intervalle wird ein Anfangswertproblem gelöst; dies kann parallel durchgeführt werden. Schlussendlich wird die gesamte Lösung mit Hilfe von linearer Interpolation zusammengesetzt. Für lineare Probleme produziert die Methode eine exakte Lösung; im nichtlinearen Fall injiziert die Interpolation einen Fehler. Das Verfahren ist ein direkter Löser.

Seit Nievergelt's Methode entstanden vier verschiedene Forschungsrichtungen für zeitparallele Verfahren: Mehrfachschießverfahren, Raum-Zeit Gebietszerlegung, Mehrgitterverfahren sowie direkte Löser. Die ersteren drei Klassen sind iterative Löser; sie unterscheiden sich hauptsächlich durch verschiedene Aufteilung des Raum-Zeit-Gebiets.

Mehrfachschießverfahren

Schießverfahren wurden ursprünglich zur Lösung von Randwertproblemen entwickelt (Keller, 1992). Diese behandeln das zu lösende Problem statt als Randwertproblem der Form

$$u'' = f, \quad u(0) = a, \quad u(1) = b$$

als ein Anfangswertproblem:

$$u'' = f, \quad u(0) = a, \quad u'(0) = s$$

Das Ziel ist die Bestimmung des sogenannten Schuss-Parameters s , sodass die Lösung durch das Ziel $u(1) = b$ geht. Der Name der Methode stammt von der Vorstellung, dass damit die Ausrichtung s einer Kanone berechnet wird um mit dem Schuss das Ziel zu treffen. Im Regelfall ist dafür die Lösung einer nichtlinearen Gleichung nötig; dies kann zum Beispiel mit der Newton Methode erreicht werden.

Zeitabhängige Probleme sind jedoch Anfangswertprobleme wie zum Beispiel das folgende n -dimensionale Differentialgleichungssystem mit f stetig differenzierbar auf $[0, 1] \times \mathbb{R}^n$:

$$u'(t) = f(t, u(t)), \quad u(0) = a, \quad t \in [0, 1], a \in \mathbb{R}^n$$

Das Ziel des Schusses ist daher nicht bekannt. Mehrfachschießverfahren umgehen dieses Hindernis durch Einteilung des Zeitintervalls in mehrere Subintervalle. Auf jedem Subintervall kann nun ein Anfangswertproblem definiert werden; der Anfangswert, welcher nur für das erste Subintervall

bekannt ist, stellt dabei den Schuss-Parameter dar. In diesem Beispiel wird das Zeitintervall gedrittelt; daraus ergeben sich folgende Anfangswertprobleme:

$$\begin{aligned} u'_0(t) &= f(t, u_0(t)), & u_0(0) &= U_0 \\ u'_1(t) &= f(t, u_1(t)), & u_1\left(\frac{1}{3}\right) &= U_1 \\ u'_2(t) &= f(t, u_2(t)), & u_2\left(\frac{2}{3}\right) &= U_2 \end{aligned}$$

Zusätzlich müssen folgende Bedingungen erfüllt sein; $u(t, a)$ ist definiert als die Lösung von u an Zeitpunkt t mit Anfangswert a :

$$U_0 = a, \quad U_1 = u_0\left(\frac{1}{3}, U_0\right), \quad U_2 = u_1\left(\frac{2}{3}, U_1\right)$$

Zur Bestimmung der Anfangswerte führt dies zu einem nichtlinearem Gleichungssystem.

$$F((U_0, U_1, U_2)^T) := \begin{pmatrix} U_0 - a \\ U_1 - u_0\left(\frac{1}{3}, U_0\right) \\ U_2 - u_1\left(\frac{2}{3}, U_1\right) \end{pmatrix} = 0$$

Die Anwendung der Newton Methode und wenige Umformungen führen letztendlich zu folgender Rechenvorschrift, hier allgemein für N Subintervalle und der $k + 1$ Newton Iteration notiert:

$$U_{0,(k+1)} = a, \tag{2.10}$$

$$U_{i,(k+1)} = u_{i-1}(t_i, U_{i-1,(k)}) + \frac{\partial u_{i-1}}{\partial U_{i-1}}(t_i, U_{i-1,(k)})(U_{i-1,(k+1)} - U_{i-1,(k)}), \quad i = 1, 2, \dots, N \tag{2.11}$$

Wird dieses Verfahren näher betrachtet werden die Ähnlichkeiten zu Nievergelts Verfahren offensichtlich. Statt der Interpolation zur Verknüpfung der Lösung liegt hier eine iterative Methode vor. Bellen und Zennaro (1989) waren die ersten, die auf dieser Weise Nievergelts Methode weiterentwickelt haben. Sie verwendeten jedoch anstatt der Newton Methode die sogenannte Steffensen Methode, eine Variante der Newton Methode. 1993 veröffentlichten Chartier und Philippe (1993) eine formalisierte und allgemeinere Methode, welche dem oben beschriebenen Verfahren entspricht. Darauf aufbauend publizierten Saha et al. (1997) ihr zeitparalleles Verfahren zur Berechnung des Sonnensystems über einen sehr langen Zeitraum. Ihre Methode benutzt eine Approximation der Ableitung in Gleichung (2.10) und die Autoren zeigen mögliche verbesserte Laufzeiten für die Simulation des Sonnensystems.

In 2001 führten Lions et al. (2001) den Parareal Algorithmus in einem sehr kurzen Paper ein. Das Verfahren ist eine der am meist untersuchten zeitparallelen Methoden. Die Idee ist das parallele, iterative Lösen des Problems auf Subintervallen in der Zeit auf der Basis eines Prädiktor-Korrektor Ansatzes. Die ursprüngliche Idee stammt zwar von Ansätzen aus der Raum-Gebietszerlegung, doch die Methode kann als Mehrfachschießverfahren eingestuft werden. Die Autoren zitieren in ihrer Veröffentlichung die Arbeit von Chartier und Philippe (1993). In Abschnitt 2.2.3 wird näher auf das Parareal Verfahren eingegangen.

Es existieren viele Paper zu Untersuchungen der Parareal Methode auf verschiedene Anwendungen. Außerdem wurden viele Varianten und Weiterentwicklungen veröffentlicht. PITA (parallel implicit time-integrator) wurde 2003 von Farhat und Chandesris (2003) entwickelt und unterscheidet sich nur ein wenig in der Herangehensweise der Korrektur. Bekannte Verfahren, die auf dem Mehrgitter Prinzip basieren, wie PFASST und auch MGRIT (siehe Mehrgitterverfahren), stammen von Parareal ab.

Raum-Zeit Gebietszerlegung

Die Verfahren dieser Klasse zerlegen das Raum-Zeit Gebiet in Teilgebiete, die jeweils die gesamte Zeitspanne beinhalten. In gewisser Weise sind auch diese Methoden zeitparallel, da sie die Lösung auf dem lokalen Teilgebiet über die gesamte Zeitspanne $[a, b]$ bestimmen ohne die exakten Werte der Schnittstelle benutzen zu können. In der derzeitigen Forschung zeitparalleler Methoden steht vor allem die Kombination dieser Verfahren mit anderen zeitparallelen Ansätzen im Fokus.

Die Verfahren dieser Klasse gehen auf Picard (1893) und Lindelöf (1894) Ende des 19. Jahrhunderts zurück. Picard entwickelte ein iteratives Verfahren um zu beweisen, dass Lösungen bestimmter gewöhnlicher Differentialgleichungen existieren. Mit diesem Verfahren wird das Anfangswertproblem $u' = f(u, t)$ mit Anfangswert $u_{(0)}$ mit Hilfe einer Quadratur zu einer Folge von Probleme gewandelt, welche einfacher zu lösen sind:

$$u_{(k)}(t) = u_{(0)} + \int_0^t f(\tau, u_{(k-1)}(\tau))d\tau \quad (2.12)$$

Lindelöf bewies die superlineare Konvergenzrate der sogenannten Picard-Iteration.

Das Verfahren ist jedoch sehr ineffizient. 1982 veröffentlichten Lelarsmee et al. (1982) die Wellenform-Relaxation, welche eine Variante der Picard Iteration ist. Diese Methode wurde ursprünglich entwickelt zur Simulation von Schaltkreisen, deren Systemumfang immer größer und komplexer wurde. Das Verfahren basiert auf der Unterteilung eines großen Schaltkreises in kleinere Schaltkreise, welche unabhängig voneinander berechnet werden können. Eine Iteration verbessert die gesamte Lösung unter bestimmten Voraussetzungen an Übertragungen zwischen den Schaltkreisen. Der Name des Verfahrens stammt daher, dass Signale entlang Kabel „waveforms“ genannt werden.

M. J. Gander (1996) übertrug die Idee der Wellenform-Relaxation auf die Lösung partieller Differentialgleichungen. Anstatt eines Schaltkreises wird das gesamte Raum-Zeit Gebiet in überlappende Teilgebiete zerlegt und analog zur Wellenform-Relaxation iterativ gelöst. Diese Verfahren werden Schwarz Wellenform-Relaxation genannt, da die Teilgebiete überlappen wie in der Schwarz Methode. Durch verbesserte Bedingungen an Übertragungen wurden die Verfahren verbessert (Bennequin et al., 2009); sie sind anwendbar auf lineare und nichtlineare Probleme.

Alternative Varianten der Wellenform-Relaxation wurden von Mandal (2012) und Kwok (2014) präsentiert. Außerdem existieren Verfahren, die die Ansätze des Mehrfachschießens und der Wellenform-Relaxation kombinieren. Ein Beispiel ist die Parareal Schwarz Wellenform-Relaxation Methode (M. Gander, Jiang et al., 2013).

Mehrgitterverfahren

Zeitparallele Mehrgitterverfahren betrachten das gesamte Raum-Zeit Gebiet gleichzeitig. Die Methoden dieser Klasse sind nicht natürlich zeitparallel wie die vorhergehenden gebietszerlegenden Algorithmen, doch einige Komponenten der Verfahren können zeitparallel ausgeführt werden.

Die erste bekannte Methode ist das Parabolische Mehrgitterverfahren von Hackbusch (1984). Die Methode wurde entwickelt zur Lösung parabolischer Anfangsrandwertprobleme. Im originalen Algorithmus wird das Gitter nur im Ort und nicht in der Zeit gröber und als Glätter wird ein Jacobi-Verfahren im Ort an mehreren Zeitabschnitten verwendet. Das parabolische Mehrgitterverfahren rechnet sehr effizient zeitparallel wie Horton (1992) zeigt. Dank des Jacobi Glätters benötigt das Verfahren jedoch sehr spezielle Diskretisierungen um zu konvergieren und hat deshalb nur begrenzte Anwendungsmöglichkeiten (Vandewalle und Horton, 1995).

Lubich und Ostermann (1987) präsentierten 1987 die Mehrgitter Wellenform-Relaxation. Genau wie stationäre Defektkorrekturverfahren glättet die Wellenform-Relaxation bei Anwendung den Fehler. Dies nutzen die beiden Autoren aus und zeigen, dass die Wellenform-Relaxation durch Grobgitterkorrektur beschleunigt werden kann. Daraus entstand das Verfahren, welches einen anderen Glätter als das parabolische Mehrgitterverfahren verwendet und nicht auf reguläre Gitter limitiert ist. Die Methode wurde auf viele verschiedene parabolische Probleme angewendet und getestet wie die Wärmeleitungsgleichung (Lubich und Ostermann, 1987) oder die inkompressible Navier-Stokes Gleichung (Oosterlee und Wesseling, 1993). Parallele Effizienz wurde näher untersucht (Vandewalle und Van de Velde, 1994; Vandewalle und Piessens, 1992); in einem neueren Paper wurde die Anwendung auf parabolische Randwertprobleme diskutiert (Van Lent und Vandewalle, 2005).

Sowohl das Parabolische Mehrgitterverfahren als auch die Mehrgitter Wellenform-Relaxation lassen die Zeitschrittweite unverändert in ihrer Gitterhierarchie. Die ersten, die Vergrößerung nicht nur im Ort sondern auch in der Zeit betrachteten, waren Horton und Vandewalle (1995). Die Autoren entwickelten das erste Raum-Zeit Mehrgitterverfahren, welche die Zeit einfach als eine weitere Dimension behandeln. Ihre Methode zeichnet sich durch eine adaptive Vergrößerungs-Strategie aus: Abhängig von der Anisotropie der Diskretisierung auf jedem Level wird entweder im Ort oder in der Zeit vergrößert. Horton und Vandewalle analysieren die Methode angewandt auf die Wärmeleitungsgleichung sehr detailliert. Alternative Raum-Zeit Mehrgitterverfahren wurden von De Sterck et al. (2004) und Weinzierl und Köppl (2012) veröffentlicht. All diese Verfahren sind jedoch für spezielle Probleme mit speziellen Diskretisierungen entwickelt worden und nur auf diese anwendbar.

Mittlerweile wurden zwei weitere zeitparallele Mehrgitterverfahren entwickelt, welche beide auf Parareal basieren. Diese sind aktuell die mit am meisten untersuchten Verfahren in der Forschungsgemeinschaft. Eines ist MGRIT, welches in dieser Arbeit die Methode der Wahl ist und daher in Kapitel 3 näher erläutert wird. Das andere ist das „Parallel Full Approximation Scheme in Space-Time“ (PFASST) von Emmett und M. L. Minion (2012). Dieses basiert auf dem „Spectral Deferred Correction“ (SDC) (Dutt et al., 2000) Zeitintegrator. SDC ist ein iteratives Verfahren, welches die Integralgleichung der Picard-Iteration (2.12) löst, um die Lösung eines Anfangswertproblems $u' = f(u, t)$ zu bestimmen. Wird Definition 2.1.1 des Fehlers in Gleichung (2.12) eingesetzt, liefert das folgende Gleichung:

$$e(t) = u(0) + \int_0^t f(\tau, v(\tau) + e(\tau))d\tau - v(t)$$

Das Residuum der Näherungslösung v lautet folgendermaßen:

$$r(t) = v(0) + \int_0^t f(\tau, v(\tau))d\tau - v(t) =: F(v(\tau)) \quad (2.13)$$

Zusammengesetzt resultiert folgender Zusammenhang:

$$e(t) = r(t) + \int_0^t f(\tau, v(\tau) + e(\tau)) - f(\tau, v(\tau))d\tau \quad (2.14)$$

$$\Leftrightarrow e'(t) = r'(t) + f(t, v(t) + e(t)) - f(t, v(t)) \quad (2.15)$$

Das SDC Verfahren unterteilt nun einen Zeitschritt $[t_{i-1}, t_i]$ in mehrere Teilzeitschritte. Der erste Schritt des Verfahrens ist das Bestimmen einer diskreten Lösung des Anfangswertproblems an den Zeitpunkten des unterteilten Zeitschrittes mit einem simplen Zeitschrittverfahren wie dem expliziten Euler Verfahren. Dadurch wird eine erste Näherungslösung berechnet. Diese wird durch jede folgende Iteration folgendermaßen verbessert. Mit einer Quadratur hoher Ordnung wird Gleichung (2.13) gelöst. Mit diesem Ergebnis wird durch Anwendung des expliziten Euler Verfahrens Gleichung (2.14) gelöst und dadurch der Korrekturterm bestimmt. Wird der Korrekturterm auf die Näherungslösung addiert, ergibt sich eine neue Approximation, für welche gezeigt wurde, dass sie eine Ordnung höher ist. Dies kann bis zur Ordnung der verwendeten Quadratur wiederholt werden. SDC ist demnach ein iteratives Verfahren, welches durch ein Zeitschrittverfahren niedriger Ordnung ein Runge-Kutta Verfahren mit Ordnung entsprechend der verwendeten Quadratur approximiert.

SDC ist jedoch sequentiell in der Zeit. Das Verfahren wird nacheinander auf Zeitabschnitte $[t_{i-1}, t_i]$ angewandt. M. Minion (2010) parallelisiert diesen Vorgang in der Zeit durch den Vorschlag, das Fertigstellen SDCs nicht abzuwarten, sondern die Näherungslösung einer früheren Iteration auf dem folgenden Zeitabschnitt als Anfangswert zu verwenden. Dadurch können die Zeitabschnitte unter Einbuße der Exaktheit parallel berechnet werden. PFASST verwendet die parallele SDC Methode als Relaxations Verfahren innerhalb des Full Approximation Scheme. Auf den gröberen Gittern wird eine gröbere Raum-Zeit-Diskretisierung verwendet, auf welcher die Ordnung der Zeitintegration im Regelfall um Eins sinkt.

PFASST wurde für viele verschiedene Anwendungen untersucht. Emmett und M. L. Minion (2012) und M. Minion et al. (2014) zeigen erhebliche Reduzierung der Laufzeit mit bis zu 24 Zeitprozessen für parabolische Gleichungen. Weitere vielversprechende Varianten lassen sich in den Veröffentlichungen von Emmett und M. Minion (2014) und Speck (2017) finden. Bolten et al. (2016) zeigen die Äquivalenz MGRITs mit PFASST im linearen Fall. PFASST ist ein sehr vielversprechendes Verfahren. Die Nachteile der Methode sind jedoch, dass kein eigenes, problemspezifisches Zeitschrittverfahren verwendet werden kann sowie die aufwendige Integrierung des Verfahrens in existierende Programme.

Direkte Löser

Neben den vorgestellten iterativen Verfahren wurden auch einige zeitparallele direkte Löser entwickelt. Miranker und Liniger erfanden 1967 das erste zeitparallele direkte Verfahren, welches mit Hilfe modifizierter Prädiktor-Korrektor Formeln mit zwei Prozessoren gleichzeitig die Lösung

berechnet (Miranker und Liniger, 1967). Einen Überblick über parallele Prädiktor-Korrektor Methoden sowie parallele Block Methoden, welche ähnlich vorgehen und von Shampine und Watts (1969) vorgestellt wurden, präsentiert Franklin (1978). All diese Löser sind im Regelfall nur für kleine Probleme geeignet.

Ein weiterer Ansatz ist die Anwendung von Methoden, welche für Randwertprobleme entwickelt worden, auf Anfangswertprobleme. Eine der ersten Einführungen dazu geben Axelsson und Verwer (1985). Die Idee ist die folgende: Wird die Differentialgleichung eines Anfangswertproblems mit Hilfe der Mittelpunktsregel diskretisiert, benötigt es neben dem Anfangswert noch einen weiteren Wert, da die explizite Mittelpunktsregel ein Zwei-Schritt-Verfahren ist. Normalerweise wird dazu mit Hilfe des impliziten Euler-Verfahrens der nächste Schritt nach dem Anfangswert berechnet. Wird das implizite Euler-Verfahren anstatt dessen auf den letzten Zeitpunkt angewendet, liegt eine Art Randwertproblem vor. Das resultierende System kann nun an allen Zeitpunkten gleichzeitig gelöst werden. Es muss jedoch beachtet werden, dass sich durch die Transformation zu einer Randwertmethode die Stabilitätseigenschaften komplett ändern. Daher muss das daraus resultierende Verfahren genauestens geprüft werden (Brugnano und Trigiante, 1996; Brugnano und Trigiante, 1998). Aus diesem Ansatz können sehr vielversprechende Lösungsmethoden hervorkommen, wie zum Beispiel Brugnano und Trigiante (1993) zeigt.

Womble (1990) veröffentlichte den „parallelen Zeitschritt“ Algorithmus, mit dessen Hilfe der Autor als erster Beschleunigungen des Rechengangs auf einem Computer mit 1024 Prozessoren zeigen konnte. Dadurch wurde die Interesse an der Forschung zeitparalleler Verfahren verstärkt. Die Methode des parallelen Zeitschritts ist keine direkte Methode; die iterativen Komponenten werden jedoch in anderer Weise als bei den restlichen iterativen Verfahren verwendet. Die Zeitschritte werden mit Hilfe eines Mehrgitterverfahrens gelöst; jedoch wird, wie bei der parallelen SDC Methode, nicht auf die Konvergenz des Löser des vorhergehenden Zeitschrittes gewartet, sondern es wird die aktuelle (nicht konvergierte) Approximationslösung als Anfangswert des nächsten Schrittes benutzt. Dadurch können mehrere Zeitschritte parallel berechnet werden. DESHPANDE et al. (1995) zeigen jedoch, dass das Verfahren nur limitierte Laufzeitreduktion erreichen kann. Saltz und Naik (1988) präsentieren ein sehr ähnliches Verfahren, welches anstatt des Mehrgitterverfahrens die Block Jacobi-Methode verwendet.

Worley (1991) betrachtet zyklische Reduktionverfahren und deren Potential für Zeitparallelität. Sowohl die Reduktion als auch die Rücksubstituierung können parallel vollzogen werden, da jede Menge an zwei Gleichungen unabhängig von der anderen ist. Durch paralleles Vorgehen ist es sogar möglich, die Komplexität des Verfahrens auf $\log n$ mit n als Problemgröße zu verringern. Die genauere Theorie sowie die Kombination mit Wellenform-Relaxation kann in dem Paper von Horton, Vandewalle und Worley (1995) nachgelesen werden. Mit Hilfe der schnellen Fourier-Transformation und der Wellenform-Relaxation sowie als Grundlage ein zyklisches Reduktionsverfahren entwickelten Simoens und Vandewalle (2000) einen optimalen zeitparallelen Algorithmus.

Mit Hilfe der Laplace-Transformation des Problems entwickelten Sheen et al. (2000) einen zeitparallelen direkten Löser. Zum Lösen wird die inverse Laplace-Transformation benötigt, deren Integral mit einer bestimmten Quadratur zeitparallel berechnet werden kann. Das Verfahren ist limitiert auf Probleme, die die Laplace-Transformation ermöglichen. Ist die Methode jedoch anwendbar, gibt es vielversprechende Ergebnisse (Douglas et al., 2011; Sheen et al., 2003; Thomée, 2005).

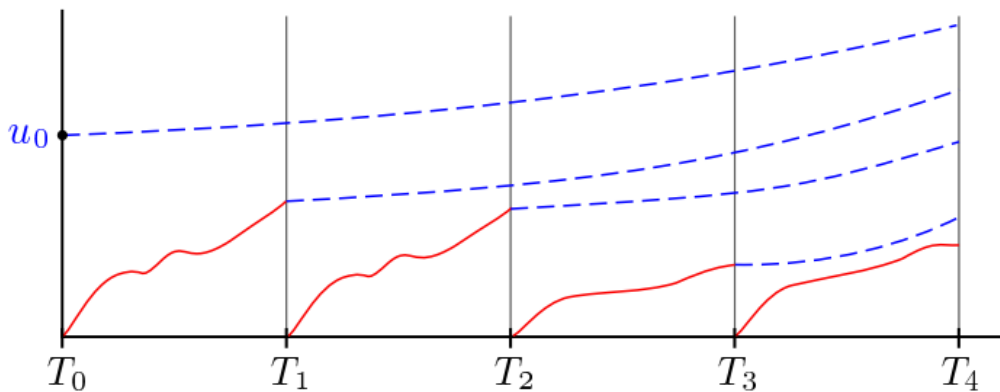


Abbildung 2.6: Überlappende Aufteilung des Zeitintervalls der ParaExp Methode. Rote Linien stellen inhomogene Probleme dar, blaue hingegen homogene. Bild übernommen (M. Gander und Güttel, 2013).

Zuletzt wird auf das ParaExp Verfahren verwiesen, welches von M. Gander und Güttel (2013) veröffentlicht wurde. Die Methode ist vor allem für zeitparallele Integration hyperbolischer Probleme geeignet, für welche fast alle anderen Ansätze große Schwierigkeiten vorweisen. ParaExp benutzt eine stark überlappende Aufteilung des Zeitgebiets, siehe Abbildung 2.6, und nutzt aus, dass die Lösung homogener Anfangswertprobleme mit Hilfe einer Arnoldi Approximation der Exponentialfunktion der Matrix sehr viel schneller ermittelt werden kann als die Lösung eines inhomogenen Problems. ParaExp ist eine Zweischritt-Methode. Im ersten Schritt werden auf nicht überlappenden Subintervalle des Zeitintervalls inhomogene Probleme gelöst; diese entsprechen den roten Linien in Abbildung 2.6. Im zweiten Schritt werden homogene Probleme gelöst, welche durch die blau gestrichelten Linien dargestellt sind. Durch Addition der Lösungen wird die Lösung des Gesamtproblems bestimmt. Wichtig ist, dass das Lösen der homogenen Probleme sehr schnell und effizient abläuft. M. Gander und Güttel (2013) präsentieren vielversprechende Ergebnisse des Verfahrens angewendet auf die Wärmeleitungsgleichung als auch auf die Wellengleichung.

2.2.3 Das Parareal Verfahren

Das von Lions et al. (2001) entwickelte zeitparallele Verfahren Parareal bildet die Grundlage vieler aktueller Methoden wie PFASST aber auch MGRIT. Daher wird in diesem Abschnitt das Verfahren näher erläutert; außerdem werden verschiedene Interpretationen Parareals diskutiert. Als Quellen dienen die Dissertation von Friedhoff (2014) sowie das Paper von M. J. Gander und Vandewalle (2007).

Betrachtet wird ein System gewöhnlicher Differentialgleichungen, welches zum Beispiel durch Diskretisierung einer partiellen Differentialgleichung mit Hilfe der Linien-Methode auf dem Zeitintervall $[a, b]$ entsteht.

$$u'(t) = f(u(t)), \quad u(t_0) = u_0, \quad t \in [a, b] \quad (2.16)$$

Das Zeitintervall $[a, b]$ wird in N Subintervalle $[a = t_0, t_1], [t_1, t_2], \dots, [t_{N-1}, t_N = b]$ eingeteilt. Außerdem werden zwei verschiedene Zeitschrittverfahren benötigt, welche durch die Operatoren F und G gekennzeichnet werden. $G(t_2, t_1, u_1)$ berechnet eine grobe Approximation der Lösung an Zeitpunkt t_2 mit dem Anfangswert $u(t_1) = u_1$. Die Anwendung des Operators ist weniger aufwendig als die des Operators $F(t_2, t_1, u_1)$, welcher eine genauere Approximation von $u(t_2)$ liefert mit demselben Anfangswert $u(t_1) = u_1$. Typischerweise werden dafür Runge-Kutta-Verfahren verschiedener Ordnung oder das gleiche Verfahren mit verschiedener Zeitschrittweite verwendet.

Der Algorithmus berechnet zuerst sequentiell mit Hilfe des groben Zeitschrittverfahrens die Initialwerte $U_{i,(0)}$ für $i = 0, 1, \dots, N$ mit $U_{0,(0)} = u_0$. Daraufhin wird für $k = 0, 1, 2, \dots$ folgende Korrektur iterativ durchgeführt bis eine Konvergenzbedingung erfüllt ist:

$$U_{i+1,(k+1)} = G(t_{i+1}, t_i, U_{i,(k+1)}) + F(t_{i+1}, t_i, U_{i,(k)}) - G(t_{i+1}, t_i, U_{i,(k)}) \quad (2.17)$$

Wird Gleichung (2.17) näher betrachtet fällt folgendes auf. Konvergiert das Verfahren, gleichen sich die Terme mit dem Operator G aus und die Lösung erfüllt die Gleichung $U_{i+1} = F(t_{i+1}, t_i, U_n)$; das heißt, Parareal erreicht dieselbe Genauigkeit wie der feine Operator F . Durch die Iterationsvorschrift ist es möglich, die Rechenschritte des teureren Operators $F(t_{i+1}, t_i, U_{i,(k)})$ parallel durchzuführen, da keine Werte der aktuellen Iteration benötigt werden. Dies gilt jedoch nicht für den groben Operator. Dieser muss sequentiell rechnen, da er auf die Berechnung von $U_{i,(k+1)}$ warten muss.

Wird Gleichung (2.17) mit der Rechenvorschrift eines **Mehrfachschießverfahrens** verglichen, fallen die Ähnlichkeiten sofort auf. Die Iterationsvorschrift eines Mehrfachschießverfahrens sieht folgendermaßen aus (siehe Gleichung (2.10)):

$$U_{0,(k+1)} = a, \\ U_{i+1,(k+1)} = u_i(t_{i+1}, U_{i,(k)}) + \frac{\partial u_i}{\partial U_i}(t_{i+1}, U_{i,(k)})(U_{i,(k+1)} - U_{i,(k)}), \quad n = 0, 1, \dots, N - 1$$

Wird der feine Operator F der Parareal Methode als numerischen Methode zur Bestimmung des Terms $u_i(t_{i+1}, U_{i,(k)})$ verwendet, gilt $u_i(t_{i+1}, U_{i,(k)}) = F(t_{i+1}, t_i, U_{i,(k)})$. Es bleibt zu klären, wie der Term $\frac{\partial u_i}{\partial U_i}(t_{i+1}, U_{i,(k)})$ berechnet oder approximiert wird. Eine Möglichkeit ist die Approximation der Auswirkung des Terms auf die Differenz $(U_{i,(k+1)} - U_{i,(k)})$ mit Hilfe eines Finiten Differenzen Ansatzes:

$$\frac{\partial u_i}{\partial U_i}(t_{i+1}, U_{i,(k)})(U_{i,(k+1)} - U_{i,(k)}) = G(t_{i+1}, t_i, U_{i,(k+1)}) - G(t_{i+1}, t_i, U_{i,(k)})$$

Wird all dies zusammengesetzt stellt sich heraus, dass Parareal ein Mehrfachschießverfahren ist.

Eine weitere mögliche Interpretation des Parareal Algorithmus ist, dass das Verfahren ein **Zwei-Level Mehrgitter Reduktionsverfahren** ist. Wie in Abschnitt 2.2.1 beschrieben, kann das System gewöhnlicher Differentialgleichungen in Gleichung (2.16) mit Hilfe von Einschrittverfahren gelöst werden. Unter den Annahmen, dass das verwendete Zeitgitter $\mathcal{T}^{(\delta t)}$ mit Gitterweite δt äquidistant ist, das Einschrittverfahren zeitunabhängig ist und die Funktion g linear ist, kann ein Einschrittverfahren umgeschrieben werden zu folgenden N -dimensionalen Gleichungssystem:

$$Au \equiv \begin{pmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{pmatrix} \equiv f \quad (2.18)$$

Alle lösungsunabhängigen Terme sind in $f_i^{(\delta T)}$ gesammelt. Dies kann umgeformt, mit Null addiert und in Matrixschreibweise umgeschrieben werden:

$$\begin{aligned} -\Phi^{(\delta T)} u_{i,(k+1)}^{(\delta T)} + u_{i+1,(k+1)}^{(\delta T)} &= \Phi^m u_{i,(k)}^{(\delta T)} - u_{i,(k)}^{(\delta T)} - \Phi^{(\delta T)} u_{i,(k)}^{(\delta T)} + u_{i,(k)}^{(\delta T)} + f_i^{(\delta T)} \\ \Leftrightarrow B^{(\delta T)} u_{(k+1)}^{(\delta T)} &= -A^{(\delta T)} u_{(k)}^{(\delta T)} + B^{(\delta T)} u_{(k)}^{(\delta T)} + f^{(\delta T)} \\ \Leftrightarrow u_{(k+1)}^{(\delta T)} &= u_{(k)}^{(\delta T)} + (B^{(\delta T)})^{-1} (f^{(\delta T)} - A^{(\delta T)} u_{(k)}^{(\delta T)}) \end{aligned}$$

Das Ergebnis ist ein einfaches iteratives Korrekturverfahren mit dem Residuum. Die Anwendung von $A^{(\delta T)}$ kann parallel berechnet werden, während die Anwendung von $(B^{(\delta T)})^{-1}$ sequentiell ist.

Damit wurde der Parareal Algorithmus mit einer alternativen Notation und Vorgehensweise beschrieben. Nun bleibt zu klären, weshalb Parareal als Zwei-Level Mehrgitter Reduktionsverfahren interpretiert werden kann. Wie in Abschnitt 2.1.4 wird das zu lösende System in C- und F-Punkte eingeteilt; C-Punkte entsprechen der Menge der Zeitpunkte des groben Zeitgitters $\mathcal{T}^{(\delta T)}$ und F-Punkte alle anderen Punkte auf dem feinen Zeitgitter $\mathcal{T}^{(\delta t)}$. Betrachtet wird die Matrix $A \in \mathbb{R}^{N \times N}$ des feinen Gitters, welche so umgeordnet werden kann, dass die zu den F-Punkten gehörenden Zeilen über den restlichen Zeilen angeordnet sind, siehe Abschnitt 2.1.4. Diese kann in die bekannte LDU-Zerlegung zerlegt werden, solange A invertierbar mit invertierbarer Submatrix A_{FF} ist. Sei $N^{(F)} = N - N^{(\delta T)}$ die Anzahl der F-Punkte, dann sind die Blockmatrizen $A_{FF} \in \mathbb{R}^{N^{(\delta T)} \times N^{(\delta T)}}$, $A_{CC} \in \mathbb{R}^{N^{(\delta T)} \times N^{(\delta T)}}$, $A_{CF} \in \mathbb{R}^{N^{(\delta T)} \times N^{(F)}}$, $A_{FC} \in \mathbb{R}^{N^{(F)} \times N^{(\delta T)}}$, $I_F \in \mathbb{R}^{N^{(F)} \times N^{(F)}}$ und $I_C \in \mathbb{R}^{N^{(\delta T)} \times N^{(\delta T)}}$; I_F und I_C sind Einheitsmatrizen. Es gilt:

$$A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} = \begin{pmatrix} I_F & 0 \\ A_{CF} A_{FF}^{-1} & I_C \end{pmatrix} \begin{pmatrix} A_{FF} & 0 \\ 0 & A_{CC} - A_{CF} A_{FF}^{-1} A_{FC} \end{pmatrix} \begin{pmatrix} I_F & A_{FF}^{-1} A_{FC} \\ 0 & I_C \end{pmatrix} \quad (2.21)$$

$S := A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}$ entspricht dem Schur-Komplement von A_{FF} in A . Ein Mehrgitterverfahren benötigt einen Restriktionsoperator R sowie einen Prolongationsoperator P . Daher werden zusätzlich folgende Operatoren definiert:

$$R := \begin{pmatrix} -A_{CF} A_{FF}^{-1} & I_C \end{pmatrix}, \quad P := \begin{pmatrix} -A_{FF}^{-1} A_{FC} \\ I_C \end{pmatrix}, \quad E := \begin{pmatrix} I_F \\ 0 \end{pmatrix}$$

R und P entsprechen der idealen Restriktion beziehungsweise Interpolation, das heißt, dass die Anwendung der Operatoren keinen Fehler einführt. Die Einträge der mittleren Matrix aus Gleichung (2.21) können nun als $A_{FF} = E^T A E$ und $S = R A P$ geschrieben werden. Damit gilt aus Gleichung (2.21) hergeleitet:

$$A^{-1} = P(R A P)^{-1} R + E(E^T A E)^{-1} E^T \quad (2.22)$$

Durch Einsetzen der Definitionen folgt außerdem $R A E = 0$. All dies zusammen liefert letztendlich folgende multiplikative Identität:

$$0 = I - A^{-1} A = I - P(R A P)^{-1} R A - E(E^T A E)^{-1} E^T A \quad (2.23a)$$

$$= (I - P(R A P)^{-1} R A) (I - E(E^T A E)^{-1} E^T A) \quad (2.23b)$$

Bei näherer Betrachtung definiert Gleichung (2.23b) den Fehlerfortpflanzungsoperator eines exakten Zwei-Level Mehrgitterverfahren. Der erste Faktor beschreibt die Fehlerfortpflanzung der Grobgitterkorrektur an den C-Punkten, welche durch einen idealen Petrov-Galerkin Operator definiert ist, dem

Schur-Komplement. Der zweite Faktor stellt die Fehlerfortpflanzung der Verbreitung der Korrektur auf dem gesamten feinen Zeitgitter dar, welche der Anwendung des Einschrittverfahrens ausgehend von den C-Punkten auf die entsprechenden F-Punkte entspricht. Mehrgitter Reduktionsverfahren sind iterative Methoden, da sie entweder eine Approximation des idealen Grobgitteroperators, der idealen Interpolation oder beider verwenden.

Wird das exakte Zwei-Level Mehrgitterverfahren mit Parareal verglichen, fallen die Ähnlichkeiten auf. Der Grobgitteroperator $A^{(\delta T)}$ ist exakt und äquivalent zum Schur-Komplement, sodass $A^{(\delta T)} = RAP$ gilt. Außerdem sind die Restriktionsoperatoren R und R_ϕ äquivalent. Um zu zeigen, dass Parareal ein Mehrgitter Reduktionsverfahren ist, muss der Fehlerfortpflanzungsoperator der Methode untersucht werden. Wird Abschnitt 2.2.3 betrachtet, kann dieser mit Hilfe der Injektion $R_I := \begin{pmatrix} 0 & I_C \end{pmatrix}$ folgendermaßen definiert werden:

$$P(I - (B^{(\delta T)})^{-1}A^{(\delta T)})R_I \quad (2.24)$$

Wird nun die Petrov-Galerkin Bedingung $A^{(\delta T)} = RAP$ eingesetzt ergibt dies $(I - P(B^{(\delta T)})^{-1}RA)PR_I$. Durch Einsetzen der Definitionen der Äquivalenz $A_{FF} = E^TAE$ gilt $(I - E(E^TAE)^{-1}E^TA) = PR_I$. Damit kann der Fehlerfortpflanzungsoperator der Parareal Methode folgendermaßen geschrieben werden:

$$(I - P(B^{(\delta T)})^{-1}RA)(I - E(E^TAE)^{-1}E^TA) \quad (2.25)$$

Dies zeigt, dass der ideale Grobgitteroperator $A^{(\delta T)}$ des exakten Zwei-Level Mehrgitterverfahren durch den Operator $B^{(\delta T)}$ approximiert wird. Dadurch wurde bewiesen, dass Parareal als ein Zwei-Level Mehrgitter Reduktionsverfahren interpretiert werden kann. Mit Hilfe des Full Approximation Scheme kann dies außerdem auf den nichtlinearen Fall erweitert werden. M. J. Gander und Vandewalle (2007) beweisen, dass Parareal als ein Zwei-Level Mehrgitterverfahren interpretiert werden kann. Dazu werden etwas andere Komponenten beziehungsweise Operatoren gewählt, jedoch kann dort die Einbindung des Full Approximation Scheme nachgelesen werden.

2.3 Die Simulation des neuromuskulären Systems

Das Ziel des DiHu-Projekts der Universität Stuttgart ist die Simulation des neuromuskulären Systems mit Hilfe von Hochleistungsrechnern, um den menschlichen Körper besser verstehen zu können und damit zur Verbesserung der personalisierten Gesundheitsvorsorge beizutragen. Hierzu werden elektromyographische Signale simuliert. Die Daten können von Ärzten einfach erhoben werden und dadurch ist es möglich, die experimentellen Daten mit den simulierten zu vergleichen und die Modellgüte zu bestimmen. Innerhalb des Projektes wird Wert darauf gelegt, das mathematische Modell weiter zu verbessern und eine realistische Anzahl an Muskelfasern und motorischen Einheiten zu simulieren. Dafür sollen effiziente numerische Methoden verwendet werden, die nicht nur stabil sondern auch hochparallel auf Hochleistungsrechnern rechnen können. Weitere Informationen zu diesem Projekt gibt es auf der Projektwebsite (O. et al., 2020), sowie in dem dazu veröffentlichten Beitrag (Maier et al., 2019).

2.3.1 Das OpenDiHu-Simulationspaket

Das Software Framework des Digital Human Projekts ist OpenDiHu genannt und kann auf Github gefunden werden (Maier, o.D.). Der Hauptteil des Codes ist in C++ implementiert, sodass mit Hilfe von MPI und der PETSc Bibliothek sämtliche Simulationen effizient parallel im Ort berechnet werden können. Außerdem werden geschickte Datenstrukturen verwendet, welche Vektorisierung ermöglichen und unnötiges Kopieren von Daten vermeiden. Durch Paraview kann ein Muskel und die berechneten elektromyographischen Signale visualisiert werden. Abbildung 2.8 zeigt ein Beispiel davon.

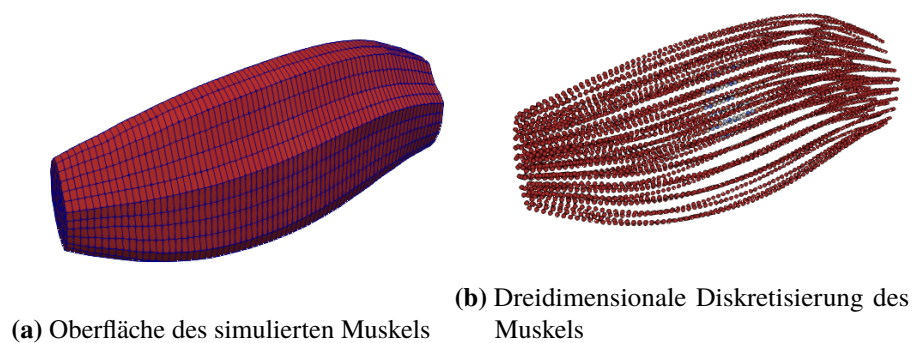


Abbildung 2.8: Beispielsausgabe des Lösungsverfahrens des Multidomain Modells. Dargestellt ist ein dreidimensionaler Muskel mit 49 Muskelfasern und 74 Punkten pro Faser.

Innerhalb der Framework ist das Lösen von statischen und dynamischen Multiphysik Problemen möglich. Dabei sind Diskretisierungen auf ein-, zwei- und dreidimensionalen Gittern realisierbar. Mit der Hilfe eines Python3-Interpreters können alle erwünschten Einstellungen zu Gittern, Lösern und Visualisierung beliebig gesetzt werden.

In dieser Arbeit werden zwei Gleichungen OpenDiHus näher betrachtet. Die erste Gleichung ist die Wärmeleitungsgleichung, die eine der einfachsten zeitabhängigen partiellen Differentialgleichungen ist. Die Anwendung des MGRIT Verfahrens auf dieses Problem wurde schon mehrmals untersucht und dabei wurden vielversprechende Ergebnisse erzielt. Die Betrachtung dieser Gleichung liefert eine Vergleichsbasis zu den Resultaten, die aus der Untersuchung des zweiten Problems gewonnen werden. Im zweiten Fall wird das MGRIT Verfahren auf ein Multidomain Problem angewandt. Dieses ist ein System partieller Differentialgleichungen, welches das neuromuskuläre System beschreibt.

2.3.2 Referenzproblem: Diffusionsgleichung

Sei $\Omega \subset \mathbb{R}^1$ ein beschränktes Gebiet mit Rand Γ und Zeitpunkt $T > 0$. Betrachtet wird die isotropische Diffusionsgleichung, welche eine parabolische partielle Differentialgleichung ist, mit homogener Dirichlet-Randbedingung und Anfangswerten $z : \bar{\Omega} \rightarrow \mathbb{R}$. Gesucht wird $u : \bar{\Omega} \times]0, T[\rightarrow \mathbb{R}$ mit $\Delta u(x, t) = \frac{\partial^2 u(x, t)}{\partial x^2}$:

$$\frac{\partial u(x, t)}{\partial t} - c \cdot \Delta u(x, t) = 0, \quad (x, t) \in \Omega \times]0, T[, c \in \mathbb{R} \quad (2.26a)$$

$$u(x, t) = 0, \quad (x, t) \in \Gamma \times]0, T[\quad (2.26b)$$

$$u(x, t) = z(x), \quad t = 0, x \in \bar{\Omega} \quad (2.26c)$$

Oft wird dieses Problem auch als Wärmeleitungsproblem bezeichnet. In dieser Arbeit wird der Parameter $c = 5$ gewählt und in den folgenden Überlegungen aus Übersichtlichkeitsgründen weggelassen.

Um die Lösung der Wärmeleitungs-Anfangs-Randwertaufgabe zu berechnen, muss diese sowohl im Ort als auch in der Zeit diskretisiert werden. Eine Möglichkeit ist die Rothe-Methode. Dazu wird das Problem zunächst in der Zeit und dann im Ort diskretisiert. Das Zeitintervall wird in ein Zeitgitter mit $N + 1$ Zeitpunkten

$$0 = t_0 < t_1 < t_2 < \dots < t_N = T, \quad \delta t_i = t_i - t_{i-1}, \quad i = 1, 2, \dots, N$$

zerlegt. Auf jedem Zeitpunkt des Gitters ist die zeitdiskrete Funktion $(u_i)_{i=0}^N$ mit $u_i \in H_0^1(\Omega)$ definiert. Mit der Anwendung des allgemeinen semidiskreten Theta-Verfahren wird folgendes erhalten:

$$u_0 = h, \quad u_{i+1} - \delta t_{i+1} \theta \Delta u_{i+1} = u_i + \delta t_{i+1} (1 - \theta) \Delta u_i$$

Das Ergebnis sind folglich mehrere quasi-stationäre partielle Differentialgleichungen. Die Wahl des Parameters $\theta \in [0, 1]$ entscheidet, welches Zeitschrittverfahren verwendet wird. Für das Referenzproblem wird die implizite Euler-Methode verwendet, das bedeutet $\theta = 1$. Die Differentialgleichungen werden im nächsten Schritt im Ort diskretisiert.

Eine gängige Methode zur Diskretisierung im Ort ist die Finite Elemente Methode, welche detailliert in dem Buch von Zienkiewicz et al. (2013) beschrieben wird. Dazu wird die schwache Formulierung des semidiskreten Problems näher betrachtet. Für alle Testfunktionen $\varphi \in H_0^1(\Omega)$ soll gelten:

$$\begin{aligned} \langle u_0, \varphi \rangle_{L^2(\Omega)} &= \langle z, \varphi \rangle_{L^2(\Omega)} \\ \langle u_{i+1}, \varphi \rangle_{L^2(\Omega)} + \delta t_{i+1} \langle \nabla u_{i+1}, \nabla \varphi \rangle_{L^2(\Omega)} &= \langle u_i, \varphi \rangle_{L^2(\Omega)} \end{aligned}$$

Das Grundprinzip der Finiten Elemente Methode ist die Überdeckung des gesamten Gebiets $\bar{\Omega}$ durch einfache geometrische Elemente, ohne dass sich diese überschneiden. Im hier benutzten eindimensionalen Fall ist die einfachste Einteilung beziehungsweise Triangulierung des Gebiets die Unterteilung in äquidistante Strecken. Für höherdimensionale Fälle sind Formen wie Dreiecke und Rechtecke oder Würfel und Tetraeder möglich. Für jeden Zeitpunkt t_i kann eine Triangulierung $\Omega_i^{(h)}$ sowie ein endlich dimensionaler Ansatzraum $V_i^{(h)} \in H_0^1(\Omega)$ mit geeigneter Basis $\{\varphi_{j,i}^{(h)}\}_{j=0}^n$ gewählt werden. Der Einfachheit halber wird hier dieselbe Triangulierung und derselbe Ansatzraum für jeden Zeitpunkt gewählt, das heißt es gilt $\Omega_i^{(h)} = \Omega^{(h)}$ beziehungsweise $V_i^{(h)} = V^{(h)}$ für alle

$i = 0, 1, \dots, N$. Als Ansatz- und Testfunktionen werden spezielle Polynome verwendet, welche in Kombination mit den zu bestimmenden Koeffizienten $\bar{u}_i^{(h)}$ als Linearkombination die kontinuierliche Lösung approximieren. Es resultiert ein Galerkin-Problem für jeden Zeitschritt mit $u_i^{(h)} \in V^{(h)}$:

$$\langle u_0^{(h)}, \varphi^{(h)} \rangle_{L^2(\Omega)} = \langle z, \varphi^{(h)} \rangle_{L^2(\Omega)}, \quad \forall \varphi^{(h)} \in V^{(h)} \quad (2.28)$$

$$\langle u_{i+1}^{(h)}, \varphi^{(h)} \rangle_{L^2(\Omega)} + \delta t_{i+1} \langle \nabla u_{i+1}^{(h)}, \nabla \varphi^{(h)} \rangle_{L^2(\Omega)} = \langle u_i^{(h)}, \varphi^{(h)} \rangle_{L^2(\Omega)}, \quad \forall \varphi^{(h)} \in V^{(h)} \quad (2.29)$$

Die Lösung $u_i^{(h)}$ wird nun als Linearkombination aus den Basisfunktionen $\varphi^{(h)}$ des Ansatzraums $V^{(h)}$ mit Dimension n dargestellt.

$$u_i^{(h)}(x) = \sum_{j=1}^n (\bar{u}_i^{(h)})_j \varphi_{j,i}^{(h)}(x), \quad \bar{u}_i^{(h)} = ((\bar{u}_i^{(h)})_1, \dots, (\bar{u}_i^{(h)})_n)^T$$

Um alles anschaulich zusammensetzen zu können wird noch die Steifigkeitsmatrix

$$A^{(h)} = (a_{jl})_{j,l=1}^n \in \mathbb{R}^{n \times n}, \quad a_{jl} = \langle \nabla \varphi_l^{(h)}, \nabla \varphi_j^{(h)} \rangle_{L^2(\Omega_h)} \quad (2.30)$$

sowie die Massenmatrix

$$M^{(h)} = (m_{jl})_{j,l=1}^n \in \mathbb{R}^{n \times n}, \quad m_{jl} = \langle \varphi_l^{(h)}, \varphi_j^{(h)} \rangle_{L^2(\Omega_h)} \quad (2.31)$$

benötigt. Schlussendlich resultiert dies mit Hilfe der L^2 -Projektion $P^{(h)} : L^2(\Omega_h) \rightarrow V^{(h)}$ in die Berechnung der Anfangswertkoeffizienten

$$u_0^{(h)} = P^{(h)} z$$

und daraufhin in das in jedem Zeitschritt zu lösende lineare Gleichungssystem:

$$(M^{(h)} + \delta t_{i+1} A^{(h)}) \bar{u}_{i+1}^{(h)} = M^{(h)} \bar{u}_i^{(h)} \quad (2.32)$$

In dem hier behandelten eindimensionalen Fall werden als Triangulierung $\Omega^{(h)}$ äquidistante Strecken der Länge h verwendet. Durch die Wahl des Parameters h kann die Feinheit der Triangulierung bestimmt werden. Dadurch ist es möglich, verschieden feine Finite Elemente Gitter zu kreieren. Als Basisfunktionen $\varphi^{(h)}$ werden die global stetigen und lokal stückweise linearen Lagrange-Elemente der Dimension eins gewählt. Das lineare Gleichungssystem (2.32) wird mit Hilfe eines Krylow-Unterraum Verfahrens, dem GMRES Verfahren, bis zu der absoluten Toleranz $1e - 10$ gelöst. Die Anwendung der Linienmethode resultiert in demselben Gleichungssystem.

2.3.3 Das Modell des neuromuskulären Systems: das Multidomain Problem

Die Diffusionsgleichung ist das Standardbeispiel einer parabolische Differentialgleichung. Um das neuromuskuläre System des menschlichen Körpers zu modellieren wird jedoch ein System aus komplexeren Differentialgleichungen benötigt. Selbst alltägliche Aufgaben wie das Hochheben eines Gegenstandes benötigen das koordinierte Handeln vieler verschiedener Körperkomponenten. Ein wesentlicher Bestandteil solch einer Aktion ist die Kontraktion der Skelettmuskulatur ausgelöst durch das Nervensystem. Ein Muskel der Skelettmuskulatur besteht aus bis zu hunderten motorischen Einheiten. Diese bestehen aus einem Motoneuron und vielen Muskelfasern; zusammen sind sie

die kleinste individuell steuerbare Einheit des Muskels. Muskelfaserbündel sind untereinander verbunden durch die extrazelluläre Matrix. Für eine genaue Beschreibung des Aufbaus und der Funktion der Skelettmuskulatur wird auf das Buch von MacIntosh et al. (2006) verwiesen.

Sendet ein Motoneuron einen Impuls, das sogenannte Aktionspotential, an die zugehörigen Muskelfasern, so wird die motorische Einheit stimuliert. Elektromyographische Aufnahmen (Merletti und Parker, 2004) messen die elektrische Aktivität der Skelettmuskulatur und stellen eine Möglichkeit dar, das neuromuskuläre System zu untersuchen. Die aus diesen Messungen gewonnenen Resultate haben jedoch Limitationen. Beispiele sind die nicht invasiven Messungen der elektromyographischen Signale, welche ausschließlich das Aufzeichnen von Aktivitäten von äußeren Muskelkomponenten ermöglichen und dadurch die Identifizierungen verschiedener Phänomene wie der „Cross-Talk“ erschweren (Farina et al., 2016), sowie die schwierige Bestimmung, welche motorische Einheit welchen Einfluss auf die gemessenen elektromyographischen Signale genommen hat (Holobar et al., 2010). Simulationen der elektromyographischen Signale ermöglichen ein besseres Verständnis der neuromuskulären Systems und folglich verbesserte Messungen und Interpretationen real gemessener Signale.

Es gibt mehrere Modelle des neuromuskulären Systems zur Simulation elektromyographischer Signale. Der am weitest verbreitete Ansatz ist das Modellieren der Skelettmuskeln als konzentrierter Volumenleiter (Klotz et al., 2019). Einen Überblick über verschiedene solcher Modelle liefert Mesin (2013). Das Bidomain Modell (Miller et al., 1978) ist im Gegensatz dazu ein Ansatz aus der Kontinuumsmechanik zur Simulation elektrisch aktiver Gewebe. Es ist vor allem populär zur Simulation elektrischer Potentiale des Herzens (Pullan et al., 2005) und ermöglicht gegenüber den anderen Modellen eine detailliertere Beschreibung der elektrischen Aktivität. Das Bidomain Modell basiert auf der Annahme, dass an jedem Gewebepunkt sowohl ein intrazellulärer als auch ein extrazellulärer Raum homogen existiert. Beide Räume (Bidomain) sind elektrisch durch einen Ionenfluss über die Zellmembrane verbunden. Die Simulation elektromyographischer Signale mittels des Bidomain Modells wurde im Rahmen des Digital Human Projektes implementiert, durchgeführt und analysiert (Bradley et al., 2018).

Doch beide Ansätze der Modellierung haben entscheidende Nachteile. Modelle auf Basis der konzentrierten Volumenleiter müssen elektronische Potentialfelder innerhalb der Muskelfasern vor der Simulation berechnen und gehen daher von der Unabhängigkeit der Felder von dem elektrischen Zustand des extrazellulären Raumes aus. Dies führt zu Mängeln im Modell, welche noch nicht untersucht oder quantifiziert worden sind (Klotz et al., 2019). Das Bidomain Modell auf der anderen Seite geht nicht auf die Unterteilung der Skelettmuskulatur in motorische Einheiten ein. Klotz et al. (2019) präsentieren in ihrer Veröffentlichung eine Erweiterung des Bidomain Modells, das Multidomain Modell. Dieses ermöglicht die Übertragung der Simulation elektrischer Signale des Bidomain Modells auf mehrere motorische Einheiten (Multidomain). Eine genaue Herleitung und Beschreibung des Gleichungssystems, welches das Multidomain Modell beschreibt, kann in dem Paper von Klotz et al. (2019) nachgelesen werden. Auf diesem basiert dieses Kapitel.

Das Multidomain Modell

Damit die Gleichungen übersichtlich bleiben wird analog zur Vorgehensweise von Klotz et al. (2019) das Zeitargument individueller Funktionen, das heißt die Abhängigkeit von der Variable t , nicht notiert. An jedem Punkt $P \in \Omega_m$ innerhalb des Muskels Ω_m existieren ein extrazellulärer

(dargestellt durch Exponent $(\cdot)^{(e)}$) sowie M intrazelluläre (dargestellt durch Exponent $(\cdot)^{(i)}$) Räume. M entspricht der Anzahl an motorischen Einheiten; der Index $k \in M_{\text{me}} := \{1, 2, \dots, M\}$ stellt die entsprechende motorische Einheit dar. In jedem Raum existiert ein elektronisches Potential $\phi^{(e)}$ beziehungsweise $\phi_k^{(i)}$. Für jede motorische Einheit wird ein transmembranes Potential $V_k := \phi_k^{(i)} - \phi^{(e)}$ eingeführt. Die elektrischen Eigenschaften eines Raumes werden durch die Leitfähigkeits-Tensoren $\sigma^{(e)}$ beziehungsweise $\sigma_k^{(i)}$ beschrieben. F_k charakterisiert die Geometrie der Muskelfasern; es entspricht dem Verhältnis zwischen der Fläche der Hülle und dem Volumen der Faser. Der Parameter $f_k \in [0, 1]$ beschreibt die Mikrostruktur des Muskels; f_k stellt die Wahrscheinlichkeit dar, mit welcher ein beliebiger Punkt in der entsprechenden motorischen Einheit liegt. Um den Ionenstrom $I_k^{(\text{ion})}(y_k, V_k, I_k^{(\text{stim})})$ über die Muskelfasermembran zu modellieren wird der biophysikalische Ansatz von Hodgkin und Huxley (1952) verwendet. Diese führen eine nichtlineare, multidimensionale Gleichung $G_k(y_k, V_k, I_k^{(\text{stim})})$ ein, welche die Dynamik der Fasermembran beschreibt. Für den genauen Aufbau der Funktion und der subzellulären Zustandsvariable y_k wird auf die Veröffentlichung von Hodgkin und Huxley (1952) hingewiesen. Zusätzlich werden die Membrankapazität einer Muskelfaser C_k , welche konstant für eine motorische Einheit ist, sowie einen externen Stimulus $I_k^{(\text{stim})}$, welcher den Befehl des Nervensystems darstellt, benötigt. All dies zusammengesetzt führt zu dem folgenden Gleichungssystem:

$$0 = \text{div} \left(\left(\sigma^{(e)} + \sum_{k=1}^M f_k \sigma_k^{(i)} \right) \text{grad}(\phi^{(e)}) \right) + \sum_{k=1}^M f_k \text{div} \left(\sigma_k^{(i)} \text{grad}(V_k) \right), \quad \text{in } \Omega_m \quad (2.33a)$$

$$\frac{\partial V_k}{\partial t} = \frac{1}{F_k C_k} \text{div} \left(\sigma_k^{(i)} \text{grad}(V_k + \phi^{(e)}) \right) - \frac{1}{C_k} I_k^{(\text{ion})} \left(y_k, V_k, I_k^{(\text{stim})} \right), \quad \text{in } \Omega_m \quad (2.33b)$$

$$\frac{\partial y_k}{\partial t} = G_k \left(y_k, V_k, I_k^{(\text{stim})} \right), \quad \text{in } \Omega_m \quad (2.33c)$$

Gleichung (2.33a) und Gleichung (2.33b) bilden ein System bestehend aus $(M + 1)$ gekoppelten partiellen Differentialgleichungen, welches geschlossen wird durch das System gewöhnlicher Differentialgleichungen aus Gleichung (2.33c). Die erste Gleichung beschreibt die Erhaltung der elektrischen Ladung an jedem Punkt des Muskels; das heißt, der Stromfluss zwischen intra- und extrazellulären Räumen wird ausgeglichen. Das System partieller Differentialgleichungen der zweiten Gleichung in Kombination mit dem System gewöhnlicher Differentialgleichungen aus der dritten Gleichung beschreibt die Evolution des transmembranen Potentials entlang einer Muskelfaser der entsprechenden motorischen Einheit. Das Ziel ist die Bestimmung der extrazellulären elektronischen Potentiale $\phi^{(e)}$, welche den elektromyographischen Signalen entsprechen.

Für die Wahl $M = 1$ entspricht das Multidomain Modell dem Bidomain Modell. Daher sind die mathematischen Eigenschaften beider Modelle sehr ähnlich. Die Gleichungen (2.33b) sind Reaktionsdiffusionsgleichungen und damit parabolische partielle Differentialgleichungen. Der Reaktionsteil dieser Gleichungen sowie die gewöhnlichen Differentialgleichungen (2.33c) sind nichtlinear, unter anderem wegen des externen Stimulus. Der Diffusionsteil in Kombination mit Gleichung (2.33a), welche eine stationäre Gleichung ist, verhält sich diffusiv.

Zu guter Letzt müssen die Gleichungen mit Randbedingungen geschlossen werden. Klotz et al. (2019) modellieren dafür eine Fettschicht, das heißt ein elektrisch inaktives Gewebe, um das Muskelgewebe herum. Dies führt zu einer Reihe von Gleichungen, welche in dieser Arbeit nicht mit in Betracht gezogen werden. Anstatt dessen werden homogene Dirichlet-Randbedingungen sowie homogene Neumann-Randbedingungen verwendet. Diese stellen sicher, dass kein Strom aus dem simulierten Körperteil abfließen kann.

Lösungsverfahren

Dem Autor ist keine analytische Lösung solch eines Differentialgleichungssystems bekannt. Daher muss dieses mittels numerischer Verfahren gelöst werden. Für Reaktionsdiffusionsgleichungen wie Gleichung (2.33b) bietet sich ein Operator-Splitting Verfahren an. Dadurch kann der nichtlineare Reaktionsterm mit Hilfe eines expliziten Verfahrens und der Diffusionsterm mit einer impliziten Methode gelöst werden. Anstatt das von Klotz et al. (2019) vorgeschlagene Godunov-Splitting zu verwenden, wird analog zum Lösungsverfahren des Bidomain Modells (Bradley et al., 2018) die Strang-Splitting Methode angewendet. Diese ist in Kombination mit dem Heun Verfahren sowie der Crank-Nicolson Methode ein Zeitschrittverfahren zweiter Ordnung und reduziert dadurch den Rechenaufwand. Abbildung 2.9 visualisiert das Strang-Splitting Verfahren. Die nichtlinearen Terme werden mit Hilfe einer beliebigen Anzahl expliziter Heun Verfahren Anwendungen mit Zeitschrittweite $\delta t^{(H)}$ bis zum Zeitpunkt $t_* := (t_{i+1} - t_i)/2$ berechnet. Die Lösung wird an den impliziten Multidomain Löser übermittelte. Dieser berechnet die Lösung der diffusiven Terme mit Hilfe eines impliziten Zeitschrittverfahrens mit Zeitschrittweite $\delta t^{(MD)} = t_{i+1} - t_i$. Die ermittelten Resultate werden an das Heun Verfahren übergeben, welches diese als Anfangswerte am Zeitpunkt t_* verwendet und die nichtlinearen Terme bis zum Zeitpunkt t_{i+1} löst.

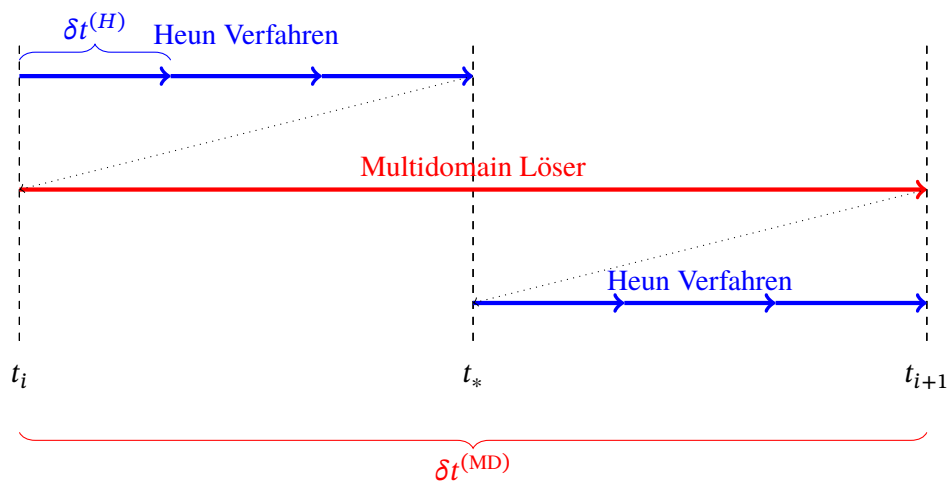


Abbildung 2.9: Ein Zeitschritt des Strang-Splitting Verfahrens angewandt auf das Multidomain Modell

Im folgenden werden die einzelnen Lösungsschritte genauer beschrieben. Ein Schritt des expliziten Heun Verfahren angewandt auf Gleichung (2.33c) und dem Reaktionsterm der Gleichung (2.33b) für alle $k \in M_{\text{me}}$ ist folgendermaßen gegeben, wobei der Übersichtlichkeit halber $\delta t^{(H)} = t_* - t_i$ gesetzt wird:

$$\begin{pmatrix} y_{k,p} \\ V_{k,p} \end{pmatrix} = \begin{pmatrix} y_{k,i} \\ V_{k,i} \end{pmatrix} + \delta t^{(H)} \begin{pmatrix} G_k(y_{k,i}, V_{k,i}, I_{k,i}^{(\text{stim})}) \\ -\frac{1}{C_k} I_k^{(\text{ion})}(y_{k,i}, V_{k,i}, I_{k,i}^{(\text{stim})}) \end{pmatrix},$$

$$\begin{pmatrix} y_{k,*} \\ V_{k,*} \end{pmatrix} = \begin{pmatrix} y_{k,i} \\ V_{k,i} \end{pmatrix} + \frac{\delta t^{(H)}}{2} \begin{pmatrix} G_k(y_{k,i}, V_{k,i}, I_{k,i}^{(\text{stim})}) + G_k(y_{k,p}, V_{k,p}, I_{k,p}^{(\text{stim})}) \\ -\frac{1}{C_k} (I_k^{(\text{ion})}(y_{k,i}, V_{k,i}, I_{k,i}^{(\text{stim})}) + I_k^{(\text{ion})}(y_{k,p}, V_{k,p}, I_{k,p}^{(\text{stim})})) \end{pmatrix}$$

Die Lösung der diffusiven Terme erfolgt durch den Multidomain Löser. Der diffusive Term der Reaktionsdiffusionsgleichung (2.33b) wird analog zum Vorgehen in Abschnitt 2.3.2 in der Zeit durch das semidiskrete Theta-Verfahren diskretisiert:

$$\begin{aligned} & V_{k,i+1} - \frac{\delta t^{(\text{MD})} \theta}{F_k C_k} \text{div}(\sigma_k^{(i)} \text{grad}(V_{k,i+1} + \phi^{(e)})) \\ &= V_{k,i} + \frac{\delta t^{(\text{MD})} (1 - \theta)}{F_k C_k} \text{div}(\sigma_k^{(i)} \text{grad}(V_{k,i} + \phi^{(e)})) \\ &\Leftrightarrow \frac{V_{k,i+1} - V_{k,i}}{\delta t^{(\text{MD})}} = \frac{1}{F_k C_k} (\theta \text{div}(\sigma_k^{(i)} \text{grad}(V_{k,i+1})) + (1 - \theta) \text{div}(\sigma_k^{(i)} \text{grad}(V_{k,i}))) \\ &\quad + \frac{1}{F_k C_k} (\theta \text{div}(\sigma_k^{(i)} \text{grad}(\phi_{i+1}^{(e)})) + (1 - \theta) \text{div}(\sigma_k^{(i)} \text{grad}(\phi_i^{(e)}))) \end{aligned}$$

Die resultierenden quasi-stationären Differentialgleichungen werden nun wie für das Referenzproblem im Ort mit Hilfe Finiter Elemente diskretisiert. Dazu wird die schwache Formulierung des Problems betrachtet; diskretisiert wird auf dem Gitter $\Omega^{(h)}$ mit einem endlich dimensionalen Ansatzraum mit geeigneter Basis $\{\varphi_j^{(h)}\}_{j=0}^n$. Die Zwischenschritte werden hier nicht notiert; sie erfolgen durch Einsetzen und simple Umformungen. Es wird angemerkt, dass durch die gewählten homogenen Dirichlet-Randbedingungen alle entstehenden Randintegrale wegfallen. Schlussendlich kann das Ergebnis in Matrixschreibweise folgendermaßen zusammengefasst werden. Um die Koeffizienten $\bar{V}_{k,i+1}^{(h)}$ und $\bar{\phi}_{i+1}^{(e),(h)}$ zu bestimmen, werden die Steifigkeitsmatrix (2.30) und die Massenmatrix (2.31) benötigt:

$$A_k = \begin{pmatrix} \frac{\theta \sigma_k^{(i)}}{F_k C_k} A^{(h)} - \frac{1}{\delta t^{(\text{MD})}} M^{(h)} & \frac{\theta \sigma_k^{(i)}}{F_k C_k} A^{(h)} \end{pmatrix} \quad (2.34)$$

$$B_k = \begin{pmatrix} \frac{(\theta-1) \sigma_k^{(i)}}{F_k C_k} A^{(h)} - \frac{1}{\delta t^{(\text{MD})}} M^{(h)} & (\theta-1) \sigma_k^{(i)} A^{(h)} \end{pmatrix} \quad (2.35)$$

$$\Rightarrow A_k \begin{pmatrix} \bar{V}_{k,i+1}^{(h)} \\ \bar{\phi}_{i+1}^{(e),(h)} \end{pmatrix} = B_k \begin{pmatrix} \bar{V}_{k,i}^{(h)} \\ \bar{\phi}_i^{(e),(h)} \end{pmatrix} \quad (2.36)$$

3 Multigrid Reduction in Time

Das „Multigrid Reduction in Time“ Verfahren (MGRIT) ist ein iteratives, zeitparalleles Verfahren zur Lösung zeitabhängiger partieller Differentialgleichungen. Traditionelle Zeitschrittverfahren sind sequentiell und benötigen die optimale Anzahl, das heißt so wenig wie möglich Anwendungen des Verfahrens. MGRIT besitzt diese Eigenschaft nicht. Um Parallelität in der Zeit zu ermöglichen, werden Zeitschritte mehrmals berechnet und damit mehr Rechenkraft benötigt. Jedoch existiert ein Übergangspunkt, an welchem das Rechnen auf mehreren Recheneinheiten gleichzeitig die zusätzlichen Rechenoperationen kompensiert. Über diesen Punkt hinaus, liefert die Methode eine Senkung der Laufzeit im Vergleich zu dem sequentiellen Verfahren.

In Abschnitt 3.1 wird das MGRIT Verfahren und dessen Eigenschaften beschrieben. Darauf folgend werden in Abschnitt 3.2 Fehlerschranken der Methode präsentiert und die benötigten Voraussetzungen diskutiert.

3.1 Das MGRIT Verfahren

MGRIT baut auf der Interpretation der Parareal Methode als Zwei-Level Mehrgitter Reduktionsverfahren aus Abschnitt 2.2.3 auf. Parareal benötigt jedoch eine sequentielle Lösung auf dem groben Gitter. Für große Probleme ist dies aufwendig und nicht parallel, womit die Vorteile eines Hochleistungsrechners nicht ausgenutzt werden können. Dies kann vermieden werden mit Ansätze der Mehrgitter Reduktionsverfahren. Durch rekursive Anwendung der Methode kann das Verfahren von einem Zwei-Level zu einem Multi-Level Verfahren umgewandelt werden, sodass letztendlich der sequentielle Teil auf einem sehr groben Gitter durchgeführt wird. Außerdem wird eine etwas andere Relaxationsmethode verwendet, deren Idee aus der Forschung optimaler Mehrgitter Reduktionsverfahren im Ort stammt. Zuerst werden alle benötigten Komponenten des Verfahrens beschrieben, analog zur Darstellung des geometrischen Mehrgitterverfahrens in Abschnitt 2.1.2. Daraufhin werden diese zum Algorithmus zusammengesetzt. Am Ende werden noch Eigenschaften der Methode diskutiert.

3.1.1 Die Komponenten des Verfahrens

Gitter

Betrachtet wird eine Gitterhierarchie bestehend aus äquidistanten Zeitgittern. Das feinste Gitter, das heißt das Gitter mit der Zeitschrittweite δt der ursprünglich gewählten Zeitdiskretisierung wird, als Gitterlevel 0 gesetzt; das größte Gitter wird als Gitterlevel L gesetzt. Wird die Zeitschrittweite eines Gitters mit dem Vergrößerungsfaktor $m \in \mathbb{N} \setminus \{1\}$ multipliziert, wird die Schrittweite des darauffolgenden, größeren Gitters bestimmt. Demnach hat das Gitter auf Gitterlevel 1 die Gitterweite

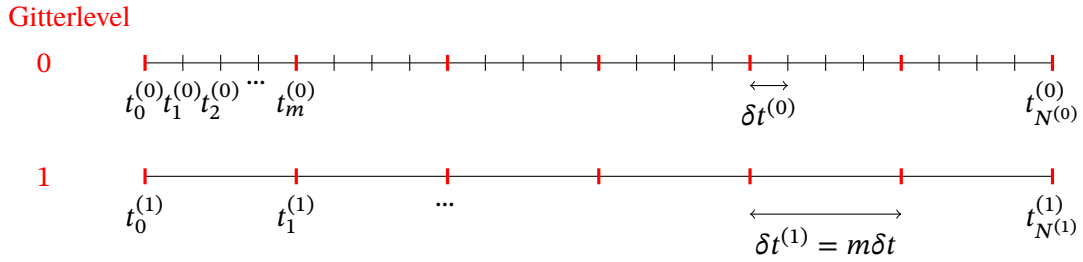


Abbildung 3.1: Zwei-Level MGRIT Gitterhierarchie; schwarze Zeitpunkte sind F-Punkte, rote sind C-Punkte

$m\delta t$; dies kann wiederholt werden bis das Zeitgitter genügend klein ist. Die insgesamt Anzahl an Zeitpunkten auf Gitter l ist $N^{(l)}$. C-Punkte sind auf einem Zeitgitter die Zeitpunkte, die auf dem darauf folgenden Gitter übernommen werden. Alle anderen Zeitpunkte sind F-Punkte. Abbildung 3.1 zeigt eine Zwei-Level Gitterhierarchie. Die rot gekennzeichneten Zeitpunkte sind C-Punkte.

Relaxation

In Abbildung 3.2 sind die zwei möglichen Relaxationstypen des MGRIT Verfahrens dargestellt. F-Relaxation aktualisiert die Werte an allen F-Punkten des Zeitgitters basierend auf den Werten der C-Punkte. Dafür wird das Einschrittverfahren der entsprechenden Zeitdiskretisierung angewendet.

Definition 3.1.1 (F-Relaxation)

Sei $N^T \in \mathbb{N}$ die Anzahl der C-Punkte auf dem betrachteten äquidistanten Zeitgitter, $m \in \mathbb{N} \setminus \{1\}$ der Vergrößerungsfaktor und $u_i \in \mathbb{R}^n$ die approximative Lösung an Zeitpunkt t_i . Außerdem sei Φ die zeitunabhängige Verfahrensfunktion eines Einschrittverfahrens, wie es in Definition 2.2.3 definiert wird. Dann ist für $k = 0, 1, \dots, N^{(\delta T)} - 1$ der Aktualisierungsschritt der F-Relaxation folgendermaßen definiert:

$$\begin{aligned} u_{km+1} &\leftarrow \Phi u_{km} + f_{km+1} \\ u_{km+2} &\leftarrow \Phi u_{km+1} + f_{km+2} \\ &\vdots \\ u_{(k+1)m-1} &\leftarrow \Phi u_{(k+1)m-2} + f_{(k+1)m-1} \end{aligned}$$

F-Relaxation entspricht einer sequentiellen Anwendung des Zeitschrittverfahrens auf einem F-Subintervall $[km, (k+1)m - 1]$. Die Rechnungen auf verschiedenen F-Subintervallen sind jedoch unabhängig voneinander und können parallel durchgeführt werden.

C-Relaxation aktualisiert statt dessen die Werte an allen C-Punkten des Zeitgitters, basierend auf den Werten der vorausgehenden F-Punkte. Diese kann folgendermaßen definiert werden:

Definition 3.1.2 (C-Relaxation)

Es gelten dieselben Voraussetzungen wie in der Definition 3.1.1 der F-Relaxation. Dann ist für $k = 1, 2, \dots, N^{(\delta T)}$ der Aktualisierungsschritt der C-Relaxation folgendermaßen definiert:

$$\begin{aligned} u_0 &\leftarrow \Phi f_0 \\ u_{km} &\leftarrow \Phi u_{km-1} + f_{km} \end{aligned}$$

Die C-Relaxation kann an allen Punkten parallel durchgeführt werden.

Parareal verwendet eine einmalige Anwendung der F-Relaxation als Relaxationsmethode. MGRITs Relaxationsstrategie ist die F-Relaxation gefolgt von einer C-Relaxation sowie eine erneute F-Relaxation, eine sogenannte **FCF-Relaxation**. Dies führt zu einer Veränderung des Fehlerfortpflanzungsoperators der Parareal Methode aus Gleichung (2.25). Der zweite Term $(I - E(E^T A E)^{-1} E^T A) \Leftrightarrow P R_I$ beschreibt die Fehlerfortpflanzung der F-Relaxation. Dieser kann durch folgendes Produkt ersetzt werden, welches die Fehlerfortpflanzung der FCF-Relaxation beschreibt:

$$(I - E(E^T A E)^{-1} E^T A)(I - R_I^T (R_I A R_I^T)^{-1} R_I A)(I - E(E^T A E)^{-1} E^T A) \quad (3.1)$$

$$\Leftrightarrow P(I - A_{CC}^{-1}(A_{CC} - A_{CF} A_{FF}^{-1} A_{FC})) R_I = P(I - A_{CC}^{-1}(R A P)) R_I = P(I - A^{(\delta T)}) R_I \quad (3.2)$$

Die letzte Äquivalenz besteht, da für ein Einschrittverfahren $A_{CC} = I$ gilt. FCF-Relaxation entspricht demnach einer Glättung mit Hilfe der Jacobi Methode auf dem größeren Zeitgitter.

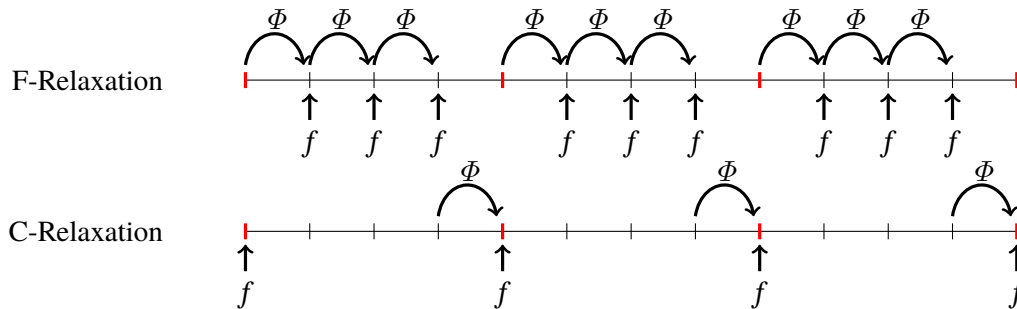


Abbildung 3.2: MGRITs Relaxationsschritte

Transfer

MGRIT verwendet die Injektion als Restriktionsoperator. Das bedeutet, dass die Werte an den C-Punkten auf die entsprechenden Gitterpunkte des darauffolgenden Zeitgitters kopiert werden. In Matrixschreibweise ist der Operator definiert als $R_I := \begin{pmatrix} 0 & I_C \end{pmatrix}$. Verglichen mit dem von Parareal verwendeten Restriktionsoperator $R := \begin{pmatrix} -A_{CF} A_{FF}^{-1} & I_C \end{pmatrix}$ benötigt der Injektionsoperator weniger Rechenarbeit. Trotzdem wird die Petrov-Galerkin Bedingung $A^{(\delta T)} = R A P = A_{CC} - A_{CF} A_{FF}^{-1} A_{FC} = R_I A P$ erfüllt.

Als Interpolationsoperator wird die ideale Interpolation $P := \begin{pmatrix} -A_{FF}^{-1} A_{FC} & I_C \end{pmatrix}^T$ von Parareal benutzt. Der Operator entspricht einer Injektion an den C-Punkten des feineren Gitters gefolgt von einer F-Relaxation.

Löser

Auf dem größten Gitter L muss ein Problem zur Berechnung der Fehlerapproximation gelöst werden. Dies geschieht durch sequentielle Anwendung des zum Zeitgitter passenden Zeitschrittverfahrens mit Verfahrensfunktion $\Phi^{(L)}$.

Konvergenzkriterium

MGRIT ist ein iteratives Verfahren, das heißt, es wird wiederholt angewendet bis ein Konvergenzkriterium erfüllt ist. Für MGRIT ist dies die euklidische Norm des Residuums an den C-Punkten des feinsten Gitters. Ist dieses globale Residuum $r^{(g)}$ kleiner als eine bestimmte Schranke, bricht das Verfahren ab.

$$r^{(g)} := \sqrt{\sum_{i=0}^{N^{(0)}/m^{(0)}} \|r_{mi}\|_{L_2}^2}$$

3.1.2 Das lineare Zwei-Level MGRIT Verfahren

All diese Komponenten werden nun zu einem Algorithmus zusammengesetzt. Algorithmus 3.1 beschreibt das Zwei-Level MGRIT Verfahren für den Fall einer linearen Gleichung. Die Indexe δt und δT aus der Herleitung der Methode wurden durch die entsprechenden Gitterlevel ersetzt. Wird die FCF-Relaxation mit der F-Relaxation ersetzt, so ist das Verfahren äquivalent zu Parareal.

Algorithmus 3.1 Zwei-Level MGRIT (linear)

```

while Residuumsnorm > Toleranz do
  Wende FCF-Relaxation auf  $A^{(0)}v^{(0)} = f^{(0)}$  an
  Injiziere das Residuum an den C-Punkten:  $r_i^1 \leftarrow f_{im}^{(0)} - (A^{(0)}v^{(0)})_{im}$ 
  Löse  $A^{(1)}e^1 = r^{(1)}$ 
  Injiziere  $e^1$  an den C-Punkten des feinen Gitters und addiere den Korrekturterm  $v_{im}^{(0)} \leftarrow$ 
 $v_{im}^{(0)} + e_{im}^{(0)}$ 
  Wende F-Relaxation auf  $A^{(0)}v^{(0)} = f^{(0)}$  an
end while

```

3.1.3 Das nichtlineare Multi-Level MGRIT Verfahren

Durch rekursive Anwendung des Verfahrens in Schritt Drei wird die Zwei-Level Methode zu einer Multi-Level Methode. Dadurch kann das sequentielle Lösen auf dem größten Gitter auch für große Probleme klein gehalten werden. Außerdem ist es möglich, das Verfahren so zu modifizieren, dass es auch für nichtlineare Gleichungen wie die partiellen Differentialgleichungen, die das neuromuskuläre System beschreiben, konvergiert. Dazu werden MGRITs Komponenten in das in Abschnitt 2.1.3 beschriebene „Full Approximation Scheme“ integriert.

Algorithmus 3.2 beschreibt das MGRIT Verfahren mit V-Zyklen. Es ist ohne Probleme möglich, dieselben Zyklenarten des geometrischen Mehrgitterverfahrens auf MGRIT zu übertragen. Diese werden in Abbildung 2.4 dargestellt.

Algorithmus 3.2 Ein V-Zyklus des Multigrid Reduction in Time

```

procedure MGRIT( $l, v^{(l)}, f^{(l)}$ )
  if  $l=L$  then
    Löse  $A^{(L)}u^{(L)} = A^{(L)}v^{(L)} + f^{(L)}$ 
    Berechne den approximativen Fehler  $e^{(L)} = u^{(L)} - v^{(L)}$ 
  else
    Wende FCF-Relaxation auf  $A^{(l)}v^{(l)} = f^{(l)}$  an
    Injiziere das Residuum an den C-Punkten:  $r_i^{(l+1)} \leftarrow f_{im}^{(l)} - (A^{(l)}v^{(l)})_{im}$  und setze  $f^{(l+1)} =$ 
 $r^{(l+1)}$ 
    Injiziere die Approximationslösung an den C-Punkten:  $v_i^{(l+1)} \leftarrow v_{im}^{(l)}$ 
    Berechne den approximativen Fehler  $e^{(l+1)} = \text{MGRIT}(l+1, v^{(l+1)}, f^{(l+1)})$  // Rekursiv
    Injiziere  $e^{(l+1)}$  an den C-Punkten des Gitters  $l$  und addiere den Korrekturterm  $v_{im}^{(l)} \leftarrow$ 
 $v_{im}^{(l)} + e_{im}^{(l)}$ 
    Wende F-Relaxation auf  $A^{(l)}v^{(l)} = f^{(l)}$  an
  end if
end procedure

```

3.1.4 Eigenschaften

Ein großer Vorteil MGRITs gegenüber anderen zeitparallelen Verfahren ist, dass der Algorithmus nur sehr wenige restriktive Bedingungen an die verwendete Diskretisierung sowie das verwendete Zeitschrittverfahren voraussetzt. In keinem Schritt des Verfahrens wird bis auf die vorausgesetzte Stabilität der Methode eine spezielle Eigenschaft der Verfahrensfunktion gefordert. Dies hat zur Folge, dass ein schon existierender, in der Zeit sequentieller Löser in MGRIT einfach integriert werden kann.

Parallelisierung

MGRIT ermöglicht es, sowohl in den Raumdimensionen als auch in der Zeitdimension des Problems Rechnungen parallel durchzuführen. Die Aufteilung des Ortsproblems auf Prozessoren, die das in der Zeit sequentielle Zeitschrittverfahren unternimmt, behält MGRIT bei. MGRIT unterteilt zusätzlich die Zeitdimension. Abbildung 3.3 skizziert die resultierenden Raum-Zeit-Gebiete für das sequentielle Zeitschrittverfahren sowie MGRIT.

Für viele zeitabhängige Probleme lohnt sich nur die Nutzung einer limitierten Anzahl an Prozessen im Ort. Ist es jedoch möglich, mehr Prozesse zu nutzen, so können diese mit Hilfe von MGRIT in der Parallelisierung der Zeitdimension genutzt werden. Es gilt jedoch zu beachten, dass sehr viele Prozesse benötigt werden, um ein stark paralleles Problem im Ort zusätzlich in der Zeit zu parallelisieren. Das liegt daran, dass jeder Zeitabschnitt, der parallel berechnet werden soll, die volle Anzahl an Prozessoren im Raum benötigt. Wird zum Beispiel ein Problem betrachtet, welches

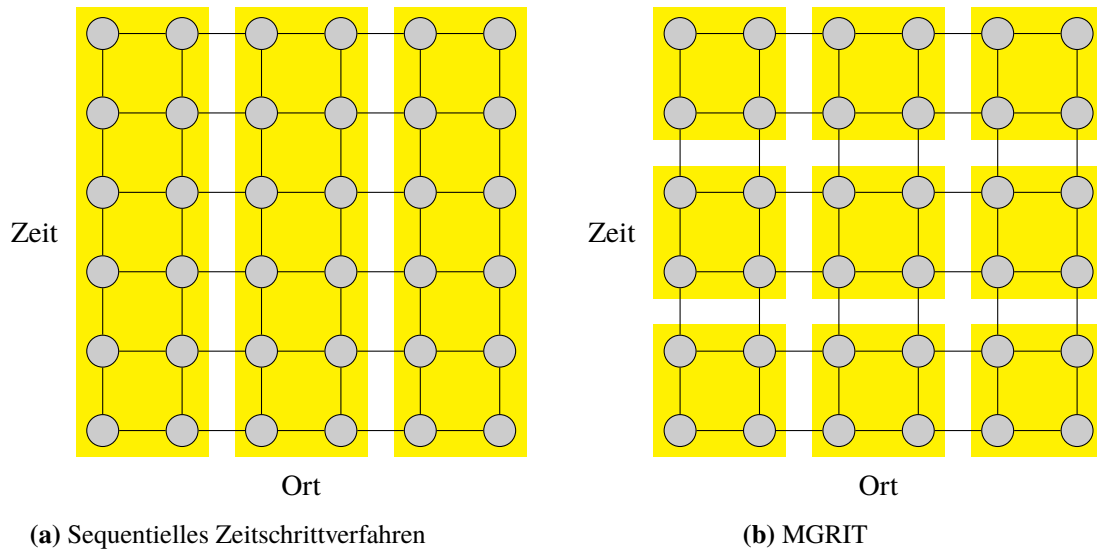


Abbildung 3.3: Aufteilung in Raum-Zeit Teilgebiete, auf denen parallel gerechnet werden kann

im Ort mit 16 Prozessen gelöst wird, so werden $16 \times 16 = 256$ Prozesse benötigt um auch in der Zeit auf 16 Zeitintervallen gleichzeitig zu rechnen. Daher ist MGRIT vor allem für Rechnungen und Simulationen auf Hochleistungsrechnern geeignet.

Anzahl an Iterationen

MGRIT benötigt nur wenige Iterationen um zu konvergieren. Ausgehend von exakter Arithmetik wird mit einer F-Relaxation die exakte Lösung an den F-Punkten des ersten F-Subintervalls berechnet, das heißt an allen F-Punkten zwischen t_0 und t_m mit Vergrößerungsfaktor m . FCF-Relaxation bestimmt bei einmaliger Anwendung die exakte Lösung an allen F-Punkten der ersten beiden F-Subintervalle sowie an C-Punkt t_m . Jede weitere Anwendung der Relaxationen bestimmt an weiteren Subintervallen die exakte Lösung. Dies führt dazu, dass Parareal nach $\frac{N^{(0)}}{m}$ Iterationen und MGRIT nach $\frac{N^{(0)}}{2^m}$ Iterationen die exakte diskrete Lösung berechnen, was der Anzahl an Zeitpunkten auf dem ersten gröberen Gitter beziehungsweise der Hälfte an Zeitpunkten entspricht. Damit das Verfahren aber tatsächlich eine verringerte Laufzeit ermöglichen kann, muss es in $\mathcal{O}(1)$ Iterationen konvergieren.

Speicher

Eine der größten Nachteile zeitparalleler Verfahren ist der benötigte Speicher (Falgout et al., 2017). MGRIT löst das Problem auf mehreren Subintervallen gleichzeitig; für jedes Subintervall wird daher zusätzlicher Speicherplatz nötig. Es ist nicht notwendig, an jedem Zeitpunkt des feinen Gitters einen Vektor zu speichern; sollte der Wert eines F-Punktes benötigt werden, kann dieser durch F-Relaxation ermittelt werden. Insgesamt muss aber an jedem C-Punkt in der gesamten Gitterhierarchie ein Lösungsvektor gespeichert werden, dies entspricht dem ersten Term der nachfolgenden Definition. Da das „Full Approximation Scheme“ verwendet wird, müssen auf den gröberen Gittern zusätzlich

zwei Hilfsvektoren gespeichert werden: die restringierte Lösung des feinen Gitters sowie das Residuum als rechte Seite des Grobgitterproblems. Dies wird mit dem zweiten Term der Gleichung beschrieben.

Lemma 3.1.3 (Benötigter Speicherplatz)

Betrachtet wird Algorithmus 3.2. Sei $L + 1$ die Anzahl der Gitter der Gitterhierarchie, $m^{(l)}$ der Vergrößerungsfaktor von Level l auf Level $l + 1$, $N^{(0)}$ die Anzahl der Zeitpunkte auf dem feinsten Gitter, p die Anzahl der temporären Prozesse und s der benötigte Speicherplatz zum Speichern eines Vektors. Der ungefähr benötigte Speicherplatz S pro temporärem Prozess kann dann folgendermaßen bestimmt werden:

$$S \approx \sum_{i=0}^{L-1} \left[\frac{N^{(0)}}{\prod_{l=0}^i m^l p} \right] s + \sum_{i=0}^{L-2} 2 \left[\frac{N^{(0)}}{\prod_{l=0}^i m^l p} \right] s$$

Dies entspricht einem Speichermultiplikator pro Prozess der Größenordnung $\mathcal{O}(\log_m N^{(0)})$. Für große Probleme mit vielen Zeitschritten kann dies problematisch werden. Es gibt verschiedene Strategien, den Speicherplatz klein zu halten.

Der meiste Speicherplatz wird an den C-Punkten des feinen Gitters benötigt, da diese im Gegensatz zu späteren Gittern zahlreich sind. Daher ist es ratsam, auf dem ersten Level stark zu vergrößern. Ein großer Vergrößerungsfaktor $m^{(0)}$ führt zu weniger C-Punkten auf dem feinen Gitter, deren Anzahl durch $\frac{N^{(0)}}{m^{(0)}}$ gegeben ist. Dabei ist jedoch zu beachten, dass dafür Parallelisierung eingebüßt wird. Durch größere Vergrößerungsfaktoren werden die F-Subintervalle größer und somit wird auch der sequentielle Teil der Relaxationsstrategie größer. Dies muss problemspezifisch abgewogen werden.

Eine weitere Möglichkeit zur Reduzierung des Speicherplatzbedarfs ist die Aufteilung des Gesamtproblems. Anstatt alle Zeitschritte auf einmal zu berechnen, kann das Zeitintervall in Subintervalle eingeteilt werden, welche nacheinander berechnet werden. Dadurch wird nicht der Speicherplatz für das gesamte Problem gebraucht, sondern nur für einen Teil davon. Auch diese Strategie büßt Parallelisierung für weniger benötigten Speicherplatz ein.

3.1.5 Das XBraid-Softwarepaket

XBraid ist ein Open Source Softwareprojekt in der Programmiersprache C des Lawrence Livermore National Laboratory (*XBraid: Parallel multigrid in time* o.D.). In dem Softwarepaket ist der MGRIT Algorithmus implementiert. Der Nutzer muss neben einem Zeitschrittverfahren nur einfache Basisoperationen selbst implementieren. Dies beruht darauf, dass die Methode nicht invasiv ist und leicht auf existierende, sequentielle Implementationen von Zeitschrittverfahren anwendbar ist. In dem Framework sind alle möglichen Optionen umsetzbar: insbesondere Zyklentypen, variable Zeitschritte, variable Wahl des Vergrößerungsfaktors sowie der Relaxationsstrategie und vieles mehr.

3.2 Theorie

MGRIT ist ein jüngerer Verfahren und daher ist seine theoretische Fundierung bisher nur ansatzweise ausgebaut. Friedhoff (2014) untersucht die Konvergenz MGRITs mit Hilfe von „semi-algebraic mode analysis“ für einfache Problemfälle. Eines der wichtigsten theoretischen Paper über MGRIT ist die Veröffentlichung von Dobrev et al. (2017), in welcher eine Konvergenztheorie des Zwei-Level MGRIT Verfahrens unter bestimmten Bedingungen wie Linearität sowie Kommutativität und Diagonalisierbarkeit der verschiedenen feinen Zeitschrittverfahren entwickelt wurde. Southworth generalisiert das Framework und leitet Konvergenzbedingungen ab (Southworth, 2019). Hessenthaler et al. (2020) weitet die Ergebnisse auf das Multi-Level Verfahren aus.

In dieser Arbeit wird ein Ausschnitt der wichtigsten Ergebnisse aus den Papern von Dobrev et al. (2017), Southworth (2019) und Hessenthaler et al. (2020) präsentiert. Zuerst wird der Fehlerfortpflanzungsoperator MGRITs definiert und eine Abschätzung des globalen Ort-Zeit-Fehlers gegeben. Darauf aufbauend werden etwas allgemeinere Fehlerschranken präsentiert. All diese Ergebnisse benötigen jedoch einschränkende Bedingungen bezüglich des Problems und des Zeitschrittverfahrens. Ein Theoriegebäude zu nichtlinearen und zeitabhängigen Zeitschrittverfahren existiert nach Kenntnisstand des Autors nicht; eine mögliche Übertragung der Ergebnisse von Parareal bezüglich nichtlinearer Zeitschrittoperatoren auf MGRIT wird am Ende des Kapitels diskutiert.

Dieses Kapitel setzt die Ergebnisse aus der Herleitung der MGRIT Methode in Abschnitt 2.2.3 fort. Sei Φ der invertierbare Operator eines feinen Zeitschrittverfahrens und sei $\Phi^{(\delta T)}$ der invertierbare grobe Zeitschrittoperator. Beide Operatoren entsprechen einer quadratischen Matrix $\Phi, \Phi^{(\delta T)} \in \mathbb{R}^{n \times n}$, deren Größe abhängig von der Anzahl an Unbekannten im Ort n ist. Folgende vier Bedingungen werden an die Zeitschrittverfahren gestellt:

1. Φ und $\Phi^{(\delta T)}$ sind linear.
2. Φ und $\Phi^{(\delta T)}$ sind stabil.
3. Φ und $\Phi^{(\delta T)}$ sind zeitunabhängig.
4. Φ und $\Phi^{(\delta T)}$ kommutieren und sind unitär diagonalisierbar.

Linearität und Zeitunabhängigkeit erlauben eine einfachere und detailliertere Analyse MGRITs. Stabilität des Zeitschrittverfahrens ist eine normale und nachvollziehbare Annahme. Die vierte Bedingung ist auf den ersten Blick einschränkend, für skalare partielle Differentialgleichungen jedoch wenig restriktiv. Alle Runge-Kutta-Methoden sowie fast alle Zeitschrittverfahren sind rationale Funktionen eines invertiblen Operators \mathcal{L} , welcher dem Operator der Ortsdiskretisierung entspricht. Sind also beide Zeitschrittverfahren Φ und $\Phi^{(\delta T)}$ rationale Funktionen des Operators \mathcal{L} , so kommutieren sie auch, da jede rationale Funktion des Operators kommutiert. Ist \mathcal{L} zusätzlich diagonalisierbar, so sind auch die Zeitschrittverfahren diagonalisierbar. Die meisten Zeitschrittoperatoren angewandt auf skalare partielle Differentialgleichungen sind unitär diagonalisierbar; für Operatoren, die von Systemen partieller Differentialgleichungen abstammen, trifft dies nicht zu, da diese häufig nicht symmetrisch sind.

Sei $u_i \in \mathbb{R}^n$ der Lösungsvektor an Zeitpunkt t_i und f_i die entsprechende rechte Seite. MGRIT wird nun zum Lösen des folgenden Gleichungssystem in Block Matrix Form angewandt:

$$Au \equiv \begin{pmatrix} I & & & \\ -\Phi & I & & \\ & \ddots & \ddots & \\ & & -\Phi & I \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{pmatrix} \equiv f \quad (3.3)$$

Sequentielle Zeitschrittverfahren lösen das System durch direktes Vorwärtseinsetzen während MGRIT dies iterativ löst.

Der Fehler der Approximationslösung $v_i \approx u_i$ sei $e_i = u_i - v_i$ und das Residuum sei $r_i = f_i - Av_i = Ae_i$. Neben der diskreten l_2 -Norm ($\|\cdot\|$) wird die A^*A -Norm ($\|\cdot\|_{A^*A}$) betrachtet, welche folgendermaßen definiert sind:

$$\|e_i\|^2 = \langle e_i, e_i \rangle, \quad \|e_i\|_{A^*A}^2 = \langle A^*Ae_i, e_i \rangle = \|r_i\|^2$$

3.2.1 Fehlerfortpflanzungsoperator

Da MGRIT ein Fixpunktverfahren ist, kann eine Schranke der Konvergenzrate durch Untersuchung der Norm des Fehlerfortpflanzungsoperator \mathcal{E} bestimmt werden. Für eine beliebige Approximation M der Matrix A ist der Fehlerfortpflanzungsoperator eines Fixpunktverfahrens gegeben als $\mathcal{E} = I - M^{-1}A$. Zusätzlich kann der Residuumsfortpflanzungsoperator \mathcal{R} betrachtet werden. Durch folgende Umformungen

$$e_i = \mathcal{E}^i e_0 \Leftrightarrow A^{-1}r_i = \mathcal{E}^i A^{-1}r_0 \Leftrightarrow (A\mathcal{E}A^{-1})^i r_0$$

stellt sich heraus, dass $\mathcal{R} = (A\mathcal{E}A^{-1}) = I - AM^{-1}$ gilt. Durch Einsetzen der Normdefinitionen kann gefolgert werden, dass die A^*A -Norm des Fehlerfortpflanzungsoperator äquivalent zur diskreten l_2 -Norm des Residuumsfortpflanzungsoperators ist:

$$\|\mathcal{E}\|_{A^*A}^2 = \sup_{y \neq 0} \frac{\langle A^*A\mathcal{E}y, \mathcal{E}y \rangle}{\langle A^*Ay, y \rangle} = \sup_{x \neq 0} \frac{\langle A\mathcal{E}A^{-1}x, A\mathcal{E}A^{-1}x \rangle}{\langle x, x \rangle} = \sup_{x \neq 0} \frac{\langle \mathcal{R}x, \mathcal{R}x \rangle}{\langle x, x \rangle} = \|\mathcal{R}\|^2 \quad (3.4)$$

Das bedeutet, dass wenn einer der beiden Operatoren bekannt ist der andere mittels der Äquivalenz Gleichung (3.4) einfach bestimmt werden kann. Es wird deutlich, dass der Fehlerfortpflanzungsoperator $\mathcal{E} = I - M^{-1}A$ ein Maß der Güte der Matrix M als Approximation einer Linksinversen der Matrix A ist und der Residuumsfortpflanzungsoperator $\mathcal{R} = I - AM^{-1}$ ein Maß der Güte der Matrix M als Approximation einer Rechtsinversen der Matrix A ist.

In Abschnitt 2.2.3 wurde der Fehlerfortpflanzungsoperator der Parareal Methode hergeleitet. Die Methode approximiert die exakte reduzierte Systemmatrix $A^{(\delta T)}$ mit der Matrix $B^{(\delta T)}$ mittels des groben Zeitschrittverfahrens $\Phi^{(\delta T)}$:

$$A^{(\delta T)} = \begin{pmatrix} I & & & \\ -\Phi^m & I & & \\ & \ddots & \ddots & \\ & & -\Phi^m & I \end{pmatrix}, \quad B^{(\delta T)} = \begin{pmatrix} I & & & \\ -\Phi^{(\delta T)} & I & & \\ & \ddots & \ddots & \\ & & -\Phi^{(\delta T)} & I \end{pmatrix}$$

Um den Fehlerfortpflanzungsoperator Parareals herzuleiten, wurde die Matrix A folgendermaßen aufgeteilt:

$$A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} = \begin{pmatrix} I_F & 0 \\ A_{CF}A_{FF}^{-1} & I_C \end{pmatrix} \begin{pmatrix} A_{FF} & 0 \\ 0 & A_{CC} - A_{CF}A_{FF}^{-1}A_{FC} \end{pmatrix} \begin{pmatrix} I_F & A_{FF}^{-1}A_{FC} \\ 0 & I_C \end{pmatrix}$$

Zusätzlich wurden folgende Operatoren definiert:

$$R := \begin{pmatrix} -A_{CF}A_{FF}^{-1} & I_C \end{pmatrix}, \quad R_I := \begin{pmatrix} 0 & I_C \end{pmatrix}, \quad P := \begin{pmatrix} -A_{FF}^{-1}A_{FC} \\ I_C \end{pmatrix}, \quad E := \begin{pmatrix} I_F \\ 0 \end{pmatrix}$$

Letztendlich ergaben sich dann folgende äquivalente Darstellungen des Fehlerfortpflanzungsoperator Parareals:

$$(I - P(B^{(\delta T)})^{-1}RA)(I - E(E^TAE)^{-1}E^TA) \quad (3.5)$$

$$\Leftrightarrow (I - P(B^{(\delta T)})^{-1}RA)PR_I \quad (3.6)$$

$$\Leftrightarrow P(I - (B^{(\delta T)})^{-1}A^{(\delta T)})R_I \quad (3.7)$$

Das Parareal Verfahren ist äquivalent zum Zwei-Level MGRIT Verfahren mit F-Relaxation. Wird anstattdessen FCF-Relaxation verwendet, muss der zweite Term des Fehlerfortpflanzungsoperators angepasst werden, siehe Abschnitt 3.1.1. In Gleichung (3.1) wird der modifizierte zweite Term beschrieben. Dies wird in dem folgenden Lemma zusammengefasst.

Lemma 3.2.1 (Fehlerfortpflanzungsoperator des Zwei-Level MGRIT Verfahrens)

Der Fehlerfortpflanzungsoperator des Zwei-Level MGRIT Verfahrens mit F-Relaxation ist folgendermaßen gegeben:

$$\mathcal{E}_F = (I - (B^{(\delta T)})^{-1}A^{(\delta T)})PR_I$$

Der Fehlerfortpflanzungsoperator des Zwei-Level MGRIT Verfahrens mit FCF-Relaxation ist folgendermaßen gegeben:

$$\mathcal{E}_{FCF} = (I - (B^{(\delta T)})^{-1}A^{(\delta T)})P(I - A^{(\delta T)})R_I$$

Beweis Herleitung ■

Die Fehlerfortpflanzungsoperatoren aus Lemma 3.2.1 sind eher abstrakt. Daher wird ein weiteres Lemma präsentiert, in welchem der Fehler nach einer Iteration mit Hilfe von Φ und $\Phi^{(\delta T)}$ charakterisiert wird.

Lemma 3.2.2 (Fehler nach einer Iteration)

Sei u_i die Lösung des Systems in Gleichung (3.3) und sei $v_{i,(0)} = u_i - e_{i,(0)}$ eine Näherungslösung mit dem Fehler $e_{i,(0)}$; außerdem sei m der Vergrößerungsfaktor und $N^{(\delta T)}$ die Anzahl an C-Punkten auf dem feinen Gitter. Nach einer Iteration des Zwei-Level MGRIT Verfahrens mit F-Relaxation gilt für die Näherungslösung $v_{i,(1)} = u_i - e_{i,(1)}$. Der Fehler $e_{i,(1)}$ wird durch folgende Gleichung an den C-Punkten charakterisiert:

$$e_{im,(1)} = \begin{cases} 0 & , i = 0 \\ \sum_{q=0}^{i-1} (\Phi^{(\delta T)})^{i-1-q} (\Phi^m - \Phi^{(\delta T)}) e_{qm,(0)} & , i = 1, 2, \dots, N^{(\delta T)} \end{cases}$$

siehe Lemma 3.2.1, und die Fehlerschranken von Southworth (2019) gelten an allen Punkten, jedoch nicht für die erste Iteration. Diese Einschränkungen bereiten für das Zwei-Level Szenario keine Probleme; für den Multi-Level Fall müssen die Abschätzungen jedoch für alle Iterationen und alle Punkte gelten. Wird zum Beispiel eine Drei-Level Gitterhierarchie betrachtet, so wird auf Level Eins das Zwei-Level MGRIT Verfahren zur Berechnung des Korrekturterms angewendet. Mit den Resultaten der Zwei-Level Theorie sinkt der Fehler an den C-Punkten, es ist jedoch möglich, dass der insgesamt Fehler auf Level Eins ansteigt. Dadurch würde die Korrektur auf Level Null die Approximation verschlechtern und das Verfahren im Regelfall divergieren.

Der Vollständigkeit halber werden folgend die Fehlerfortpflanzungsoperatoren der Multi-Level MGRIT Methode präsentiert. Interessierte Leser können Herleitung und Beweise in der Veröffentlichung von Hessenthaler et al. (2020) nachlesen. Operatoren auf Level l werden durch den Superskript $(\cdot)^{(l)}$ gekennzeichnet; daher ist $B^{(l)}$ die Approximation des exakten reduzierten Operators $A^{(l)}$ auf Gitterlevel l (auf Gitterlevel 0 gilt $B^{(0)} = A^{(0)}$). Letztendlich ist der Multi-Level Fehlerfortpflanzungsoperator ein Ergebnis aus Gleichung (2.23) und Gleichung (2.22).

Lemma 3.2.3 (Fehlerfortpflanzungsoperator des Multi-Level MGRIT Verfahrens mit V-Zyklus)

Es wird eine Zeitgitterhierarchie mit $L + 1$ Gitterleveln betrachtet. Der Fehlerfortpflanzungsoperator des Multi-Level MGRIT Verfahrens mit F -Relaxation und V -Zyklus ist folgendermaßen gegeben:

$$\begin{aligned} \mathcal{E}_F^{(L+1)} &= P^{(0)}R_{I^{(0)}} - \left(\prod_{i=0}^{L-1} P^{(i)} \right) (B^{(L)})^{-1} \left(\prod_{i=L-1}^0 R^{(i)} \right) A^{(0)}P^{(0)}R_{I^{(0)}} \\ &\quad - \sum_{j=0}^{L-2} \left(\prod_{i=0}^j P^{(i)} \right) E^{(j+1)} \left((E^{(j+1)})^T B^{(j+1)} E^{(j+1)} \right)^{-1} (E^{(j+1)})^T \left(\prod_{i=j}^0 R^{(i)} \right) A^{(0)}P^{(0)}R_{I^{(0)}} \\ &= \left(I^{(0)} - (B^{(L)})^{-1}A^{(L)} - \sum_{j=0}^{L-2} E^{(j+1)} \left((E^{(j+1)})^T B^{(j+1)} E^{(j+1)} \right)^{-1} (E^{(j+1)})^T A^{(j+1)} \right) P^{(0)}R_{I^{(0)}} \end{aligned}$$

Der Fehlerfortpflanzungsoperator des Multi-Level MGRIT Verfahrens mit FCF -Relaxation und V -Zyklus ist folgendermaßen gegeben:

$$\begin{aligned} \mathcal{E}_{FCF}^{(L+1)} &= P^{(0)} \left(I^{(0)} - (R_{I^{(0)}}A^{(0)}R_{I^{(0)}}^T)^{-1}R_{I^{(0)}}A^{(0)}P^{(0)} \right) R_{I^{(0)}} \\ &\quad - \left(\prod_{i=0}^{L-1} P^{(i)} \right) (B^{(L)})^{-1} \left(\prod_{i=L-1}^0 R^{(i)} \left(I^{(0)} - B^{(i)}R_{I^{(0)}}^T (R_{I^{(0)}}B^{(i)}R_{I^{(0)}}^T)^{-1} R_{I^{(0)}} \right) \right) A^{(0)}P^{(0)}R_{I^{(0)}} \\ &\quad - \sum_{j=1}^{L-1} \left(\prod_{i=0}^{j-1} P^{(i)} \right) \left(E^{(j)} \left((E^{(j)})^T B^{(j)} E^{(j)} \right)^{-1} (E^{(j)})^T \left(I^{(j)} - B^{(j)}R_{I^{(j)}}^T (R_{I^{(j)}}B^{(j)}R_{I^{(j)}}^T)^{-1} R_{I^{(j)}} \right) \right. \\ &\quad \left. + R_{I^{(j)}}^T (R_{I^{(j)}}B^{(j)}R_{I^{(j)}}^T)^{-1} R_{I^{(j)}} \right) \left(\prod_{i=j-1}^0 R^{(i)} \left(I^{(i)} - B^{(i)}R_{I^{(i)}}^T (R_{I^{(i)}}B^{(i)}R_{I^{(i)}}^T)^{-1} R_{I^{(i)}} \right) \right) A^{(0)}P^{(0)}R_{I^{(0)}} \end{aligned}$$

Beweis Es wird auf das Paper von Hessenthaler et al. (2020) verwiesen. ■

Für den Zwei-Level Fall sind die Summen jeweils leer und das Resultat ist äquivalent zu Lemma 3.2.1. In dieser Arbeit wird auf weitere Multi-Level Ergebnisse nicht weiter eingegangen.

3.2.2 Fehlerabschätzung

Mit Hilfe der Fehleroperatoren wird in diesem Abschnitt eine Abschätzung des globalen Raum-Zeit Fehlers der Zwei-Level MGRIT Methode entwickelt. Ziel ist mit Hilfe der Eigenvektoren der Verfahrensfunktionen die Zeitschrittverfahren und damit Blöcke des Fehlerfortpflanzungsoperators des groben Gitters $\mathcal{E}^{(\delta T)}$ zu diagonalisieren.

Es werden alle vier Bedingungen aus der Liste am Anfang des Theoriekapitels an die verwendeten Zeitschrittverfahren vorausgesetzt. Die unitäre Diagonalisierbarkeit in Kombination mit der Kommutation der Zeitschrittverfahren bedeutet, dass diese simultan unitär diagonalisierbar beziehungsweise unitär diagonalisierbar mit denselben Eigenvektoren sind. Sei $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{n \times n}$ eine unitäre Matrix bestehend aus den Eigenvektoren x_i von Φ und $\Phi^{(\delta T)}$. Seien λ_i die dazu passenden Eigenwerte des feinen Zeitschrittoperators und μ_i die des groben Zeitschrittoperators. Sei X^* die adjungierte Matrix der Matrix X . Die Operatoren können dann folgendermaßen transformiert werden:

$$\begin{aligned}\hat{\Phi} &= X^* \Phi X = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \\ \hat{\Phi}^{(\delta T)} &= X^* \Phi^{(\delta T)} X = \text{diag}(\mu_1, \mu_2, \dots, \mu_n)\end{aligned}$$

Eine Eigenvektor-Expansion des Fehlers vor und nach einer MGRIT Iteration ermöglicht die Analyse der Konvergenzeigenschaften der Methode. Dazu wird folgender Vektor definiert, welcher die nötigen Konstanten der Eigenvektor-Expansionen beinhaltet. Der erste Index entspricht dem Zeitpunkt und der zweite dem zum Vektor passenden Eigenvektor x_i .

$$\hat{e}_w = \begin{pmatrix} \hat{e}_{0,w} \\ \hat{e}_{m,w} \\ \hat{e}_{2m,w} \\ \vdots \\ \hat{e}_{N^{(\delta T)}m,w} \end{pmatrix} = \begin{pmatrix} x_w^* e_0 \\ x_w^* e_m \\ x_w^* e_{2m} \\ \vdots \\ x_w^* e_{N^{(\delta T)}m} \end{pmatrix}, \quad w = 1, 2, \dots, n$$

Damit sieht die Eigenvektor-Expansion des Fehlers vor und nach einer MGRIT Iteration folgendermaßen aus:

$$e_{i,(0)} = \sum_{w=1}^n \hat{e}_{i,w,(0)} x_w, \quad e_{i,(1)} = \sum_{w=1}^n \hat{e}_{i,w,(1)} x_w, \quad i = m, 2m, \dots, N^{(\delta T)}m$$

Unter der Ausnutzung, dass X unitär ist und somit $XX^* = I$ gilt, kann die Gleichung aus Lemma 3.2.2 durch Multiplikation mit Eins so umgeformt werden, dass die Konstanten der Eigenvektor-Expansionen bestimmt werden können. Für das Zwei-Level MGRIT Verfahren mit F-Relaxation für $w = 1, 2, \dots, n$ sieht das folgendermaßen aus:

$$\hat{e}_{im,w,(1)} = \begin{cases} 0 & , i = 0 \\ \sum_{q=0}^{i-1} (\mu_w)^{i-1-q} (\lambda_w^m - \mu_w) \hat{e}_{qm,w,(0)} & , i = 1, 2, \dots, N^{(\delta T)}m \end{cases}$$

Mit Hilfe dieses Lemmas kann der folgende Satz gezeigt werden:

Satz 3.2.5 (Fehlerabschätzung des Zwei-Level MGRIT Verfahrens)

Φ und $\Phi^{(\delta T)}$ erfüllen die vier Bedingungen. Dann gilt für den globalen Raum-Zeit Fehler der Zwei-Level MGRIT Methode an den C-Punkten $\bar{e} = ((e_{0,(0)})^T, (e_{m,(0)})^T, (e_{2m,(0)})^T, \dots, (e_{N^{(\delta T)}m,(0)})^T)^T$ folgende Ungleichungen:

$$\begin{aligned} \|\mathcal{E}_F^{(\delta T)}\bar{e}\| &\leq \max_{1 \leq w \leq n} \left\{ |\lambda_w^m - \mu_w| \frac{(1 - |\mu_w|^{N^{(\delta T)}})}{(1 - |\mu_w|)} \right\} \|\bar{e}\| \\ \|\mathcal{E}_{FCF}^{(\delta T)}\bar{e}\| &\leq \max_{1 \leq w \leq n} \left\{ |\lambda_w^m - \mu_w| |\lambda_w|^m \frac{(1 - |\mu_w|^{N^{(\delta T)}-1})}{(1 - |\mu_w|)} \right\} \|\bar{e}\| \end{aligned}$$

Beweis Die Fehlerfortpflanzung einer Iteration des Zwei-Level MGRIT Verfahrens kann durch $\hat{e}_{w,(1)} = \mathcal{E}_{F,w}^{(\delta T)} \hat{e}_{w,(0)}$ dargestellt werden. Unter Ausnutzung der Definition der euklidischen Norm, Verwendung dieser Darstellung und Anwendung des Lemma 3.2.4 gilt:

$$\|\mathcal{E}_F^{(\delta T)}\bar{e}\|^2 = \sum_{i=0}^{N^{(\delta T)}} \|e_{i,(1)}\|^2 = \sum_{w=1}^n \|\hat{e}_{w,(1)}\|^2 = \sum_{w=1}^n \|\mathcal{E}_{F,w}^{(\delta T)} \hat{e}_{w,(0)}\|^2$$

Dies kann nun abgeschätzt werden:

$$\|\mathcal{E}_F^{(\delta T)}\bar{e}\|^2 = \sum_{w=1}^n \|\mathcal{E}_{F,w}^{(\delta T)} \hat{e}_{w,(0)}\|^2 \leq \left(\max_{1 \leq w \leq n} \|\mathcal{E}_F^{(\delta T)}\|^2 \right) \sum_{w=1}^n \|\hat{e}_{w,(0)}\|^2 = \|\mathcal{E}_F^{(\delta T)}\|^2 \|\bar{e}\|^2 \quad (3.10)$$

Im letzten Schritt wurde erneut das Hilfslemma ausgenutzt. Folgende Normungleichung gilt für den Operator $\mathcal{E}_F^{(\delta T)}$:

$$\|\mathcal{E}_F^{(\delta T)}\| \leq \sqrt{\|\mathcal{E}_F^{(\delta T)}\|_1 \|\mathcal{E}_F^{(\delta T)}\|_\infty} = \|\mathcal{E}_F^{(\delta T)}\|_1 = \|\mathcal{E}_F^{(\delta T)}\|_\infty$$

Wird diese Ungleichung und die Gleichung (3.8) in die Abschätzung in Gleichung (3.10) eingesetzt folgt das Resultat für das Zwei-Level MGRIT Verfahren mit F-Relaxation. Die Abschätzung des Zwei-Level MGRIT Verfahrens mit FCF-Relaxation kann analog bewiesen werden. ■

Folgende Anmerkungen können bezüglich Satz 3.2.5 gemacht werden:

- In Southworth, 2019 wurde gezeigt, dass die Schranken aus Satz 3.2.5 scharf bis zur Ordnung $\mathcal{O}(1/N^{(\delta T)})$ sind.
- Satz 3.2.5 impliziert, dass im schlechtesten Fall der Konvergenzfaktor der Zwei-Level MGRIT Methode durch den max-Term beschränkt ist.
- Wird der Term $(1 - |\mu_w|^{N^{(\delta T)}})$ im Nenner näher betrachtet, fällt auf, dass eine schnellere Konvergenz bei kleinerem $N^{(\delta T)}$ erwartet werden kann, da $|\mu_w| < 1$ gilt.
- Im Gegensatz zu MGRIT mit F-Relaxation steht im Nenner der FCF-Relaxation der Term $|\lambda_w|^m$. Für hochfrequente Ortspunkte w gilt $|\lambda_w| \ll 1$. Daher kann in diesen Fällen die FCF-Relaxation signifikant schneller konvergieren.

Gerade die letzten beiden Erkenntnisse sind interessant für die numerischen Tests in Kapitel 4.

3.2.3 Differentialgleichungssysteme

Alle bisherigen Resultate setzen voraus, dass die vier Annahmen, welche am Anfang des Kapitels aufgezählt sind, für die Zeitschrittoperatoren gelten. Die vierte Bedingung, die Kommutation und unitäre Diagonalisierbarkeit, schließt jedoch die meisten Zeitschrittverfahren angewandt auf Differentialgleichungssysteme aus. Wird allerdings auf die Bedingung der unitären Transformation verzichtet und „nur“ die simultane Diagonalisierbarkeit vorausgesetzt, so gelten die generalisierteren Schranken auch für viele Zeitschrittverfahren in Anwendung auf Differentialgleichungssysteme.

Dobrev et al. (2017) erreichen dies mittels der Einführung der durch die Massenmatrix induzierten Norm. Die Äquivalenz aus dem Lemma 3.2.4 gilt ohne die unitäre Transformation nicht. Dies kann durch folgende Äquivalenz ersetzt werden. Für einen beliebigen Vektor v und der Matrix $M = (XX^*)^{-1}$ gilt:

$$\|v_i\|_M^2 = v_i^* M v_i = \|X^{-1} v_i\|^2 = \sum_{w=1}^{N(\delta T)} |\hat{v}_{i,w}|^2$$

Sei \mathcal{M} eine $N(\delta T) \times N(\delta T)$ Block Diagonalmatrix mit M als Diagonaleinträge. Dann kann der Fehler der Zwei-Level MGRIT Methode mit F-Relaxation folgendermaßen abgeschätzt werden:

$$\|\mathcal{E}_F^{(\delta T)} \bar{e}\|_{\mathcal{M}} \leq \left(\max_{1 \leq w \leq n} \mathcal{E}_{F,w}^{(\delta T)} \right) \|\bar{e}\|_{\mathcal{M}} = \max_{1 \leq w \leq n} \left\{ |\lambda_w^{(m)} - \mu_w| \frac{(1 - |\mu_w|^{N(\delta T)})}{(1 - |\mu_w|)} \right\} \|\bar{e}\|_{\mathcal{M}}$$

Für die FCF-Relaxation wird analog vorgegangen.

Sterck et al. (2019) und Southworth (2019) stellen alternative Vorgehensweisen vor. Diese werden hier kurz präsentiert; die jeweiligen Herleitungen und Beweise sprengen jedoch den Rahmen dieser Arbeit. Sterck et al. (2019) versuchen, mittels drei verschiedene Analyse-Werkzeuge (Raum-Zeit lokale Fourier Analyse, semi-algebraische Modi Analyse und die hier betrachtete Zwei-Level Reduktion) die Anwendung MGRITs auf hyperbolische Systeme zu untersuchen. Damit alle drei Methoden der Analyse vergleichbar sind, wird darauf Wert gelegt, dass die euklidische Norm des Fehlers beibehalten wird. Unter der Voraussetzung der simultanen Diagonalisierbarkeit der Zeitschrittoperatoren wird mittels eines Fourier Ansatzes eine Abschätzung des groben Fehlerfortpflanzungsoperator der Zwei-Level MGRIT Methode hergeleitet. Diese ist sehr ähnlich zu den in Satz 3.2.5 präsentierten Schranken; es werden jedoch die Eigenwerte der fouriertransformierten Zeitschrittoperatoren verwendet und die Schranke hängt zusätzlich von der Konditionszahl der Eigenvektormatrix der fouriertransformierten Zeitschrittoperatoren ab.

Southworth (2019) führt anstatt der Diagonalisierbarkeitsbedingung eine neuartige Voraussetzung an das grobe Zeitschrittverfahren ein. Das „temporal approximation property“ (TAP) misst, wie gut das feine Zeitschrittverfahren das grobe approximiert.

Definition 3.2.6 (Temporal Approximation Property)

Es gelte die übliche Notation. Dann erfüllt Φ die F-Relaxation TAP- p in Bezug auf $\Phi^{(\delta T)}$ mit der Konstanten $\varphi_{F,p}$, falls für alle Vektoren v gilt:

$$\|(\Phi^{(\delta T)} - \Phi^m)^p v\| \leq \varphi_{F,p} \left\{ \min_{x \in [0, 2\pi]} \|(I - e^{ix} \Phi^{(\delta T)})^p v\| \right\}$$

Φ erfüllt die FCF-Relaxation TAP- p in Bezug auf $\Phi^{(\delta T)}$ mit der Konstanten $\varphi_{FCF,p}$, falls für alle Vektoren v gilt:

$$\|(\Phi^{(\delta T)} - \Phi^m)^p v\| \leq \varphi_{FCF,p} \left\{ \min_{x \in [0, 2\pi]} \|\Phi^{-m}(I - e^{ix}\Phi^{(\delta T)})^p v\| \right\}$$

$\varphi_{F,p}$ ist typischerweise kleiner Eins. Damit der feine Zeitschrittoperator TAP erfüllt, muss Φ^m die Anwendung des Operators $\Phi^{(\delta T)}$ auf Vektoren v , welche $v \approx \Phi^{(\delta T)}v$ erfüllen, sehr gut approximieren. Die Anwendung auf andere Vektoren muss dagegen nicht so gut approximiert werden. Der zusätzliche Term der FCF-Relaxation ermöglicht im Regelfall die leichtere Erfüllung der Bedingung. Folgender Satz kann nun aufgestellt werden:

Satz 3.2.7 (Fehler in der A^*A -Norm und ausreichende Konvergenzbedingungen)

Seien die Zeitschrittoperatoren so gewählt, dass sie die ersten drei Bedingungen (linear, stabil, zeitunabhängig) erfüllen. Außerdem gelte für Φ sowohl F-TAP-1 in Bezug auf $\Phi^{(\delta T)}$ mit der Konstanten $\varphi_{F,1}$ als auch FCF-TAP-1 in Bezug auf $\Phi^{(\delta T)}$ mit der Konstanten $\varphi_{FCF,1}$. Dann gilt für Iterationen $j > 1$:

$$\begin{aligned} \frac{\|r_{F,(j+1)}\|}{\|r_{F,(j)}\|} &= \frac{\|e_{F,(j+1)}\|_{A^*A}}{\|e_{F,(j)}\|_{A^*A}} < \varphi_{F,1}(1 + \|(\Phi^{(\delta T)})^{N(\delta T)}\|) \\ \frac{\|r_{FCF,(j+1)}\|}{\|r_{FCF,(j)}\|} &= \frac{\|e_{FCF,(j+1)}\|_{A^*A}}{\|e_{FCF,(j)}\|_{A^*A}} < \varphi_{FCF,1}(1 + \|(\Phi^{-m}(\Phi^{(\delta T)})^{N(\delta T)}\Phi^m)\|) \end{aligned}$$

Somit sind $\varphi_{F,1}(1 + \|(\Phi^{(\delta T)})^{N(\delta T)}\|) < 1$ und $\varphi_{FCF,1}(1 + \|(\Phi^{-m}(\Phi^{(\delta T)})^{N(\delta T)}\Phi^m)\|) < 1$ ausreichende Konvergenzbedingungen in Bezug auf die A^*A -Norm auf allen außer einer Iteration.

Mit weiteren Voraussetzungen können Schranken anderer Normen oder auch schärfere Schranken hergeleitet werden. TAP ist ein interessanter Ansatz, um zu kontrollieren, ob das benutzte Zeitschrittverfahren geeignet für MGRIT ist. In dieser Arbeit wird dies jedoch nicht weiter verfolgt.

3.2.4 Nichtlinearität und Zeitabhängigkeit

Alle vorgestellten Resultate zur Konvergenz setzen Linearität und Zeitunabhängigkeit der Zeitschrittoperatoren voraus. Für das in dieser Arbeit betrachtete Multidomain Problem ist dies jedoch nicht erfüllt. Bis dato existiert kein - dem Autor bekanntes - Theoriegebäude für die Anwendung MGRITs auf solch komplexe Systeme partieller Differentialgleichungen.

Da aber MGRIT auf dem Parareal Verfahren basiert, wird an dieser Stelle ein Ausblick auf die existierende Theorie der Parareal Methode gegeben. M. J. Gander und Hairer (2008) präsentieren eine Analyse der Konvergenz Parareals für den nichtlinearen Fall. Es wird die Iterationsvorschrift Parareals (2.17) mit feinem Zeitschrittoperator F (Schrittweite δt) und grobem Zeitschrittoperator G (Schrittweite δT) betrachtet. Unter der Annahme, dass die Lösung des feinen Zeitschrittoperators F der exakten Lösung entspricht, werden folgende zwei Voraussetzungen benötigt. Die Differenz des feinen und des groben Zeitschrittoperators, welcher von Ordnung p sei, kann als folgende Expansion mit stetig differenzierbaren Funktionen c_j , $j = p + 1, p + 2, \dots$ und beliebigen Vektor x dargestellt werden:

$$F(t_{i+1}, t_i, x) - G(t_{i+1}, t_i, x) = c_{p+1}x\delta T^{p+1} + c_{p+2}x\delta T^{p+2} + \dots$$

Dies ist gegeben, wenn die rechte Seite f der Gleichung (2.16) glatt genug ist. Außerdem soll G folgende Lipschitzbedingung erfüllen; x und y seien beliebige, gleich große Vektoren:

$$\|G(t + \delta T, t, x) - G(t + \delta T, t, y)\| \leq (1 + C_2 \delta T) \|x - y\| \quad (3.11)$$

Mit diesen Bedingungen kann folgende Fehlerschranke aufgestellt werden:

Lemma 3.2.8

Sei $F(t + \delta T, t, U_{i-1})$ die exakte Lösung u_i und $G(t + \delta T, t, U_{i-1}) \approx F(t + \delta T, t, U_{i-1})$ eine Näherungslösung, deren lokaler Diskretisierungsfehler durch $C_3 \delta T^{p+1}$ beschränkt ist. Zusätzlich gelten die oben genannten Bedingungen. Dann gilt nach Iteration k des Parareal Verfahrens folgende Fehlerschranke:

$$\|u_i - U_{i,(k)}\| \leq \frac{C_3}{C_1} \frac{(C_1 \delta T^{p+1})^{k+1}}{(k+1)!} (1 + C_2 \delta T)^{i-k-1} \prod_{j=0}^k (i-j)$$

Beweis Für den Beweis wird auf das Paper von M. J. Gander und Hairer (2008) verwiesen. ■

Aus diesem Lemma geht hervor, dass Parareal für den nichtlinearen Fall superlinear konvergiert (Duarte et al., 2011).

Haut und Wingate (2013) zeigen jedoch, dass Parareal nur langsam konvergiert, falls es auf nichtlineare Gleichungen mit oszillierenden linearen Komponenten angewendet wird. Daher entwickelten Haut und Wingate (2013) die asymptotische Parareal Methode (APinT), welche die oszillierenden linearen Komponenten auf dem groben Gitter eliminiert. Dies geschieht durch Mittelwertbildung der Zeit auf dem groben Gitter und resultiert in sogenannten langsamen Problemen (Charney, 1948). Dies ermöglicht schnellere Konvergenz.

Duarte et al. (2011) untersuchen die Anwendung Parareals auf Splitting-Verfahren. In dem Paper wird gezeigt, dass Parareal reduzierte Konvergenzraten vorweist, wenn der grobe Zeitschrittoperator ein Splitting-Verfahren ist.

4 Numerische Tests

Eine Simulation des neuromuskulären Systems zum besseren Verständnis des menschlichen Körpers ist ein sehr rechenaufwendiges Problem. Für tausende Zeitschritte müssen Gleichungssysteme mit tausenden Unbekannten gelöst werden. Das „Digital Human“ Projekt versucht mittels Methoden des Hochleistungsrechnens eine realistische Simulation des neuromuskulären Systems zu ermöglichen. Dazu werden komplexe, im Ort hochparallelisierte numerische Verfahren implementiert.

Gerade für diese Aufgabenstellung bietet sich jedoch auch Parallelisierung in der Zeit an. Durch die Verwendung des expliziten Heun Verfahrens in dem Strang-Splitting Verfahren (siehe Abschnitt 2.3.3) ist das angewandte Zeitschrittverfahren nur für kleine Zeitschrittweiten stabil. Dies führt zu einer immensen Anzahl an Zeitschritten, welche traditionell sequentiell berechnet werden. Zeitparallele Methoden ermöglichen eine potentiell reduzierte Laufzeit mit der Verwendung von mehr Prozessoren.

Diese Arbeit untersucht die Zeitparallelisierung der Simulation des neuromuskulären Systems. Das „Multigrid Reduction in Time“ Verfahren ist dafür die Methode der Wahl. Das zu lösende System partieller Differentialgleichungen ist hochkomplex und benötigt daher spezielle numerische Behandlung. Die meisten alternativen zeitparallelen Verfahren, siehe Abschnitt 2.2.2, benötigen ihre eigenen Zeitschrittverfahren und/oder Diskretisierungen. Vielversprechende Verfahren wie die Raum-Zeit Mehrgittermethoden verwenden spezielle Ortsdiskretisierungen, was für das hier betrachtete 3D-Modell kaum umsetzbar ist. Methoden wie PFASST wenden ihre eigenen Zeitschrittverfahren an, welche für solch ein gekoppeltes System nicht geeignet sind. MGRIT hingegen hat mehrere Vorteile:

- MGRIT ermöglicht die Zeitparallelisierung eines vorhandenen, sequentiellen Zeitschrittverfahrens ohne dass dieses angepasst werden muss.
- MGRIT nimmt keinen Einfluss auf das Ortsproblem; dadurch können sowohl die Diskretisierung als auch vorhandene Laufzeitsenkungen durch Parallelisierung im Ort beibehalten werden.
- Durch das „Full Approximation Scheme“ ist die Lösung nichtlinearer Probleme möglich.
- Eine öffentlich zugängliche, gut dokumentierte Software-Bibliothek MGRITs (XBraid) existiert.

All dies zeigt das Potential MGRITs. Bei der Anwendung MGRITs müssen jedoch auch einige potentielle Probleme beachtet beziehungsweise untersucht werden.

- Die Anwendung eines (teilweise) expliziten Zeitschrittverfahrens kann zu Instabilität der Methode führen. Durch die Wahl größerer Zeitschrittweiten auf gröberen Gittern kann das Verfahren divergieren.

- Zeitparallele Methoden führen automatisch zu mehr benötigten Speicherplatz, siehe Abschnitt 3.1.4.
- Es existiert kein Theoriegebäude zu der Anwendung MGRITs auf nichtlineare, zeitabhängige Zeitschrittverfahren, siehe Abschnitt 3.2.4.
- Es hat sich herausgestellt, dass MGRIT angewendet auf nichtlineare Probleme - im Vergleich zur Anwendung auf lineare Probleme - Schwierigkeiten hat, dieselben guten Skalierungseigenschaften zu erreichen (Falgout et al., 2017).
- Bei der Verwendung vieler Prozesse im Ort werden sehr viele Prozesse für die Zeitparallelisierung benötigt.

Daher kann eine erfolgreiche Anwendung nicht vorausgesagt werden sondern muss untersucht werden.

Zuerst wird in Abschnitt 4.1 die Umsetzung der Anwendung MGRITs auf das Multidomain Problem beschrieben. Zusätzlich werden die Parameter der Testprobleme sowie technische Aspekte erläutert. In Abschnitt 4.2 wird eine Parameterstudie durchgeführt. Die gewonnenen Ergebnisse werden in Abschnitt 4.3 verwendet, in welchem Laufzeit- und Skalierungstests vorgestellt werden.

4.1 Umsetzung

Dieses Kapitel beschreibt alle nötigen Informationen, um die späteren Ergebnisse nachvollziehen zu können. Zuerst werden in Abschnitt 4.1.1 die Testproblemstellungen mit den entsprechenden Parametern beschrieben. Anschließend wird in Abschnitt 4.1.2 erklärt, wie die Anwendung MGRITs auf den Lösungsvorgang des Multidomain Problems realisiert wird. Abschließend wird die verwendete technische Testumgebung in Abschnitt 4.1.3 erläutert.

4.1.1 Modellparameter der Testproblemstellungen

Der Fokus der Arbeit liegt auf der Untersuchung der Anwendung MGRITs auf das in Abschnitt 2.3.3 vorgestellte Multidomain Problem. Für die Materialparameter des Multidomain Modells werden exakt dieselben Werte verwendet, welche Klotz et al. (2019) in ihrer Veröffentlichung präsentieren. Dies beinhaltet die Membrankapazität einer Muskelfaser C_k , das Verhältnis zwischen der Fläche der Hülle und dem Volumen einer Faser F_k sowie die Leitfähigkeitstensoren $\sigma^{(e)}$ und $\sigma_k^{(i)}$.

Die verwendete Zeitschrittweite auf dem originalen, feinen Gitter bleibt für alle Testläufe konstant bei $\delta t^{(MD)} = 0.003$ sowie $\delta t^{(H)} = 0.0015$. Eine größere Wahl der Schrittweite des Heun Verfahrens führt zu Instabilität des gesamten Lösungsvorgangs. Der Muskel wird mit einer Frequenz von 100 Hertz stimuliert; dadurch ist das zu lösende Problem nichtlinear.

Außerdem muss ein passendes dreidimensionales Ortsgitter gewählt werden. In allen Testläufen werden $49 = 7 \times 7$ Muskelfasern pro motorischer Einheit betrachtet. Entscheidend ist, wie viele Punkte pro Faser in Betracht gezogen werden. Klotz et al. (2019) erklären, dass die Simulation des neuromuskulären Systems mit Hilfe des Multidomain Modells auf das Lösen von Systemen steifer Differentialgleichungen hinausläuft. Es stellt sich heraus, dass die Auflösung des Ortes das Verhalten des Lösungsvorgangs des Multidomain Problems ändert. Je mehr Punkte pro Faser

betrachtet werden, desto steifer wird das zu lösende System. Damit das Strang-Splitting Verfahren trotzdem das Multidomain Problem effizient lösen kann, wird im Regelfall eine kleinere Schrittweite für das explizite Verfahren verwendet. In dieser Arbeit wird untersucht, ob die erhöhte Steifigkeit des Gleichungssystems einen Einfluss auf die Konvergenz MGRITs hat. Da die simulierte Zeitschrittweite konstant gewählt ist, kann dies durch zwei verschiedene Ortsauflösungen geschehen. Es werden die feine Ortsauflösung mit 149 Diskretisierungspunkten pro Faser sowie das gröbere Ortsgitter mit 21 Diskretisierungspunkten pro Faser betrachtet.

Insgesamt führt dies zu 7301 beziehungsweise 1029 Ortspunkten pro motorischer Einheit. Die Motivation der Entwicklung des Multidomain Problems war die daraus entstehende Möglichkeit, mehrere motorische Einheiten klar zuordenbar simulieren zu können. Für die feine Ortsdiskretisierung werden in jedem präsentierten Lauf zwei motorische Einheiten simuliert. Wird Gleichung (2.38) betrachtet, führt dies zu einem Gleichungssystem mit 21903 Unbekannten, welches in jedem Strang-Splitting Schritt gelöst werden muss. Für das Lösungsverfahren mit der gröberen Ortsdiskretisierung wird die Simulation mehrerer motorischen Einheiten betrachtet. In der Parameterstudie wird ein Problem mit 5 motorischen Einheiten, also ein Gleichungssystem mit 6174 Unbekannten, gelöst.

Zusätzlich zu den beiden Varianten des Multidomain Problems wird das eindimensionale Wärmeleitungsproblem aus Abschnitt 2.3.2 untersucht. Die Anwendung MGRITs auf dieses Problem wurde schon mehrmals getestet und dient daher als Vergleich; dies ermöglicht eine bessere Einordnung der Ergebnisse. Es wird eine Problemgröße von 6174 Unbekannten im Gebiet $[0, 4]$ gewählt.

Zusammenfassend wird die Lösung dreier Probleme betrachtet:

- *Diffusion* mit 6174 Unbekannten
- *Multidomain21* mit 6174 Unbekannten
- *Multidomain149* mit 21903 Unbekannten

In den folgenden Kapitel beschreiben die oben genannten Namen die entsprechenden Lösungsverfahren.

Alle Simulationsläufe werden bis zu der Abbruchtoleranz $1e-8$ durchgeführt; sobald nach einer MGRIT Iteration die euklidische Residuumsnorm auf dem feinen Gitter absolut kleiner als die Toleranz ist, ist das Verfahren konvergiert.

4.1.2 Anwendung MGRITs auf das Multidomain Problem

Wird MGRIT auf einen schon implementierten Lösungsvorgang angewendet, muss vor allem entschieden werden, wie die Grobgitteroperatoren funktionieren. Der übliche Vorgang ist die Verwendung des gleichen Zeitschrittverfahrens mit größerer Schrittweite. Dies funktioniert für A-stabile Zeitintegratoren wie zum Beispiel die implizite Euler Methode auf die Diffusionsgleichung angewandt sehr gut und ohne Probleme. Daher wird im Regelfall MGRIT nur im Zusammenhang mit impliziten Verfahren untersucht. Auf dieser Weise wird der Grobgitteroperator des *Diffusion* Problems implementiert.

An sich ist eine Zeitparallelisierung expliziter Verfahren jedoch sehr interessant, da diese durch die benötigte kleine Zeitschrittweite sehr viele Zeitschritte berechnen müssen. Friedhoff (2014) präsentiert in ihrer Dissertation auch Ergebnisse zum expliziten Euler Verfahren. Da durch die

Nutzung mehrerer Zeitgitter die verwendete Zeitschrittweite auf dem größten Gitter verglichen zu der Zeitschrittweite auf dem originalen Gitter sehr groß wird, verwendet die Autorin auf Grund der Stabilitätsprobleme der expliziten Methode auf allen größeren Gittern eine implizite Methode.

Dies wird in dem Anwendungsfall dieser Arbeit jedoch nicht weiter verfolgt. Der verwendete Löser ist sehr spezifisch an die Eigenschaften des Modells des neuromuskulären Systems angepasst und für den normalen, in der Zeit sequentiellen Fall optimiert. Das Ziel ist also die Anwendung MGRITs auf ein Strang-Splitting Verfahren, welches teilweise explizit und teilweise implizit ist. Der zuerst verwendete naive Ansatz, auf größeren Zeitgittern alle Zeitschrittweiten entsprechend zu erhöhen, ist instabil. Das Heun Verfahren divergiert auf den größeren Gittern und dadurch divergiert das gesamte MGRIT Verfahren.

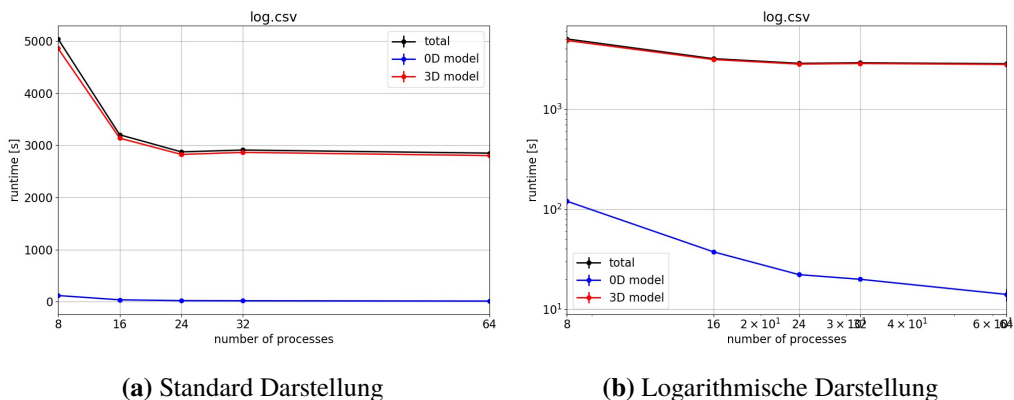


Abbildung 4.1: Starke Skalierung im Ort des *Multidomain149* Verfahren auf dem Intervall $[0, 1025]$. Totale Laufzeit (schwarz) sowie die Laufzeit des Heun Verfahrens (blau) und des Multidomain Löser (rot). Berechnet auf Hawk.

Abbildung 4.1 zeigt die benötigte Laufzeit der *Multidomain149* Methode mit zwei motorischen Einheiten und verschiedener Anzahl an Rechenkernen für eine Simulation mit Endzeit 1025 Millisekunden. Neben der gesamten Rechenzeit werden die Rechenzeiten des Multidomain Löser sowie des Heun Verfahrens dargestellt. In Abbildung 4.1a wird offensichtlich, dass das Heun-Verfahren nur einen Bruchteil der gesamten Laufzeit ausmacht. Werden 16 Porzessoren verwendet, benötigt der teure implizite Multidomain Löser 3193 Sekunden während das Heun Verfahren 37 Sekunden rechnet. Das Heun Verfahren macht demnach nur etwas mehr als ein Prozent der gesamten Laufzeit aus. Wird Abbildung 4.1b betrachtet, fällt zusätzlich auf, dass das Heun Verfahren im Gegensatz zu dem Multidomain Löser auch für ein kleineres Ortsproblem gut skaliert.

Dieser Umstand, dass der Multidomain Löser die Rechenzeit dominiert, wird nun für die Wahl der Zeitschrittverfahren auf dem größeren Gitter ausgenutzt. Das Strang-Splitting Verfahren verwendet zwei verschiedene, voneinander unabhängige Zeitschrittweiten. Zum Einen gibt es die Zeitschrittweite $\delta t^{(MD)}$ des Multidomain Löser. In dieser Arbeit ist diese äquivalent zur Weite eines Strang-Splitting Schrittes. Zum Anderen verwendet das Heun Verfahren die Zeitschrittweite $\delta t^{(H)}$. Auf einem größeren Gitter wird der Schritt des teuren, impliziten Multidomain Löser entsprechend größer gewählt; ein Schritt des expliziten Verfahrens bleibt an Stelle dessen konstant auf der gesamten Zeitgitterhierarchie. Zur Veranschaulichung wird folgendes Beispiel gegeben: Seien $\delta t^{(MD),(0)}$ und $\delta t^{(H),(0)} = \delta t^{(MD),(0)}/2$ die entsprechenden Zeitschrittweiten auf Gitterlevel 0. Wird nun der Vergrößerungsfaktor $m = 2$ gewählt, so gilt für den Multidomain Löser

$\delta t^{(\text{MD}), (1)} = 2 \cdot \delta t^{(\text{MD}), (0)}$. Die Zeitschrittweite der Heun Methode bleibt jedoch konstant; das bedeutet $\delta t^{(\text{H}), (1)} = \delta t^{(\text{MD}), (1)}/4 = \delta t^{(\text{H}), (0)}$. Pro Strang-Splitting Schritt auf größeren Gitter werden demnach mehr Schritte des Heun Verfahrens berechnet.

Durch den gewählten Aufbau des Grobgitteroperators bleibt das MGRIT Verfahren stabil. Die konstant kleine Zeitschrittweite ermöglicht die Konvergenz des expliziten Verfahren. Trotzdem ist die Anwendung des Zeitschrittverfahren auf dem größeren Gitter günstiger, da der Multidomain Löser, welcher die meiste Zeit der Rechenzeit verursacht, weniger oft angewendet werden muss.

4.1.3 Technische Aspekte

In diesem Abschnitt werden zuerst die verwendeten Computersysteme beschrieben. Darauf folgend werden mögliche Limitationen verursacht durch Hardware der in den folgenden Kapiteln präsentierten Ergebnisse diskutiert.

Die Programmläufe wurden auf zwei verschiedenen Rechensystemen durchgeführt. Das kleinere System „Helium“ besteht aus vier „Intel(R) Xeon(R) CPU E7540“ Prozessoren mit jeweils sechs Rechenkernen und einer Grundtaktfrequenz von zwei GHz. Insgesamt stehen 515976 Megabytes RAM zur Verfügung.

Außerdem konnten vereinzelte Simulationen auf einem echten Hochleistungsrechner berechnet werden. Dazu wurde der „HPE Apollo Hawk“ (kurz Hawk) Supercomputer des Höchstleistungsrechenzentrum Stuttgart benutzt, einer der schnellsten Rechner der Welt. Dieser erreicht eine Spitzenleistung von 26 Petaflops. An 5632 Rechenknoten sind jeweils zwei Prozessoren des Typs AMD EPYC 7742 mit jeweils 64 Rechenkernen verbaut. Insgesamt stehen also 720896 Rechenkerne zur Verfügung. Rechenknoten sind über InfiniBand HDR200 mit einer Bandbreite von 200 Gbit/s verbunden. Im gesamten System sind 1,44 Petabyte RAM verfügbar, pro Prozessor 128 Gigabytes.

Bei allen Testläufen wurde darauf geachtet, dass die Kommunikation zwischen Rechenknoten ausschließlich durch die Zeitparallelisierung und nicht durch die Ortsparallelisierung stattfindet. Das bedeutet, dass zum Beispiel alle 16 Prozesse des in der Zeit sequentiellen Löser auf demselben Knoten rechnen. Soll das Problem nun mit vier Prozessen parallel in der Zeit berechnet werden, werden $4 \cdot 16 = 64$ Prozesse benötigt. Dies kann auf einen oder auch auf zwei Knoten mit jeweils 32 verwendeten Rechenkernen stattfinden. Sollten zwei Knoten verwendet werden, so sind die 16 Prozesse pro Zeitprozess auf einem Knoten konzentriert. Anstatt dessen werden die Zeitprozesse aufgeteilt. Tests auf Hawk ergaben, dass in exakt diesem Fall die Verwendung von zwei Knoten trotz Kommunikation zwischen Knoten etwas geringere Laufzeiten lieferte als die Verwendung eines Knotens. Daher muss dies nicht weiter beachtet werden.

Zeitparallele Verfahren benötigen mehr Speicherplatz als entsprechende in der Zeit sequentielle Verfahren. Alleine um zu ermöglichen, dass verschiedene Rechenkerne an verschiedenen Zeitpunkten gleichzeitig rechnen können, muss an jedem dieser Zeitpunkte ein Lösungsvektor gespeichert werden. In Abschnitt 3.1.4 wird dies für MGRIT näher erläutert. Durch den sehr speichereffizienten Aufbau des OpenDiHu Frameworks und sorgfältiger Integrierung des MGRIT Verfahrens, ist dies für die in dieser Arbeit betrachteten Diskretisierungsgrößen und gewählten Anzahlen an Zeitschritten kein Problem. Das in der Zeit sequentielle Lösungsverfahren des Multidomain Problems benötigt für zwei motorische Einheiten des *Multidomain149* maximal 0.2 Gigabyte Speicher pro verwendeten Rechenkern gegenüber den zur Verfügung stehenden 2 Gigabyte (bei Nutzung

aller Rechenkerne des Rechenknotens). Die zusätzlich benötigten Vektoren MGRITs fügen wenig benötigten Speicherplatz hinzu. Trotzdem sollte ein größerer Vergrößerungsfaktor als $m = 2$ auf dem feinsten Level gewählt werden; für sehr große Simulationen wird außerdem auf die Hinweise in Abschnitt 3.1.4 verwiesen.

Zu guter Letzt wird erwähnt, dass die Visualisierung der Ergebnisse nicht beachtet wurde. Alle präsentierten Zeiten entsprechen der benötigten Rechenzeit zum Lösen des Problems. Die Ausgabe der Resultate hat keinen Einfluss auf das verwendete numerische Lösungsverfahren und wird daher ignoriert.

4.2 Parameterstudien

MGRIT ist kein Verfahren, welches direkt optimal anwendbar ist. Es gibt drei unterschiedliche Parameter des Verfahrens, welche größeren Einfluss auf das Konvergenzverhalten sowie die Rechendauer haben. Welche Vergrößerungsfaktoren machen Sinn? Welche Relaxation soll verwendet werden? Welcher Zyklus ist am besten geeignet? Die bestmögliche Wahl dieser Einstellungen für die in dieser Arbeit betrachtete Anwendung soll mittels dieser Parameterstudie ermittelt werden.

Die Ergebnisse der Studie werden in Tabellen präsentiert, damit sowohl Rechendauer als auch Anzahl der Iterationen übersichtlich dargestellt werden können. In diesem Unterkapitel wird kein Vergleich zum sequentiellen Verfahren gezogen; das heißt, interessant ist nicht der absolute Wert der Rechendauer sondern die Veränderung der Rechendauer unter dem Einfluss verschiedener Parameter.

Zuerst wird die Zwei-Level Methode mit verschiedenen Vergrößerungsfaktoren untersucht. Mit den daraus gewonnenen Ergebnissen werden Multi-Level Verfahren getestet. Alle Testläufe bis hierhin wurden mit der FCF-Relaxation des MGRIT Verfahren durchgeführt; der Einfluss der Relaxationsmethode wird daher an dieser Stelle analysiert. Zu guter Letzt werden noch die möglichen Zyklen diskutiert.

4.2.1 Einfluss des Vergrößerungsfaktors

Der Vergrößerungsfaktor m hat einen großen Einfluss auf die Eigenschaften des MGRIT Verfahrens. Je größer der Faktor desto günstiger sind die Zeitschrittverfahren auf den gröberen Gittern. Dafür ist die Approximation des Feingitteroperators weniger genau. Das zu erwartende Ergebnis für steigende Vergrößerungsfaktoren ist demnach eine Reduzierung der benötigten Zeit pro Iteration des MGRIT Verfahrens auf Kosten eines schlechteren Konvergenzfaktors pro Iteration.

Zusätzlich muss angemerkt werden, dass eine starke Vergrößerung auf dem ersten Level aus Gründen des Speicherplatzes Sinn macht. Werden 10000 Zeitschritte zeitparallelisiert mit einem Vergrößerungsfaktor $m = 2$, dann müssen mindestens 5000 Vektoren zusätzlich gespeichert werden. Für große Probleme wie eine realistische Simulation des neuromuskulären Systems ist dies nicht akzeptabel und sollte vermieden werden.

In Tabelle 4.1 sind für verschiedene Vergrößerungsfaktoren Iterationen und Rechenzeit der Zwei-Level MGRIT Methode mit FCF-Relaxation auf alle drei Testprobleme angewandt dargestellt.

m	#Iter	Dauer[s]
2	7	68
4	8	50
6	8	40
8	8	36
10	8	32
12	8	30
14	8	29
16	8	28
20	8	27
24	8	26
32	9	28
40	9	27

(a) *Diffusion*

m	#Iter	Dauer[s]
2	3	128
4	4	121
6	4	103
8	4	95
10	4	89
12	4	85
14	4	83
16	4	81
20	4	78
24	4	77
32	4	75
40	4	73

(b) *Multidomain21*

m	#Iter	Dauer[s]
2	5	619
4	6	507
6	7	503
8	7	466
10	8	499
12	8	475
14	9	509
16	9	508
20	10	541
24	10	531
32	11	569
40	12	608

(c) *Multidomain149*

Tabelle 4.1: Laufzeit und Anzahl Iterationen der Zwei-Level MGRIT Methode mit FCF-Relaxation für verschiedene Vergrößerungsfaktoren. *Diffusion* (10000 Zeitschritte, 8 Zeitprozesse mit jeweils 1 Ortsprozess) wurde auf Helium berechnet. *Multidomain21* (10000 Zeitschritte, 8 Zeitprozesse mit jeweils 4 Ortsprozessen) und *Multidomain149* (4000 Zeitschritte, 8 Zeitprozesse mit jeweils 8 Ortsprozessen) wurden auf Hawk berechnet.

Die in Tabelle 4.1c dargestellten Daten bestätigen das zu erwartende Ergebnis deutlich. Mit steigendem Vergrößerungsfaktor wird eine erhöhte Anzahl an Iterationen des Verfahrens benötigt, damit MGRIT konvergiert. Je größer der Vergrößerungsfaktor, desto weiter ist die Schrittweite des Zeitschrittverfahrens auf dem groben Gitterlevel. Dadurch ist die Approximation des exakten Zeitschrittverfahrens ungenauer und es werden mehr Korrekturen benötigt, um die erwünschte Konvergenztoleranz zu erreichen.

Dies lässt sich abgeschwächt auch für *Multidomain21* und *Diffusion* feststellen. Im Gegensatz zu *Multidomain149* ist der Unterschied jedoch nicht so groß. Bei *Multidomain149* verdoppelt sich die Anzahl an Iterationen für einen zehnfachen Vergrößerung des Zeitschrittes auf dem gröberen Gitter. Dies ist für die anderen beiden Testprobleme nicht der Fall. Die Anzahl der benötigten Iterationen steigt bei diesen nur minimal.

Die Tendenz der Anzahl der Iterationen entspricht jedoch nicht der Tendenz der benötigten Rechenzeit. Trotz erhöhter Iterationsanzahl konvergiert das Verfahren zum Teil in kürzerer Zeit. Auch dieses Ergebnis ist zu erwarten. Je größer das Gitter, desto weniger Zeitschritte müssen berechnet werden und desto schneller wird die Grobgitterkorrektur bestimmt. Das bedeutet, dass die benötigte Rechenzeit pro Iteration reduziert wird. Dies wird unter anderem in Tabelle 4.1b deutlich. Für die meisten betrachteten Vergrößerungsfaktoren benötigt *Multidomain21* vier Iterationen. Trotz dieser konstanten Anzahl an Iterationen wird die Laufzeit mit steigendem Vergrößerungsfaktor geringer.

Diffusion und *Multidomain21* weisen sehr ähnliches Verhalten auf. Die Größe des Vergrößerungsfaktors hat keinen starken Einfluss auf die Konvergenzrate MGRITs. Daher ist ein großer Vergrößerungsfaktor eine geeignete Wahl. Dies gilt nicht für *Multidomain149*. Durch die höhere Auflösung im Ort müssen sehr steife Differentialgleichungen gelöst werden. Dadurch wird durch nur geringfügige Erhöhung der Schrittweite auf einem gröberen Gitter die Approximation erheblich schlechter.

Bis hierhin wurde die Zwei-Level MGRIT Methode betrachtet. Der Nachteil der Zwei-Level Methode ist der aufwendige, in der Zeit sequentielle Lösungsvorgang auf dem groben Gitter. Dies wird durch die Einführung einer Zeitgitterhierarchie vermieden. Anstatt auf dem zweiten Level die Grobgitterkorrektur sequentiell zu berechnen, wird das Verfahren rekursiv angewendet.

Die Wahl der Vergrößerungsfaktoren muss jedoch abgewogen werden. Sehr kleine Faktoren wie $m = 2$ sind aus Gründen des benötigten Speicherplatzes auf dem feinen Level nicht tragbar. Außerdem führt ein größerer Vergrößerungsfaktor zu günstigeren Rechenkosten auf den gröberen Gittern. Wird jedoch ein großer Wert für m gewählt, so wird die Relaxation sequentieller. Große Faktoren führen zu großen Subintervallen; dies hat zur Folge, dass viele F-Relaxationsschritte, welche sequentiell auf dem jeweiligen Subintervall berechnet werden, durchgeführt werden müssen. Dies kann den positiven Effekt der Zeitparallelisierung mit vielen Prozessen in der Zeit limitieren.

Das Ziel ist demnach eine Wahl, welche einen Kompromiss zwischen den beiden Ansätzen eingeht. Friedhoff (2014) schlägt vor, auf dem feinen Zeitgitter einen relativ großen und auf den restlichen Gittern einen kleinen Vergrößerungsfaktor zu verwenden. Dadurch wird auf dem Gitterlevel 0, welches den größten Teil des zusätzlich benötigten Speicherplatzes ausmacht, die Anzahl der C-Punkte und damit das Speichern zusätzlicher Vektoren reduziert. Außerdem ist die Anzahl der Zeitschritte auf dem feinen Gitter für den in dieser Arbeit betrachteten Anwendungsfall so hoch, dass selbst mit einem größeren Vergrößerungsfaktor die Parallelität der Relaxation erhalten bleibt. Auf den groben, kleineren Gittern würde ein großer Faktor die mögliche Parallelität einschränken. Daher ist auf diesen Zeitgittern ein kleiner Faktor die bessere Wahl.

Aus diesen Gründen wird von nun an in dieser Arbeit das MGRIT Verfahren mit $m^{(0)} = 16$ auf dem feinen Gitter und $m^{(l)} = 2, l > 0$ auf allen anderen Gittern betrachtet. Die Anzahl der Zeitgitter der verwendeten Gitterhierarchie ist so gewählt, dass das größte Gitterlevel L minimal viele Zeitpunkte beinhaltet (jedoch mindestens drei). Es wird angemerkt, dass diese Strategie weiter optimiert werden kann. Eine aggressivere Vergrößerung auf dem feinen Level oder nach und nach kleiner werdende Vergrößerungsfaktoren können die Laufzeit weiter reduzieren. Dies muss jedoch problemspezifisch angepasst werden und führt nur zu geringen Verbesserungen.

4.2.2 Einfluss der Relaxationsmethode

Ein weiterer wichtiger Parameter des MGRIT Verfahrens ist die Relaxationsmethode. Im Parareal Verfahren wird pro Glättung einmalig die F-Relaxation angewendet. Motiviert durch Mehrgitter Reduktionsverfahren verwendet MGRIT FCF-Relaxation. Durch die aufwendigere Relaxationsmethode werden weniger Iterationen benötigt; außerdem stellt MGRIT mit FCF-Relaxation in vielen Fällen eine stabilere Methode dar (Falgout et al., 2017; Friedhoff, 2014). Zusätzlich wird in dieser Arbeit die F-FCF-Relaxation betrachtet; da die Anwendung der FCF-Relaxation teurer als die der F-Relaxation ist, wird in dieser Strategie auf dem großen Gitter die günstigere F-Relaxation durchgeführt und auf allen anderen Zeitgittern die FCF-Relaxation.

Die drei unterschiedlichen Relaxationsstrategien werden in Tabelle 4.2 näher untersucht. Zu sehen sind die Anzahl der Iterationen und die Rechenzeit MGRITs mit verschiedenen Relaxationsmethoden angewandt auf die drei Testprobleme für verschieden große Zeitgitter.

$N^{(0)}$	Relaxation	#Iter	Dauer[s]
111	F	2	<1
1000	F	20	2
4000	F	27	10
111	F-FCF	2	<1
1000	F-FCF	14	2
4000	F-FCF	14	5
111	FCF	2	<1
1000	FCF	8	1
4000	FCF	9	5

(a) *Diffusion*

$N^{(0)}$	Relaxation	#Iter	Dauer[s]
111	F	4	7
1000	F	5	59
4000	F	6	266
111	F-FCF	3	6
1000	F-FCF	4	49
4000	F-FCF	5	247
111	FCF	3	7
1000	FCF	4	58
4000	FCF	5	289

(b) *Multidomain21*

$N^{(0)}$	Relaxation	#Iter	Dauer[s]
111	F	7	170
1000	F	15	384
4000	F	div	-
111	F-FCF	4	40
1000	F-FCF	12	313
4000	F-FCF	19	2890
111	FCF	4	4
1000	FCF	10	399
4000	FCF	16	3295

(c) *Multidomain149*

Tabelle 4.2: Laufzeit und Anzahl Iterationen der MGRIT Methode mit FCF-Relaxation für verschieden große Anzahl an Zeitschritten $N^{(0)}$ und verschiedenen Relaxationsstrategien. Anzahl der Gitter ist maximal gewählt; es wurde Vergrößerungsfaktor 16 auf dem ersten Gitter und 2 auf allen anderen Gittern verwendet. Alle Testläufe wurden auf Helium mit 24 Zeitprozessen mit jeweils einem Ortsprozess durchgeführt.

Tabelle 4.2 zeigt, dass MGRIT mit F-Relaxation gegenüber der FCF-Relaxation mehr Iterationen benötigt. Es ist jedoch auch sichtbar, dass die Rechenzeit pro Iteration geringer ist. Dies folgt auf natürlicher Weise aus dem Aufbau der Relaxationsstrategien.

In Tabelle 4.2 wird deutlich, weshalb die FCF-Relaxation MGRITs wichtig ist und eine naive Erweiterung des Parareal Verfahrens auf mehrere Gitterlevel nicht zu empfehlen ist. Die benötigte Iterationsanzahl von *Diffusion* und *Multidomain149* steigt für größer werdende Zeitgitter stark an, wenn MGRIT mit F-Relaxation verwendet wird. Das sehr steife *Multidomain149* divergiert sogar

mit F-Relaxation. Mit Hilfe der FCF-Relaxation kann dieses Phänomen eingedämmt werden. Für *Diffusion* mit FCF-Relaxation steigt die Iterationsanzahl nur leicht an, sodass MGRIT auch für große Probleme effizient anwendbar ist.

Für *Multidomain21* macht die Wahl der Relaxation keinen großen Unterschied. Die Anzahl der Iterationen steigt nur leicht an, unabhängig welche Relaxation angewendet wird. Dies folgt durch die Wahl des Grobgitteroperators. Da die Schrittweite des expliziten Verfahrens trotz größerem Gitter nicht verändert wird und die geringere Auflösung im Ort betrachtet wird, ist die Güte der Approximation der Zeitschrittverfahren auf den groben Gittern gut.

Für *Multidomain149* ist dies nicht der Fall. Durch die höhere Auflösung im Ort wird die Ortsgitterweite kleiner und damit die Courant-Friedrichs-Lewy-Zahl des Verfahrens größer. Das Heun Verfahren hat dadurch Schwierigkeiten, die steiferen Differentialgleichungen schnell und genau zu lösen. Dies beeinflusst das Verhalten MGRITs maßgeblich. Friedhoff (2014) präsentiert Testergebnisse, die zeigen, dass MGRIT für explizite Verfahren mit Diskretisierungen mit großer Courant-Friedrichs-Lewy-Zahl erheblich mehr Iterationen benötigt. In dem hier betrachteten Anwendungsfall führt dies zu Divergenz ab einer geringen Anzahl an Zeitschritten. Dadurch ist die Anwendung MGRITs auf *Multidomain149* wenig praktikabel und wird deswegen nicht mehr betrachtet. Um eine solche Ortsauflösung mit MGRIT zu behandeln, müssen kleinere Schrittweiten des expliziten Verfahrens gewählt werden.

Dieselbe Eigenschaft wie die FCF-Relaxation, dass die Iterationsanzahl nur leicht steigt, lässt sich für die F-FCF-Relaxation beobachten. Der Unterschied zur FCF-Relaxation kann durch den Vergleich der benötigten Rechenzeiten erkannt werden. F-FCF-Relaxation spart Rechenarbeit auf dem feinen, großen Zeitgitter, ohne die Konvergenzschwierigkeiten für große Probleme der F-Relaxation vorzuweisen. Daher kann festgehalten werden, dass für die schnelle Lösung des Multidomain Problems die F-FCF-Relaxationsmethode am besten geeignet ist.

4.2.3 Einfluss der Zyklen

Zu guter Letzt wird die Wahl des Zyklus diskutiert. In XBraid ist die Nutzung des V-Zyklus sowie des F-Zyklus ermöglicht. Der F-Zyklus ist der aufwendigere der beiden Zyklen; anstatt den Korrekturterm möglichst schnell zum feinsten Gitterlevel zu transportieren, werden nach und nach größere V-Zyklen angewandt (siehe Abbildung 2.4). Dies ähnelt einer Relaxationsstrategie.

Tatsächlich verhält sich MGRIT mit F-Zyklus und F-Relaxation sehr ähnlich zu MGRIT mit V-Zyklus und FCF-Relaxation. Die Zahl der benötigten Iterationen bleibt trotz F-Relaxation niedrig. Dies folgt aus der zusätzlichen Rechenarbeit des F-Zyklus. Die geringe Glättung der F-Relaxation wird durch die zusätzliche Anwendung von V-Zyklen verbessert. Daher wird der aufwendigere F-Zyklus im Regelfall in Kombination mit dem günstigeren F-Relaxationsverfahren ausgeführt.

Angewendet auf das in dieser Arbeit betrachtete Multidomain Problem zeigen Testergebnisse jedoch keine Vorteile des F-Zyklus gegenüber dem V-Zyklus. In keinem Testfall ermöglichte das Verfahren mit dem F-Zyklus schnellere Rechenzeiten als die entsprechende Methode mit V-Zyklus. Daher wird auf das MGRIT Verfahren mit F-Zyklus nicht näher eingegangen.

4.3 Skalierung und Laufzeit

Bis hierhin wurde ausschließlich der Einfluss einzelner Parameter auf das MGRIT Verfahren untersucht. Eine der wichtigsten Eigenschaften eines parallelen Verfahrens ist die Güte der Skalierung der Methode. Dies wird in diesem Kapitel betrachtet und analysiert; außerdem wird untersucht, ob mit der Anwendung MGRITs eine Beschleunigung der Simulation des neuromuskulären Systems möglich ist.

Es wird die starke Skalierung der beiden Testproblemstellungen *Diffusion* und *Multidomain21* betrachtet. Dazu wird die benötigte Rechenzeit des Lösungsvorgangs unter der Verwendung einer steigenden Anzahl an Rechenkernen gemessen; die Problemgröße mit 10000 Zeitschritten und 6174 Unbekannten bleibt unverändert. Die Testläufe wurden auf Helium durchgeführt; alle der maximal 24 Rechenkerne wurden für die Parallelisierung in der Zeit verwendet.

Es werden zwei verschiedene Varianten MGRITs untersucht. Zum einen das Multi-Level MGRIT Verfahren mit den in Abschnitt 4.2 beschriebenen Parametern: es werden acht Gittern verwendet, sodass das größte Gitter vier Zeitpunkte beinhaltet. Auf dem feinen Gitter wird der Vergrößerungsfaktor 16 und F-Relaxation angewendet; auf allen anderen Zeitgittern wird der Vergrößerungsfaktor 2 und FCF-Relaxation angewendet. Dies wird verglichen mit der Zwei-Level MGRIT Methode mit Vergrößerungsfaktor 16 und F-Relaxation. Außerdem wird die Rechenzeit des sequentiellen Verfahrens mit einem Prozess im Ort dargestellt; der entsprechende Graph wird unabhängig von der X-Achse der Abbildungen betrachtet.

In Abbildung 4.2 sind die Ergebnisse vom Referenzproblem *Diffusion* dargestellt. Wie zu erwarten konvergiert die Multi-Level Methode ab einer bestimmten Anzahl an Zeitprozessen in kürzerer Zeit als die Zwei-Level Methode. Wird die logarithmische Darstellung in Abbildung 4.2b betrachtet, fällt die sehr gute Skalierung der Multi-Level MGRIT Methode auf. Bei einer Verdopplung der verwendeten Rechenkerne verringert sich die benötigte Rechenzeit um einen Faktor nur sehr knapp größer als 0.5. Dies zeigt, dass Multi-Level MGRIT angewandt auf *Diffusion* das Potential hat, auch für sehr große Simulationen Laufzeitsenkungen durch Parallelisierung in der Zeit zu erwirken.

Dieselben Ergebnisse können für MGRIT angewandt auf *Multidomain21* nicht beobachtet werden, siehe Abbildung 4.3. Zum einen benötigt die Zwei-Level Methode erheblich weniger Rechenzeit als die Multilevelmethode, unabhängig von der hier betrachteten Anzahl an Zeitprozessen. Zum anderen skalieren beide Varianten erheblich schlechter als die Verfahren angewandt auf die eindimensionale Wärmeleitungsgleichung. Bei einer Vervielfachung der verwendeten Rechenkerne verringert sich die benötigte Rechenzeit um einen Faktor nur sehr knapp niedriger als 0.5.

4.3.1 Analyse

Die Anwendung MGRITs auf *Multidomain21* weist somit sehr viel schlechtere Skalierungseigenschaften als die Anwendung MGRITs auf *Diffusion* auf. Dies liegt an der Wahl der Grobgitteroperatoren. Tabelle 4.3 stellt die Laufzeiten der jeweiligen Zeitschrittverfahren passend zu unterschiedlich groben Zeitgittern der Multi-Level MGRIT Methode aus Abbildung 4.2 und Abbildung 4.3 dar. Präsentiert wird die benötigte Rechenzeit der Lösungsverfahren, um die Modellprobleme auf dem Zeitintervall $[0, 30]$ zu lösen.

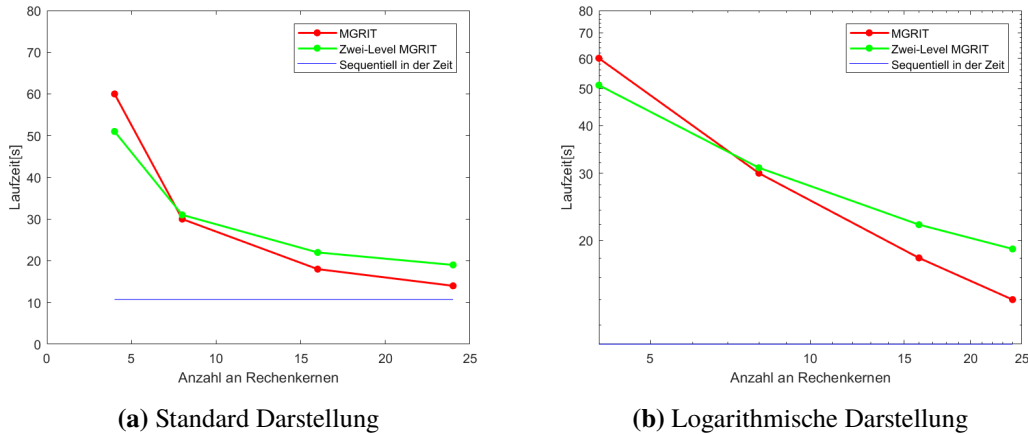


Abbildung 4.2: Starke Skalierung MGRITs auf *Diffusion*(10000 Zeitschritte) angewandt. Alle Testläufe wurden auf Helium mit jeweils einem Rechenkern im Ort durchgeführt. Rot: Anzahl der Gitter ist maximal gewählt; es wurde Vergrößerungsfaktor 16 und F-Relaxation auf dem ersten Gitter sowie Vergrößerungsfaktor 2 und FCF-Relaxation auf allen anderen Gittern verwendet. Grün: Zwei-Level MGRIT mit F-Relaxation und Vergrößerungsfaktor 16. Blau: Laufzeit des sequentiellen Verfahrens mit einem Rechenkern (unabhängig von der X-Achse).

δt	Dauer[s]
0.003	10.38
$16 \cdot 0.003$	1.06
$32 \cdot 0.003$	0.65
$64 \cdot 0.003$	0.33
$128 \cdot 0.003$	0.17
$258 \cdot 0.003$	0.08

(a) *Diffusion*

δt	Dauer[s]
0.003	257
$16 \cdot 0.003$	50
$32 \cdot 0.003$	43
$64 \cdot 0.003$	40
$128 \cdot 0.003$	37
$258 \cdot 0.003$	36

(b) *Multidomain21*

Tabelle 4.3: Laufzeit des feinen Zeitschrittoperators sowie Laufzeit der groben Zeitschrittoperatoren auf dem Intervall $[0, 30]$. Alle Testläufe wurden auf Helium mit jeweils einem Rechenkern im Ort durchgeführt.

Der Unterschied zwischen beiden Testproblemstellungen ist klar zu erkennen. Während die Kosten der Grobgitteroperatoren für *Diffusion* ungefähr entsprechend dem jeweiligen Vergrößerungsfaktor immer geringer werden, stagnieren die Rechenkosten für *Multidomain21*. Die Differenz der Laufzeiten zwischen dem feinen und dem nächst gröberen Gitter ist noch relativ hoch, doch für alle weiteren Gittern ist der Unterschied der Rechenkosten sehr gering. Dies ist dem gewählten Aufbau der Grobgitteroperatoren zuzuschreiben. Je weiter die Zeitschritte des Multidomain Lösers werden, desto dominanter wird die benötigte Rechenzeit des Heun Verfahrens, dessen Zeitschrittweite aus Stabilitätsgründen konstant bleibt. Dadurch haben die zusätzlich verwendeten Zeitgitter nicht denselben Effekt wie für *Diffusion*. Dieser Aufbau der Grobgitteroperatoren bringt zwei Vorteile mit sich: das Verfahren ist für nicht extreme Verhältnisse zwischen Zeitschrittweite und

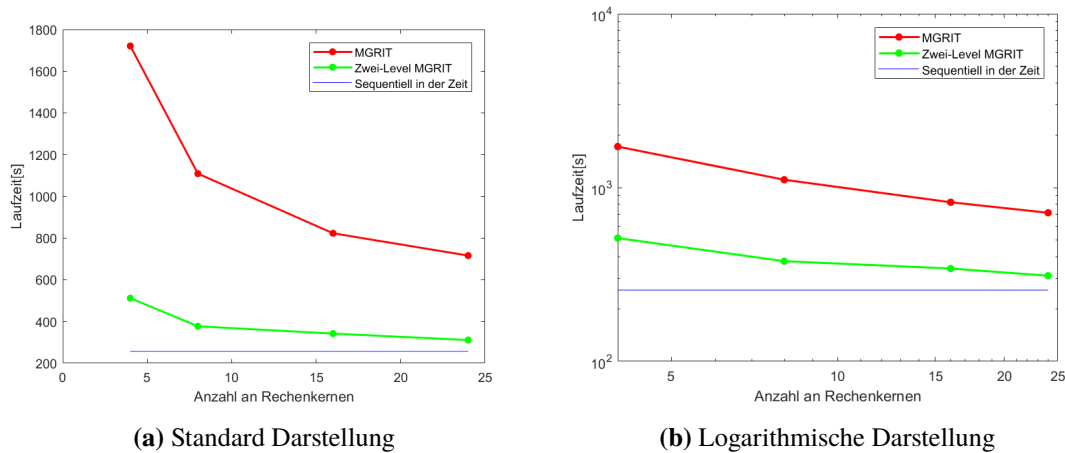


Abbildung 4.3: Starke Skalierung MGRITs auf *Multidomain21* (10000 Zeitschritte) angewandt. Alle Testläufe wurden auf Helium mit jeweils einem Rechenkern im Ort durchgeführt. Rot: Anzahl der Gitter ist maximal gewählt; es wurde Vergrößerungsfaktor 16 und F-Relaxation auf dem ersten Gitter sowie Vergrößerungsfaktor 2 und FCF-Relaxation auf allen anderen Gittern verwendet. Grün: Zwei-Level MGRIT mit F-Relaxation und Vergrößerungsfaktor 16. Blau: Laufzeit des sequentiellen Verfahrens mit einem Rechenkern (unabhängig von der X-Achse).

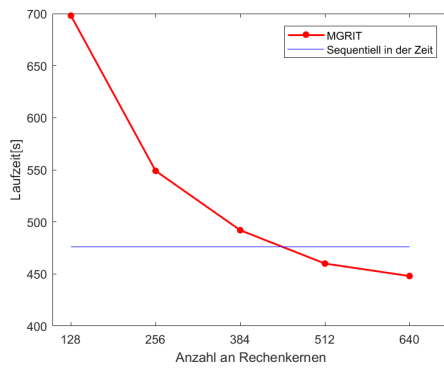
Ortsauflösung stabil. Außerdem übt die Anzahl der simulierten Zeitschritte nur einen geringen Einfluss auf die Konvergenzrate MGRITs aus. Doch die hohen Kosten der Grobgitteroperatoren sind ein entscheidender Nachteil.

4.3.2 Laufzeitsenkung gegenüber des sequentiellen Verfahrens

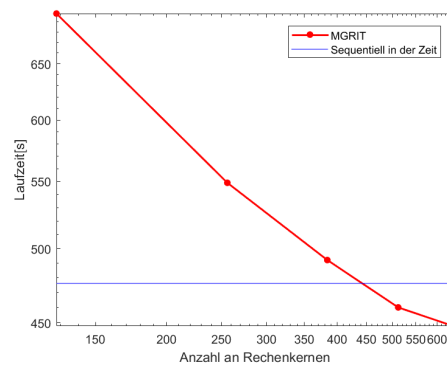
Zum Abschluss der Untersuchungen wird eine weitere starke Skalierung MGRITs präsentiert. Diese wurde auf Hawk mit einem größeren Problem durchgeführt. Analog zu *Multidomain21* wird das neuromuskuläre System für 49 Fasern mit jeweils 21 Diskretisierungspunkten betrachtet. In dieser Studie werden jedoch 20 motorische Einheiten und 40000 Zeitschritte simuliert. Dies entspricht einer 0.12 Sekunden langen Simulation elektronischer Potentiale eines Muskels. Bei der Anwendung der Zwei-Level MGRIT Methode mit F-Relaxation und Vergrößerungsfaktor 16 wurden 16 Rechenkerne im Ort pro Zeitprozess verwendet.

Abbildung 4.4 zeigt, dass eine Beschleunigung der Simulation des neuromuskulären Systems durch Zeitparallelisierung möglich ist. Auf Grund der verwendeten 16 Rechenkerne im Ort in Kombination mit der guten Skalierung im Ort des Heun Verfahrens (siehe Abbildung 4.1 ist eine Senkung der Laufzeit gegenüber der sequentiellen Methode ab 32 Zeitprozessen erreicht.

4 Numerische Tests



(a) Standard Darstellung



(b) Logarithmische Darstellung

Abbildung 4.4: Starke Skalierung Zwei-Level MGRITs auf *Multidomain21* (10000 Zeitschritte) mit 20 motorischen Einheiten angewandt. Alle Testläufe wurden auf Hawk mit jeweils 16 Rechenkernen im Ort durchgeführt. Rot: Zwei-Level MGRIT mit F-Relaxation und Vergrößerungsfaktor 16. Blau: Laufzeit des sequentiellen Verfahrens mit 16 Rechenkernen (unabhängig von der X-Achse).

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde die Anwendung des „Multigrid Reduction in Time“ Verfahrens auf die Simulation des neuromuskulären Systems untersucht, um durch zusätzliche Zeitparallelisierung das sehr umfangreiche Differentialgleichungssystem mit entsprechenden Hochleistungsrechnern in kürzerer Zeit zu lösen. Dazu wurden geeignete Parametereinstellungen bezüglich des Vergrößerungsfaktors, der Relaxation und des Zyklus bestimmt. Dabei wurde festgestellt, dass für hohe Ortsauflösungen entsprechend kleine Zeitschrittweiten des expliziten Lösers gewählt werden müssen, da sonst MGRIT divergiert. Außerdem wurde die Skalierung des zeitparallelen Verfahrens untersucht. Es stellte sich heraus, dass MGRIT mit der hier präsentierten Aufbau des Grobgitteroperators, welcher aus Gründen der Stabilität gewählt wurde, eine nicht so gute Skalierung wie bei der Anwendung auf das Referenzproblem vorweist. Es konnte eine Laufzeitsenkung gegenüber des in der Zeit sequentiellen Lösungsverfahrens erreicht werden, die allerdings wegen der schlechteren Skalierung geringer als erwartet ausfiel.

Eine weitergehende mögliche Verbesserung der Einführung der Zeitparallelität für die Simulation des neuromuskulären Systems könnte durch eine Anpassung des Grobgitteroperators des MGRIT Verfahrens erreicht werden. Ein Vorschlag ist die Nutzung numerischer Verfahren geringerer Ordnung. Dadurch werden trotz Beibehaltung der Zeitschrittweite des expliziten Lösungsverfahrens und damit der Stabilität des Verfahrens die Rechenkosten des Zeitschrittverfahrens auf den gröberen Gittern gesenkt und voraussichtlich eine bessere Skalierung ermöglicht.

Literaturverzeichnis

- Axelsson, A. O. H., J. G. Verwer (1985). „Boundary Value Techniques for Initial Value Problems in Ordinary Differential Equations“. In: *Mathematics of Computation* 45.171, S. 153–171. ISSN: 00255718, 10886842.
- Bakhvalov, N. (1966). „On the convergence of a relaxation method with natural constraints on the elliptic operator“. In:
- Bellen, A., M. Zennaro (1989). „Parallel algorithms for initial-value problems for difference and differential equations“. In: *Journal of Computational and Applied Mathematics* 25.3, S. 341–350. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0377-0427\(89\)90037-X](https://doi.org/10.1016/0377-0427(89)90037-X).
- Bennequin, D., M. Gander, L. Halpern (Jan. 2009). „A Homographic Best Approximation Problem with Application to Optimized Schwarz Waveform Relaxation“. In: *Math. Comput.* 78, S. 185–223. DOI: [10.1090/S0025-5718-08-02145-5](https://doi.org/10.1090/S0025-5718-08-02145-5).
- Bolten, M., D. Moser, R. Speck (März 2016). „A multigrid perspective on the parallel full approximation scheme in space and time“. In: *Numerical Linear Algebra with Applications*. DOI: [10.1002/nla.2110](https://doi.org/10.1002/nla.2110).
- Bradley, C. P., N. Emyam, T. Ertl, D. Göddeke, A. Hessenthaler, T. Klotz, A. Krämer, M. Krone, B. Maier, M. Mehl, T. Rau, O. Röhrle (2018). „Enabling Detailed, Biophysics-Based Skeletal Muscle Models on HPC Systems“. In: *Frontiers in Physiology* 9, S. 816. ISSN: 1664-042X. DOI: [10.3389/fphys.2018.00816](https://doi.org/10.3389/fphys.2018.00816).
- Braess, D. (1984). „The Coverage Rate of a Multigrid Method with Gauss-Seidel Relaxation for the Poisson Equation“. In: *Mathematics of Computation* 42.166, S. 505–519. ISSN: 00255718, 10886842.
- Brandt, D. (1977). „Multi-level adaptive solutions to boundary-value problems math comptr“. In: Briggs, W. L., V. E. Henson, S. F. McCormick (2000). *A Multigrid Tutorial (2nd Ed.)* USA: Society for Industrial und Applied Mathematics. ISBN: 0898714621.
- Brugnano, L., D. Trigiante (Nov. 1993). „A Parallel Preconditioning Technique for Boundary Value Methods“. In: *Appl. Numer. Math.* 13.4, S. 277–290. ISSN: 0168-9274. DOI: [10.1016/0168-9274\(93\)90002-9](https://doi.org/10.1016/0168-9274(93)90002-9).
- (1996). „Convergence and Stability of Boundary Value Methods for Ordinary Differential Equations“. In: *Proceedings of the 6th International Congress on Computational and Applied Mathematics*. Leuven, Belgium: Elsevier Science Publishers B. V., S. 97–109.
- (1998). *Solving Differential Equations by Multistep Initial and Boundary Value Methods*. Stability and Control: Theory, Methods and Applications. Taylor & Francis. ISBN: 9789056991074.
- Bulin, J. (2013). „Large-scale time parallelization for molecular dynamics problems“. Diss.
- Buneman, O. (Jan. 1969). „COMPACT NON-ITERATIVE POISSON SOLVER.“ In:
- Charney, J. (1948). *On the Scale of Atmospheric Motions*. Geofysiske publikasjoner. Grøndahl & søns boktr., I kommission hos Cammermeyer.
- Chartier, P., B. Philippe (Sep. 1993). „A parallel shooting technique for solving dissipative ODE’s“. In: *Computing* 51.3-4, S. 209–236. DOI: [10.1007/bf02238534](https://doi.org/10.1007/bf02238534).

- Davis, T. A. (Juni 2004). „Algorithm 832: UMFPACK V4.3—an Unsymmetric-Pattern Multifrontal Method“. In: *ACM Trans. Math. Softw.* 30.2, S. 196–199. ISSN: 0098-3500. DOI: [10.1145/992200.992206](https://doi.org/10.1145/992200.992206).
- De Sterck, H., T. A. Manteuffel, S. F. McCormick, L. Olson (2004). „Least-Squares Finite Element Methods and Algebraic Multigrid Solvers for Linear Hyperbolic PDEs“. In: *SIAM Journal on Scientific Computing* 26.1, S. 31–54. DOI: [10.1137/S106482750240858X](https://doi.org/10.1137/S106482750240858X).
- DESHPANDE, A., S. MALHOTRA, M. H. SCHULTZ, C. C. DOUGLAS (1995). „A RIGOROUS ANALYSIS OF TIME DOMAIN PARALLELISM“. In: *Parallel Algorithms and Applications* 6.1, S. 53–62. DOI: [10.1080/10637199508915498](https://doi.org/10.1080/10637199508915498).
- Dobrev, V. A., T. Kolev, N. A. Petersson, J. B. Schroder (Jan. 2017). „Two-Level Convergence Theory for Multigrid Reduction in Time (MGRIT)“. In: *SIAM Journal on Scientific Computing* 39.5, S501–S527. DOI: [10.1137/16m1074096](https://doi.org/10.1137/16m1074096).
- Douglas, C., I. Kim, H. Lee, D. Sheen (Sep. 2011). „Higher-order schemes for the Laplace transformation method for parabolic problems“. In: *Computing and Visualization in Science* 14, S. 39–47. DOI: [10.1007/s00791-011-0156-6](https://doi.org/10.1007/s00791-011-0156-6).
- Duarte, M., M. Massot, S. Descombes (Okt. 2011). „Parareal operator splitting techniques for multi-scale reaction waves: Numerical analysis and strategies“. In: *ESAIM Mathematical Modelling and Numerical Analysis* 45, S. 825–852. DOI: [10.1051/m2an/2010104](https://doi.org/10.1051/m2an/2010104).
- Dutt, A., L. Greengard, V. Rokhlin (Juni 2000). „Spectral Deferred Correction Methods for Ordinary Differential Equations“. In: *BIT* 40, S. 241–266. DOI: [10.1023/A:1022338906936](https://doi.org/10.1023/A:1022338906936).
- Emmett, M., M. Minion (Jan. 2014). „Efficient Implementation of a Multi-Level Parallel in Time Algorithm“. In: *Lecture Notes in Computational Science and Engineering* 98, S. 359–366. DOI: [10.1007/978-3-319-05789-7_33](https://doi.org/10.1007/978-3-319-05789-7_33).
- Emmett, M., M. L. Minion (2012). „Toward an Efficient Parallel in Time Method for Partial Differential Equations“. In: *Communications in Applied Mathematics and Computational Science* 7, S. 105–132. DOI: [10.2140/camcos.2012.7.105](https://doi.org/10.2140/camcos.2012.7.105).
- Falgout, R., T. Manteuffel, B. O’Neill, J. Schroder (Jan. 2017). „Multigrid Reduction in Time for Nonlinear Parabolic Problems: A Case Study“. In: *SIAM Journal on Scientific Computing* 39, S298–S322. DOI: [10.1137/16M1082330](https://doi.org/10.1137/16M1082330).
- Farhat, C., M. Chandesris (2003). „Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications“. In: *International Journal for Numerical Methods in Engineering* 58.9, S. 1397–1434. DOI: [10.1002/nme.860](https://doi.org/10.1002/nme.860).
- Farina, D., D. F. Stegeman, R. Merletti (2016). „Biophysics of the Generation of EMG Signals“. In: *Surface Electromyography : Physiology, Engineering, and Applications*. John Wiley & Sons, Ltd. Kap. 2, S. 1–24. ISBN: 9781119082934. DOI: [10.1002/9781119082934.ch02](https://doi.org/10.1002/9781119082934.ch02).
- Fedorenko, R. (Jan. 1962). „A relaxation method for solving elliptic difference equations“. In: *USSR Computational Mathematics and Mathematical Physics* 1.4, S. 1092–1096. DOI: [10.1016/0041-5553\(62\)90031-9](https://doi.org/10.1016/0041-5553(62)90031-9).
- Foerster, H., K. Stüben, U. Trottenberg (1981). „NON-STANDARD MULTIGRID TECHNIQUES USING CHECKERED RELAXATION AND INTERMEDIATE GRIDS“. In: *Elliptic Problem Solvers*. Elsevier, S. 285–300. DOI: [10.1016/b978-0-12-632620-8.50027-9](https://doi.org/10.1016/b978-0-12-632620-8.50027-9).
- Frankel, S. P. (1950). „Convergence Rates of Iterative Treatments of Partial Differential Equations“. In: *Mathematical Tables and Other Aids to Computation* 4.30, S. 65–75. ISSN: 08916837. URL: <http://www.jstor.org/stable/2002770>.
- Franklin, M. A. (Mai 1978). „Parallel Solution of Ordinary Differential Equations“. In: *IEEE Trans. Comput.* 27.5, S. 413–420. ISSN: 0018-9340. DOI: [10.1109/TC.1978.1675121](https://doi.org/10.1109/TC.1978.1675121).

- Friedhoff, S., R. D. Falgout, T. V. Kolev, S. P. MacLachlan, J. B. Schroder (2013). „A Multigrid-in-Time Algorithm for Solving Evolution Equations in Parallel“. In: *Presented at: Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, United States, Mar 17 - Mar 22, 2013*.
- Friedhoff, S. (2014). „Design and analysis of multigrid methods for parabolic problems“. dissertation. Tufts University.
- Gander, M. (Jan. 2015). „50 Years of Time Parallel Time Integration“. In: Bd. 9, S. 69–113. DOI: [10.1007/978-3-319-23321-5_3](https://doi.org/10.1007/978-3-319-23321-5_3).
- Gander, M. J. (1996). „Overlapping Schwarz for Linear and Nonlinear Parabolic Problems“. eng. In: 9th International Conference on Domain Decomposition Methods. ID: unige:8288, S. 97–104.
- Gander, M. J., E. Hairer (2008). „Nonlinear Convergence Analysis for the Parareal Algorithm“. In: *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, S. 45–56. DOI: [10.1007/978-3-540-75199-1_4](https://doi.org/10.1007/978-3-540-75199-1_4).
- Gander, M. J., S. Vandewalle (Jan. 2007). „Analysis of the Parareal Time-Parallel Time-Integration Method“. In: *SIAM Journal on Scientific Computing* 29.2, S. 556–578. DOI: [10.1137/05064607x](https://doi.org/10.1137/05064607x).
- Gander, M., S. Güttel (März 2013). „PARAEXP: a parallel integrator for linear initial-value problems“. In: *SIAM Journal on Scientific Computing* 35. DOI: [10.1137/110856137](https://doi.org/10.1137/110856137).
- Gander, M., Y.-L. Jiang, R.-J. Li (Jan. 2013). „Parareal Schwarz Waveform Relaxation Methods“. In: *Lecture Notes in Computational Science and Engineering* 91, S. 451–458. DOI: [10.1007/978-3-642-35275-1_53](https://doi.org/10.1007/978-3-642-35275-1_53).
- Göddeke, D. (2010). „Fast and Accurate Finite-Element Multigrid Solvers for PDE Simulations on GPU Clusters“. dissertation. Technischen Universität Dortmund.
- Göddeke, D., R. Strzodka (Jan. 2011). „Cyclic Reduction Tridiagonal Solvers on GPUs Applied to Mixed Precision Multigrid“. In: *IEEE Transactions on Parallel and Distributed Systems* 22.1, S. 22–32. DOI: [10.1109/TPDS.2010.61](https://doi.org/10.1109/TPDS.2010.61).
- Hackbusch, W. (1978). „On the multi-grid method applied to difference equations“. In: *Computing* 20.4, S. 291–306. ISSN: 1436-5057. DOI: [10.1007/BF02252378](https://doi.org/10.1007/BF02252378).
- (Juni 1984). „Parabolic multi-grid methods“. In: S. 189–197.
- (1994). *Iterative Solution of Large Sparse Systems of Equations*. First. Bd. 95. Applied Mathematical Sciences. New York: Springer. DOI: [10.1007/978-1-4612-4288-8](https://doi.org/10.1007/978-1-4612-4288-8).
- Hairer, E., C. Lubich, G. Wanner (Jan. 2006). *Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations. 2nd ed.* Bd. 31. DOI: [10.1007/3-540-30666-8](https://doi.org/10.1007/3-540-30666-8).
- Haut, T., B. Wingate (März 2013). „An Asymptotic Parallel-in-Time Method for Highly Oscillatory PDEs“. In: *SIAM Journal on Scientific Computing* 36. DOI: [10.1137/130914577](https://doi.org/10.1137/130914577).
- Hessenthaler, A., B. S. Southworth, D. Nordsletten, O. Röhrle, R. D. Falgout, J. B. Schroder (Jan. 2020). „Multilevel Convergence Analysis of Multigrid-Reduction-in-Time“. In: *SIAM Journal on Scientific Computing* 42.2, A771–A796. DOI: [10.1137/19m1238812](https://doi.org/10.1137/19m1238812).
- Hestenes, M. R., E. Stiefel (1952). „Methods of conjugate gradients for solving linear systems“. In: *Journal of research of the National Bureau of Standards* 49, S. 409–435.
- Hockney, R. W. (Jan. 1965). „A Fast Direct Solution of Poisson’s Equation Using Fourier Analysis“. In: *J. ACM* 12.1, S. 95–113. ISSN: 0004-5411. DOI: [10.1145/321250.321259](https://doi.org/10.1145/321250.321259).
- Hodgkin, A. L., A. F. Huxley (1952). „A quantitative description of membrane current and its application to conduction and excitation in nerve“. In: *The Journal of Physiology* 117.4, S. 500–544. DOI: [10.1113/jphysiol.1952.sp004764](https://doi.org/10.1113/jphysiol.1952.sp004764).

- Holobar, A., M. Minetto, A. Botter, F. Negro, D. Farina (Feb. 2010). „Experimental Analysis of Accuracy in the Identification of Motor Unit Spike Trains From High-Density Surface EMG“. In: *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* 18, S. 221–9. DOI: [10.1109/TNSRE.2010.2041593](https://doi.org/10.1109/TNSRE.2010.2041593).
- Horton, G., S. Vandewalle (Juli 1995). „A Space-Time Multigrid Method for Parabolic Partial Differential Equations“. In: *SIAM J. Sci. Comput.* 16.4, S. 848–864. ISSN: 1064-8275. DOI: [10.1137/0916050](https://doi.org/10.1137/0916050).
- Horton, G., S. Vandewalle, P. Worley (Mai 1995). „An Algorithm with Polylog Parallel Complexity for Solving Parabolic Partial Differential Equations“. In: *SIAM J. Sci. Comput.* 16.3, S. 531–541. ISSN: 1064-8275. DOI: [10.1137/0916034](https://doi.org/10.1137/0916034).
- Horton, G. (1992). „The time-parallel multigrid method“. In: *Communications in Applied Numerical Methods* 8.9, S. 585–595. DOI: [10.1002/cnm.1630080906](https://doi.org/10.1002/cnm.1630080906).
- Jacobi, C. G. J. (1845). „Ueber eine neue Auflösungsart der bei der Methode der kleinsten Quadrate vorkommenden lineären Gleichungen“. In: *Astronomische Nachrichten* 22.20, S. 297–306. DOI: [10.1002/asna.18450222002](https://doi.org/10.1002/asna.18450222002).
- Keller, H. (1992). *Numerical Methods for Two-point Boundary-value Problems*. Dover books on advanced mathematics. Dover Publications. ISBN: 9780486669250.
- Klotz, T., L. Gizzi, U. Yavuz, O. Röhrle (2019). „Modelling the electrical activity of skeletal muscle tissue using a multi-domain approach“. In: *Biomechanics and Modeling in Mechanobiology* 19, S. 335–349.
- Kwok, F. (Jan. 2014). „Neumann–Neumann Waveform Relaxation for the Time-Dependent Heat Equation“. In: *Lecture Notes in Computational Science and Engineering* 98, S. 189–198. DOI: [10.1007/978-3-319-05789-7_15](https://doi.org/10.1007/978-3-319-05789-7_15).
- Lelarasmee, E., A. E. Ruehli, A. L. Sangiovanni-Vincentelli (1982). „The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits“. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 1.3, S. 131–145.
- Li, X. S. (Sep. 2005). „An Overview of SuperLU: Algorithms, Implementation, and User Interface“. In: *toms* 31.3, S. 302–325.
- Lindelöf, E. (1894). „Sur l’application des méthodes d’approximations successives à l’étude des intégrales réelles des équations différentielles ordinaires.“ French. In: *Journ. de Math. (4)* 10, S. 117–128.
- Lions, J.-L., Y. Maday, G. Turinici (2001). „A ”parareal” in time discretization of PDE’s“. In: *Comptes Rendus de l’Académie des Sciences - Series I - Mathematics* 332, S. 661–668. DOI: [10.1016/S0764-4442\(00\)01793-6](https://doi.org/10.1016/S0764-4442(00)01793-6).
- Lubich, C., A. Ostermann (1987). „Multi-grid dynamic iteration for parabolic equations“. In: *BIT Numerical Mathematics* 27.2, S. 216–234. DOI: [10.1007/BF01934186](https://doi.org/10.1007/BF01934186).
- MacIntosh, B., P. Gardiner, A. McComas (2006). *Skeletal Muscle: Form and Function*. Human Kinetics. ISBN: 9780736045179.
- Maier, B. (o.D.). *Opendihu - Documentation*. URL: <https://opendihu.readthedocs.io/en/latest/>.
- Maier, B., N. Emamy, A. Krämer, M. Mehl (2019). „Highly Parallel Multi-Physics Simulation of Muscular Activation and EMG“. In: *Proceedings of VIII International Conference on Computational Methods for Coupled Problems in Science and Engineering*.
- Mandal, B. (Juni 2012). „A Time-Dependent Dirichlet-Neumann Method for the Heat Equation“. In:
- Merletti R, P. P. (2004). *Electromyography: physiology, engineering, and non-invasive applications*. Wiley-IEEE Press. ISBN: 978-0-471-67580-8.

- Merletti, R., P. Parker (2004). *Electromyography: Physiology, Engineering, and Non-Invasive Applications*. IEEE Press Series on Biomedical Engineering. Wiley. ISBN: 9780471675808.
- Mesin, L. (2013). „Volume conductor models in surface electromyography: Computational techniques“. In: *Computers in biology and medicine* 43 7, S. 942–52.
- Miller, W., D. Geselowitz, C. Res, I. The, N. Heart, W. T. Miller, David, B. Geselowitz (1978). „Simulation studies of the electrocardiogram“. In: *I. The normal heart. Circ Res* 43, S. 301–315.
- Minion, M. (Dez. 2010). „A Hybrid Parareal Spectral Deferred Corrections Method“. In: *Communications in Applied Mathematics and Computational Science* 5. DOI: [10.2140/camcos.2010.5.265](https://doi.org/10.2140/camcos.2010.5.265).
- Minion, M., R. Speck, M. Bolten, M. Emmett, D. Ruprecht (Juli 2014). „Interweaving PFASST and Parallel Multigrid“. In: *SIAM Journal on Scientific Computing* 37. DOI: [10.1137/14097536X](https://doi.org/10.1137/14097536X).
- Miranker, W. L., W. Liniger (1967). „Parallel Methods for the Numerical Integration of Ordinary Differential Equations“. In: *Mathematics of Computation* 21.99, S. 303–320. ISSN: 00255718, 10886842.
- Nievergelt, J. (Dez. 1964). „Parallel Methods for Integrating Ordinary Differential Equations“. In: *Mathematics of Computation* 7.12, S. 731–733. ISSN: 0001-0782. DOI: [10.1145/355588.365137](https://doi.org/10.1145/355588.365137).
- O., R., E. T., G. D., M. B., K. T. Emamy N., Z. L., K. A., R. T., K. M. (2020). *Towards a digital human (DiHu): Ziele und Relevanz*. URL: https://ipvs.informatik.uni-stuttgart.de/SGS/digital_human/research_goals.php (besucht am 20. 07. 2020).
- Oosterlee, C. W., P. Wesseling (1993). „Multigrid Schemes for Time-Dependent Incompressible Navier-Stokes Equations“. In: *IMPACT Comput. Sci. Eng.* 5, S. 153–175.
- Pfeifer, K., L. Vogt (Jan. 2004). „Elektromyographie (EMG)“. In: S. 165–182. ISBN: 978-3-540-62536-0. DOI: [10.1007/978-3-642-18626-4_9](https://doi.org/10.1007/978-3-642-18626-4_9).
- Picard, E. (1893). „Sur l’application des méthodes d’approximations successives à l’étude de certaines équations différentielles ordinaires“. fre. In: *Journal de Mathématiques Pures et Appliquées* 9, S. 217–272.
- Prometeus, G. (o.D.). *Top 500 - The List*. URL: <https://www.top500.org/>.
- Pullan, A., L. Cheng, M. Buist (2005). *Mathematically Modelling the Electrical Activity of the Heart: From Cell to Body Surface and Back Again*. World Scientific. ISBN: 9789812563736.
- Ries, M., U. Trottenberg, G. Winter (Feb. 1983). „A note on MGR methods“. In: *Linear Algebra and its Applications* 49, S. 1–26. DOI: [10.1016/0024-3795\(83\)90091-5](https://doi.org/10.1016/0024-3795(83)90091-5).
- Saad, Y. (Jan. 2003). *Iterative Methods for Sparse Linear Systems*. DOI: [10.1137/1.9780898718003.ch4](https://doi.org/10.1137/1.9780898718003.ch4).
- Saha, P., J. Stadel, S. Tremaine (1997). „A Parallel Integration Method for Solar System Dynamics“. In: *The Astronomical Journal* 114, S. 409.
- Saltz, J., V. Naik (1988). „Towards developing robust algorithms for solving partial differential equations on MIMD machines“. In: *Parallel Comput.* 6, S. 19–44.
- Schröder, J., U. Trottenberg, K. Witsch (1978). „On fast poisson solvers and applications“. In: *Numerical Treatment of Differential Equations*. Springer Berlin Heidelberg, S. 153–187. DOI: [10.1007/bfb0067471](https://doi.org/10.1007/bfb0067471).
- Seidel, P. (1873). *Über ein Verfahren, die Gleichungen, auf welche die Methode der kleinsten Quadrate führt, sowie lineäre Gleichungen überhaupt, durch successive Annäherung aufzulösen*. Abhandlungen der Bayerischen Akademie der Wissenschaften, Mathematisch-Physikalische Klasse. Verlag d. Akad.
- Shampine, L. F., H. A. Watts (1969). „Block Implicit One-Step Methods“. In: *Mathematics of Computation* 23.108, S. 731–740. ISSN: 00255718, 10886842.

- Sheen, D., I. H. Sloan, V. Thomée (2000). „A Parallel Method for Time-Discretization of Parabolic Problems Based on Contour Integral Representation and Quadrature“. In: *Mathematics of Computation* 69.229, S. 177–195. ISSN: 00255718, 10886842.
- (Apr. 2003). „A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature“. In: *IMA Journal of Numerical Analysis* 23.2, S. 269–299. ISSN: 0272-4979. DOI: [10.1093/imanum/23.2.269](https://doi.org/10.1093/imanum/23.2.269).
- Simoens, J., S. Vandewalle (2000). „Waveform Relaxation with Fast Direct Methods as Preconditioner“. In: *SIAM Journal on Scientific Computing* 21.5, S. 1755–1773. DOI: [10.1137/S1064827598338986](https://doi.org/10.1137/S1064827598338986).
- Southworth, B. S. (Jan. 2019). „Necessary Conditions and Tight Two-level Convergence Bounds for Parareal and Multigrid Reduction in Time“. In: *SIAM Journal on Matrix Analysis and Applications* 40.2, S. 564–608. DOI: [10.1137/18m1226208](https://doi.org/10.1137/18m1226208).
- Speck, R. (März 2017). „Parallelizing spectral deferred corrections across the method“. In: *Computing and Visualization in Science* 19. DOI: [10.1007/s00791-018-0298-x](https://doi.org/10.1007/s00791-018-0298-x).
- Sterck, H. D., S. Friedhoff, A. J. M. Howse, S. P. MacLachlan (Nov. 2019). „Convergence analysis for parallel-in-time solution of hyperbolic systems“. In: *Numerical Linear Algebra with Applications* 27.1. DOI: [10.1002/nla.2271](https://doi.org/10.1002/nla.2271).
- Thomée, V. (2005). „A high order parallel method for time discretization of parabolic type equations based on Laplace transformation and quadrature“. In: *International Journal of Numerical Analysis and Modeling* 2, S. 85–96.
- Trottenberg, U., C. Oosterlee, A. Schuller (2000). *Multigrid*. Elsevier Science. ISBN: 9780080479569.
- Van Lent, J., S. Vandewalle (2005). „Multigrid methods for implicit Runge-Kutta and boundary value method discretizations of parabolic PDEs“. English. In: *SIAM Journal on Scientific Computing* 27.1, 67–92 (electronic). ISSN: 1064-8275.
- Vandewalle, S., E. Van de Velde (1994). „Space-time concurrent multigrid waveform relaxation“. In: *Annals of Numerical Mathematics* 1.1-4, S. 347–360. DOI: [10.13140/2.1.1146.1761](https://doi.org/10.13140/2.1.1146.1761).
- Vandewalle, S., G. Horton (1995). „Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods“. In: *Computing* 54.4, S. 317–330. DOI: [10.1007/BF02238230](https://doi.org/10.1007/BF02238230).
- Vandewalle, S., R. Piessens (1992). „Efficient Parallel Algorithms for Solving Initial-Boundary Value and Time-Periodic Parabolic Partial Differential Equations“. In: *SIAM Journal on Scientific and Statistical Computing* 13.6, S. 1330–1346. DOI: [10.1137/0913075](https://doi.org/10.1137/0913075).
- Weinzierl, T., T. Köppl (2012). „A Geometric Space-Time Multigrid Algorithm for the Heat Equation“. In: *Numerical Mathematics: Theory, Methods and Applications* 5.1, S. 110–130. DOI: [10.1017/S1004897900000258](https://doi.org/10.1017/S1004897900000258).
- Wienands, R., W. Joppich (Okt. 2004). *Practical Fourier Analysis for Multigrid Methods*. Chapman und Hall/CRC. DOI: [10.1201/9781420034998](https://doi.org/10.1201/9781420034998).
- Womble, D. (1990). „A Time-Stepping Algorithm for Parallel Computers“. In: *SIAM J. Sci. Comput.* 11, S. 824–837.
- Worley, P. (1991). „Parallelizing Across Time When Solving Time-Dependent Partial Differential Equations“. In: *Proceedings of the Fifth SIAM Conference on Parallel Processing for Scientific Computing*. USA: Society for Industrial und Applied Mathematics, S. 246–252. ISBN: 089871303X.
- XBraid: Parallel multigrid in time* (o.D.). <http://11nl.gov/casc/xbraid>.
- Young, D. (1954). „Iterative Methods for Solving Partial Difference Equations of Elliptic Type“. In: *Transactions of the American Mathematical Society* 76.1, S. 92–111. ISSN: 00029947.
- Zienkiewicz, O., R. Taylor, J. Zhu (2013). *The Finite Element Method: Its Basis and Fundamentals*. The Finite Element Method. Elsevier Science. ISBN: 9780080951355.

Alle URLs wurden zuletzt am 10.09.2020 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift