

Themenübersicht

Seminare, Bachelor- und Masterarbeiten
in den Studiengängen Technomathematik und Mathematik

Jun.-Prof. Dr. Dominik Göddeke

Lehrstuhl für Angewandte Mathematik und Numerik (LS3)
Fachgebiet Hardware-orientierte Numerik für große Systeme
Fakultät für Mathematik, TU Dortmund
dominik.goeddeke@math.tu-dortmund.de

Stand: 21. Februar 2014

Inhaltsverzeichnis

1	Vorwort	3
2	Bachelorseminare	3
3	Bachelorarbeiten und Masterseminare	3
3.1	Effiziente Realisierung der ILU(p) und ILUT(p,tau) Vorkonditionierer	4
3.2	Parallele FSAI-Vorkonditionierung auf CPUs und/oder GPUs	4
3.3	Optimierte Finite-Elemente-Assemblierung durch automatische Codegenerierung	5
3.4	Modellierung des nichtlinearen Dehnungs- und Streckungsverhaltens von Fasern am Beispiel selbst gestrickter Wintermode	5
3.5	Verbesserte SPAI-Vorkonditionierung	6
3.6	MSPAI-Probing	6
3.7	Der SPIKE-Algorithmus	7
3.8	Parareal: Lösung von Anfangswertaufgaben parallel in der Zeit	7
3.9	Deflation: Beschleunigung von Krylov-Unterraum-Verfahren durch Eigenwert-Elimination	8
3.10	Direkte Löser für dünn besetzte lineare Gleichungssysteme	8
3.11	Inner-outer-Methoden	9
3.12	Eigenwert-Schätzer	9
3.13	Vor- und Nachteile von Stabilisierungstechniken für Finite Elemente	10
3.14	Algebraische Mehrgitterverfahren auf GPUs	10
3.15	Umsortierung von unstrukturierten Gittern für die FEM	11
3.16	Mathematik der Visualisierung zeitabhängiger Daten	11
3.17	Effiziente Realisierung von Stencilcodes auf GPUs am Beispiel eines Strömungslösers	12
3.18	Vergleich unterschiedlicher Zeitschrittverfahren für Operator-Splitting Strömungslöser	12
3.19	Mehrgitterverfahren für reine Neumann-Probleme mit Anwendungen in der Strömungssimulation	12
3.20	Erweiterung des Strömungslösers um Energietransport	13
3.21	Erweiterung des Strömungslösers für Turbulenzmodelle	13
4	Masterarbeiten	14
4.1	Mehrgitterverfahren für konvektionsdominante Probleme	14
4.2	Lösung von Anfangswertaufgaben parallel in der Zeit	14
4.3	Inverse Probleme und Wavelet-Kompressionen am Beispiel der Leitfähigkeits-Tomographie	15

4.4	Asynchrone numerische Verfahren	15
4.5	Feingranulare Assemblierung der Galerkin-Grobgitterkorrektur in algebraischen Mehrgitterverfahren	16
5	Abgeschlossene Arbeiten (Auswahl)	17
5.1	Effiziente Sparse-Matrix-Vektor-Multiplikation auf moderner Hardware	17
5.2	IDR(s) und QMRCGStab: Krylov-Unterraum-Verfahren für nicht-SPD-Systeme	17
5.3	Algebraische Mehrgitter-Verfahren	18
5.4	Verbesserte (Block-) Vorkonditionierungstechniken für Probleme der linearisierten Elastizität	18
5.5	Finite-Elemente-Assemblierung auf GPUs	19
5.6	Image Stitching: Mathematische Modellierung der Berechnung von Panoramafotos	19
5.7	Strömungssimulation in Echtzeit auf Smartphones und Tablet-PCs	19
5.8	Die Spektrale Elemente Methode für die Simulation der Ausbreitung seismischer Wellen	19
5.9	Kommunikations-vermeidende Algorithmen im Höchstleistungsrechnen	20
5.10	Optimale Färbungen für parallele Algorithmen	20
	Literatur	21

1 Vorwort

Die Lektüre dieses Dokuments macht nur dann Sinn, wenn Sie vorab meine allgemeinen Hinweise zu Abschlussarbeiten¹ konsultiert haben. Die angebotenen Themen fallen grob und nicht disjunkt in die Kategorien Vorkonditionierung, Krylov-Unterraum-Theorie und lineare Löser, Mehrgitter, parallele Numerik, Finite Elemente, GPU Computing, Höchstleistungsrechnen, und cooles buntes Zeug. Außerdem ist bei den meisten Themen der Programmieranteil durchaus Verhandlungssache. Weder diese Liste noch die Skizzen weiter unten implizieren eine Priorität meinerseits – alles andere wäre ja auch Unfug.

Es existieren reichlich Ideen für weitere Themen, insbesondere im Masterbereich mit Bezug zu aktuellen Forschungsvorhaben. Wenn Sie Interesse haben, sich bspw. in einem der folgenden Bereiche „auszuto-ben“, lassen sich sicherlich konkrete Fragestellungen für Arbeiten in einem Gespräch konstruieren:

- Autotuning
- Gebietszerlegung: FETI-Krylov-Methoden, BPS- und BPX-Methoden
- Anwendungen in der Geophysik

Weiterhin gilt selbstverständlich die Invariante aus dem weiter oben referenzierten Dokument: Wenn ich ein von Ihnen vorgeschlagenes Thema grundsätzlich spannend finde, wenn ich mich in dem Bereich hinreichend auskenne, und wenn das Thema einen für mich akzeptablen Bezug zur Numerik oder zum Wissenschaftlichen Rechnen aufweist, dann betreue ich auch eine Abschlussarbeit dazu.

Aktueller Hinweis: Themen, die speziell auf das gerade abgeschlossene TMABA Studienprojekt zugeschnitten sind, finden sich ab Seite 12.

2 Bachelorseminare

Bachelorseminare betreue ich aktuell typischerweise nur im Kontext des Studienprojekts Technomathematik. Ausnahmen sind auf Anfrage selbstverständlich möglich, jedoch nur, wenn Sie eine meiner Vorlesungen im Bachelor-Bereich besucht haben. Seminararbeiten werden grundsätzlich in die Seminarreihe des Lehrstuhls integriert.

3 Bachelorarbeiten und Masterseminare

Die folgenden Themenvorschläge sind im Umfang auf Bachelorarbeiten zugeschnitten. Fast alle Themen erlauben eine konkrete Ausgestaltung auch für Seminare im Masterstudium, bspw. indem der Umfang der numerischen Tests oder der Implementierung deutlich reduziert wird. Beachten Sie bitte, dass ein Masterseminar ohne Bezug auf eine aufbauende Masterarbeit meist keinen Sinn ergibt – bei einem Gespräch zur Fixierung des Themas sollten wir direkt mögliche Erweiterungen diskutieren. Die Reihenfolge impliziert selbstverständlich keine Priorität. Meine Vorlesungsskripte ([16–18]) werden zur Redundanzminimierung nicht in den jeweiligen Literaturhinweisen aufgeführt.

¹<http://www.mathematik.tu-dortmund.de/~goeddeke/teaching/hinweise-allgemein.pdf>

3.1 Effiziente Realisierung der ILU(p) und ILUT(p,tau) Vorkonditionierer

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*, gute Programmierkenntnisse

Schwerpunkt: effiziente Implementierung, Vorkonditionierung

Zusammenfassung: Nach einer kurzen Zusammenstellung der numerischen und graphentheoretischen Grundlagen sollen in dieser Arbeit effiziente Realisierungen der Vorkonditionierer ILU(p) und ILUT(p,tau) entwickelt, beschrieben, analysiert und implementiert werden. Die Diskretisierung für Modellprobleme kann wahlweise mit Finiten Differenzen oder Finiten Elementen erfolgen. Die Implementierung soll für ein geeignetes Krylov-Unterraum-Verfahren numerisch validiert werden, indem für ein exemplarisches Modellproblem aus den Kategorien anisotrope Diffusion und Konvektion-Diffusion-Reaktion der Einfluss der beiden Parameter p und τ auf die Konvergenzgeschwindigkeit betrachtet wird. Ein besonderer Schwerpunkt soll auf der Evaluierung im Hinblick auf totale Effizienz liegen, d.h. auf der Frage, inwiefern sich für moderat komplexe Probleme die zu erwartenden numerischen Gewinne auch in eine Reduzierung der Gesamtrechnenzeit niederschlägt. Ein Vergleich mit kommerzieller Software (Intel MKL) ist explizit Teil der Arbeit.

Coderahmen: Eigenbau, Lehrstuhl-Software (C++ oder Fortran95)

Literatur: Lehrbuch von Saad [41] und darin angegebene Referenzen, eigene Recherche

3.2 Parallele FSAI-Vorkonditionierung auf CPUs und/oder GPUs

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*, ggf. gute Kenntnisse in der Programmierung von GPUs

Schwerpunkt: Parallelisierung, Vorkonditionierung

Zusammenfassung: Nach einer kurzen Vorstellung des FSAI-Vorkonditionierers sollen in dieser Arbeit effiziente Strategien zu seiner parallelen OpenMP- und/oder CUDA/OpenCL-Assemblierung auf Systemen mit geteiltem Speicher und/oder auf GPUs entwickelt, beschrieben, analysiert und implementiert werden. Die Diskretisierung kann wahlweise mit Finiten Differenzen oder Finiten Elementen erfolgen. Die Implementierung soll für ein geeignetes Krylov-Unterraum-Verfahren (CG, BiCGStab, GMRES) anhand exemplarischer Modellprobleme aus den Kategorien Poisson, anisotrope Diffusion und Konvektion-Diffusion-Reaktion numerisch validiert werden. Ein besonderer Schwerpunkt soll – eine Bearbeitung im Team vorausgesetzt – auf dem Vergleich der verschiedenen Optimierungstechniken für Mehrkern-CPU's und GPU's liegen.

Coderahmen: Eigenbau, Lehrstuhl-Software (C++ oder Fortran95)

Literatur: Originalarbeit von Kolotilina und Yeregin [28] und Folgepublikationen, eigene Recherche

3.3 Optimierte Finite-Elemente-Assemblierung durch automatische Codegenerierung

Voraussetzung: Vorlesungen *Finite Elemente* oder *Praxisorientierte Einführung in die Methode der Finiten Elemente* oder *Numerische Verfahren in der Strömungsmechanik* o. ä.

Schwerpunkt: Effiziente Programmierung, Finite Elemente

Zusammenfassung: In den letzten Jahren wurden Techniken entwickelt, die gewissermaßen als „mathematische Compiler“ arbeiten und direkt aus der schwachen Formulierung einer partiellen Differentialgleichung optimierten Code erzeugen, der die Assemblierung der Koeffizientenmatrizen realisiert. In dieser Arbeit sollen solche Techniken beispielsweise anhand der Softwarepakete DOLPHIN und FENICS untersucht werden, und an (Lehrstuhl-) eigene Software zur Lösung der resultierenden linearen Gleichungssysteme angebunden werden. Die Arbeit beinhaltet einen Vergleich der angebundenen Techniken mit den in der Lehrstuhl-Software zur Verfügung stehenden konventionellen Implementierungen, für verschiedene Modellprobleme.

Coderahmen: Eigenbau, Lehrstuhl-Software (C++ oder Fortran95)

Literatur: Artikel von Kirby et al. [27] und Logg und Wells [31], eigene Recherche

3.4 Modellierung des nichtlinearen Dehnungs- und Streckungsverhaltens von Fasern am Beispiel selbst gestrickter Wintermode

Voraussetzung: irgendeine PDE-Vorlesung (hilfreich, nicht notwendig), Risikobereitschaft, ggf. eigene Erfahrungen im Stricken

Co-Betreuer: Prof. Dr. Matthias Röger

Schwerpunkt: Modellierung, mathematische Methoden in den angewandten Textilwissenschaften

Zusammenfassung: Mathematik findet sich in allen Lebensbereichen, verblüffenderweise auch in den angewandten Textilwissenschaften. Für den konkreten Anwendungsfall von „Oma’s selbst gestrickten warmen und bequemen Socken“ soll eine mathematische Modellierung beschrieben und analysiert werden, die es erlaubt, das nichtlineare Dehnungs- und Streckungsverhalten modischer und funktionaler Fußbekleidung zu erfassen und effizient zu simulieren.

Coderahmen: Eigenbau oder Matlab

Literatur: Artikel von Kaldor et al. [25]

3.5 Verbesserte SPAI-Vorkonditionierung

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*

Schwerpunkt: Vorkonditionierung, parallele Numerik

Zusammenfassung: Die klassische SPAI-Vorkonditionierungstechnik basiert auf einer Minimierung des Abstands der approximierten von der exakten Inversen in der Frobenius-Norm. Ist der Vorkonditionierer einmal konstruiert, reduziert sich seine Anwendung auf eine Matrix-Vektor Multiplikation, was ihn aus praktischer und paralleler Sicht durchaus attraktiv macht. Erlaubt man nur die Besetzungsstruktur der Koeffizientenmatrix, so ergibt sich allerdings i.A. kein besonders leistungsfähiges spektralenreduzierendes Verfahren. Eine Verbesserungsmöglichkeit im klassischen SPAI-Algorithmus basiert darauf, Fill-in mittels einer speziellen Residuen-minimierenden Technik zu erlauben. Es liegt nahe, bei der SPAI-Vorkonditionierung auch andere Techniken zur Verbesserung der Vorkonditionierungseigenschaften durch fill-in zu betrachten. Genau dies leistet der MSPAI (*modified SPAI*) Vorkonditionierer, indem dynamische fill-in-Techniken aus ILU-artigen Verfahren modifiziert und integriert werden. Verschiedene solche Möglichkeiten fill-in zu erlauben sollen dieser Arbeit theoretisch betrachtet werden und für ausgewählte Probleme basierend auf dem von den Autoren bereitgestellten Code experimentell analysiert werden.

Coderahmen: Eigenbau, Code der MSPAI-Autoren (<http://www5.in.tum.de/wiki/index.php/MSPAI>)

Literatur: SPAI-Originalarbeit [20], MSPAI-Arbeiten [21, 22, 26], eigene Recherche

3.6 MSPAI-Probing

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*

Schwerpunkt: Vorkonditionierung, parallele Numerik, Mehrgitterverfahren oder Bildbearbeitung

Zusammenfassung: Neben den in Themenvorschlag 3.5 beschriebenen Möglichkeiten erlaubt der MSPAI-Vorkonditionierer auch sehr problemspezifische Anpassungen mit einem sogenannten *probing*-Ansatz. Die Idee besteht im Wesentlichen darin, durch die Hinzunahme von Nebenbedingungen im Minimierungsproblem bspw. gezielt hochfrequente Fehlerkomponenten auszuwählen auf denen der Vorkonditionierer wirken soll, oder analog gezielt niederfrequente Fehlerkomponenten (zunächst) nicht zu berühren. Ein Beispiel dient der Verdeutlichung: In manchen Anwendungen der Bildbearbeitung will man „Rauschen“ in der Eingabe möglichst schnell eliminieren, so dass danach steile Gradienten im Pixelgitter in einer Segmentierung (Mustererkennung) zuverlässiger erkannt werden. Mathematisch läuft dies auf eine Optimierung bzgl. einer geschickt und sehr problemabhängig gewählten Halbnorm hinaus, es existiert ein sehr rigoroser theoretischer Unterbau. Beispielsweise lässt sich mit ähnlichen Techniken zeigen, dass das CG-Verfahren sehr schnell extremale Eigenwerte im (Fehler-)Spektrum eliminiert und danach typischerweise langsam konvergiert. In der Arbeit sollen verschiedene Varianten des MSPAI-Probing beschrieben und evaluiert werden, gerne auch mit anderen Anwendungsfällen als der Bildbearbeitung.

Coderahmen: Eigenbau, Code der MSPAI-Autoren (<http://www5.in.tum.de/wiki/index.php/MSPAI>)

Literatur: SPAI-Originalarbeit [20], MSPAI-Arbeiten [21, 22, 26], eigene Recherche

3.7 Der SPIKE-Algorithmus

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*

Schwerpunkt: lineare Löser, parallele Numerik

Zusammenfassung: Der SPIKE-Algorithmus ist ein Verfahren zur Lösung dünn besetzter linearer Gleichungssysteme, bei denen die Koeffizientenmatrix Bandstruktur aufweist. Das Verfahren zeichnet sich durch gute numerische Stabilität aus, ist für nicht diagonaldominante Systeme geeignet, und kann einfach parallelisiert werden. In der Arbeit soll der Algorithmus hergeleitet, beschrieben und analysiert werden. Numerische Experimente beinhalten bspw. den Vergleich mit spezialisierten Lösungsverfahren wie z.B. für Tridiagonalsysteme, sowie die Evaluierung der Skalierbarkeit und Stabilität für ausgewählte Testprobleme wie die Konvektions-Diffusions-Reaktionsgleichung.

Coderahmen: Eigenbau, der Algorithmus ist beispielsweise in der Intel Math Kernel Library verfügbar

Literatur: [7, 40], eigene Recherche

3.8 Parareal: Lösung von Anfangswertaufgaben parallel in der Zeit

Voraussetzung: Vorlesung Numerik II

Schwerpunkt: Zeitdiskretisierung, Parallelisierung

Zusammenfassung: Anfangswertaufgaben sind klassisch sequentiell, bzw. präziser, es wird innerhalb jedes Zeitschritts nur im Ort parallelisiert. Mit der *parareal*-Methode besteht die Möglichkeit, eine Anfangswertaufgabe grob gesprochen in eine Sequenz von Randwertaufgaben umzuformulieren, so dass jede RWA zunächst unabhängig voneinander und damit parallel in der Zeit gelöst werden kann, bevor am Ende die Start- und Endpunkte der Intervalle angeglichen werden. In dieser Arbeit soll die Methode beschrieben und analysiert werden, insbesondere für andere Anwendungsfälle als den wirtschaftsmathematischen Hintergrund aus der Originalpublikation, und andere Zeitschrittverfahren.

Coderahmen: Matlab, Eigenbau, Lehrstuhl-Software (C++ oder Fortran95)

Literatur: [2], eigene Recherche

3.9 Deflation: Beschleunigung von Krylov-Unterraum-Verfahren durch Eigenwert-Elimination

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*

Schwerpunkt: lineare Löser, Theorie von Krylov-Verfahren

Zusammenfassung: Man kann zeigen, dass (betragsmäßig) kleine Eigenwerte im Spektrum für die langsame Konvergenz von Krylov-Unterraum-Verfahren verantwortlich sind. Deflation-Techniken „verschieben“ gewissermaßen eine gewisse Zahl dieser kleinen Eigenwerte auf die Null und eliminieren sie so. Der entsprechende Verschiebungsoperator liefert ein kleineres Problem, und kann dann als Vorkonditionierer verwendet werden: Im Prinzip ist diese Verschiebung eine Projektion auf einen kleineren Unterraum. Ähnlich wie bei Mehrgitterverfahren wird also ein Grobgitterproblem gelöst, anders als bei Mehrgitterverfahren reicht dies jedoch zur Konvergenz aus – ein Glättungsoperator ist nicht erforderlich. In dieser Arbeit soll die Idee der Deflation hergeleitet werden, dem Mehrgitter-Prinzip gegenübergestellt werden, und die Leistungsfähigkeit der Methode anhand exemplarischer Beispielrechnungen für typische Probleme (anisotrope Diffusion, Konvektion-Diffusion-Reaktion) untersucht werden.

Coderahmen: Matlab, Eigenbau, Lehrstuhl-Software (C++ oder Fortran95)

Literatur: [34], eigene Recherche

3.10 Direkte Löser für dünn besetzte lineare Gleichungssysteme

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*

Schwerpunkt: lineare Löser, Graphentheorie

Zusammenfassung: Iterative Gleichungssystem-Löser haben einen prinzipiellen Nachteil: Berechnete Lösungen sind nach Konstruktion immer Approximationen an die exakte Lösung. Direkte Verfahren wie bspw. die Gauß-Elimination sind hingegen für dünn besetzte Gleichungssysteme ungeeignet, da die LU-Zerlegung einer dünn besetzten Matrix nicht notwendig dünn besetzt ist. Sogenannte „sparse direct“ Verfahren bieten einen Kompromiss, indem sie spezielle Techniken wie Multifront, teilweise Pivotierung oder Umsortierung einsetzen um den Fill-In während der LU-Zerlegung dünn besetzter Matrizen zu kontrollieren und zu minimieren. Diese Verfahren arbeiten zweistufig, zunächst wird in einer symbolischen Faktorisierung die Besetzungsstruktur der Faktorisierung bestimmt, um dann während der sogenannten numerischen Faktorisierung die tatsächlichen Werte zu bestimmen. Die tatsächliche Lösung erfolgt dann wie bei allen Faktorisierungen in Dreiecksmatrizen durch Vorwärts- und Rückwärtseinsetzen. In dieser Arbeit soll (je nachdem ob in einem Seminar oder als Abschlußarbeit) mindestens ein Zugang ausführlich algorithmisch beschrieben und bzgl. asymptotischer Laufzeit und Speicherplatzbedarf analysiert werden. Laufzeitmessungen und -Vergleiche für typische Testprobleme am Lehrstuhl (oder aus diversen Matrix-Sammlungen) komplettieren die Arbeit.

Coderahmen: Matlab, Eigenbau oder Lehrstuhlsoftware als Eingabe für externe Bibliotheken

Literatur: [12, 13, 30]

3.11 Inner-outer-Methoden

Voraussetzung: Vorlesungen *High Performance Computing und Parallele Numerik* oder *Schnelle Löser*

Schwerpunkt: lineare Löser, flexible und robuste Verfahren, Krylov-Theorie

Zusammenfassung: Auf den ersten Blick wirkt die Idee, ein Krylov-Unterraum-Verfahren mit sich selbst vorzukonditionieren, wie dummes Zeug. Geht man jedoch von fehlerbehafteter oder allgemein inexakter Rechnung aus, so erlauben es sogenannte „inner-outer-Methoden“, Fehler in der Rechnung zu kontrollieren und trotzdem ein konvergentes Verfahren zu erhalten. Neben der Fehlertoleranz finden solche Verfahren Anwendung bei (fast) singulären Problemen. Man kann beispielsweise Schranken zeigen, die die Konvergenz des äußeren, nicht fehlerbehafteten Verfahrens sicherstellen. In dieser Arbeit soll das existierende Theoriegebäude solcher Zugänge ausführlich dargestellt werden, und ein Überblick über mögliche Anwendungsfälle gegeben werden. Kleine Beispielrechnungen sollen zur Verdeutlichung unterschiedlicher Aspekte herangezogen werden.

Coderahmen: Matlab, Eigenbau oder Lehrstuhlsoftware

Literatur: [6, 34, 42]

3.12 Eigenwert-Schätzer

Voraussetzung: Vorlesungen *Schnelle Löser* oder *Numerik I*

Schwerpunkt: numerische lineare Algebra, Theorie

Zusammenfassung: In vielen iterativen Lösungsverfahren sind Schätzungen des Spektrums oder gleich der Eigenwertverteilung von dünn besetzten Koeffizientenmatrizen sehr hilfreich. Wenn solche Informationen zur Verfügung stehen, so kann oft das Konvergenzverhalten entscheidend und insbesondere problemspezifisch verbessert werden. Ein aus den Einführungsveranstaltungen in die Numerik bekanntes Beispiel ist die Wahl des optimalen Gewichtungsparameters ω im (S)SOR-Verfahren, weitere Beispiele umfassen so verschiedene Methoden wie die polynomielle oder die SPAI-Vorkonditionierung, oder den Entwurf numerisch schlauer Glätter für das Mehrgitterverfahren. Ein kurzer Blick in die Literatur zeigt, dass Schätzungen des betragsmäßig größten Eigenwerts weitestgehend trivial möglich sind, während beliebige Eigenwerte, insbesondere der oft benötigte betragsmäßig kleinste, nur mit großem numerischen Aufwand zu approximieren sind. Ziel dieser Arbeit ist es, verschiedene Verfahren zur Schätzung des Spektrums bzw. der Spektralnorm bzw. der Eigenwertverteilung herzuleiten und zu diskutieren bzgl. Genauigkeit, Stabilität und Aufwand, und ihre Leistungsfähigkeit und Anwendbarkeit zu vergleichen.

Coderahmen: Experimente mit Matlab

Literatur: Kapitel 5.4.5 meines Vorlesungsskripts [18], eigene Recherche ausgehend vom Wikipedia-Eintrag zum Thema^a

^a<http://de.wikipedia.org/wiki/Eigenwerte> → Numerische Berechnung

3.13 Vor- und Nachteile von Stabilisierungstechniken für Finite Elemente

Voraussetzung: Vorlesungen *Praktische Einführung in FEM* und/oder *Finite Elemente I*

Schwerpunkt: Diskretisierungstechniken

Zusammenfassung: Beim Entwurf guter Diskretisierungstechniken in Ort und Zeit tritt häufig das Problem verschiedener, sich widersprechender Ziele auf. Methoden, die die Stabilität von Lösungsverfahren sicherstellen, erzeugen gerne künstliche Diffusion, und umgekehrt. Entweder ist man also in der Situation, dass kleine Störungen in den Eingaben keine großen Störungen im Ergebnis verursachen, oder dass scharfe Grenzschichten nicht mehr adäquat aufgelöst werden können. Das Diskretisierungsverfahren, also „die Numerik“ erzeugen im zweiten Fall unphysikalische diffusive Eigenschaften, die sich in Verschmierungen oder in dämpfendem Verhalten manifestieren. Das Thema an sich füllt Regalmeter, deshalb soll sich diese Arbeit auf die Konvektions-Diffusions-Gleichung als Modellproblem beschränken. Die konkrete Ausgestaltung ist Verhandlungssache, insbesondere ob eher in der Breite verglichen (Diskussion verschiedener Verfahren wie SUPG, SOLD, FCT) oder ob in die Tiefe gegangen werden soll mit eigenen Implementierungen. Einflüsse der verschiedenen Techniken auf weitere Qualitätskriterien von Diskretisierungstechniken (Approximationsordnung, Erhaltung physikalischer Eigenschaften wie Nichtnegativität einer Lösung etc.) sollen beleuchtet werden. Spannend wäre auch eine Verbindung mit Löseraspekten, d.h., eine experimentelle Untersuchung, welche Technik welchen Aufwand im (nicht)linearen Löser erfordert oder welche Technik welchen Einfluss auf die Kondition des resultierenden linearen Gleichungssystems hat.

Coderahmen: FEAST-Software des LS3 (C++), JAVA-Software des LSX, oder Matlab

Literatur: Übersichtsartikel von Augustin et al. [1] und darin gegebene Referenzen

3.14 Algebraische Mehrgitterverfahren auf GPUs

Voraussetzung: Ahnung von geometrischen Mehrgitterverfahren, bspw. erworben in der Vorlesung *Schnelle Löser*

Schwerpunkt: Löser, GPUs ohne GPU-Programmierung

Zusammenfassung: In dieser Arbeit sollen *algebraische* Mehrgitterverfahren (AMG) betrachtet werden, bei denen die Eingabe keine Gitterhierarchie ist, sondern lediglich das diskrete Problem auf dem feinsten Gitter. AMG-Verfahren konstruieren dann die Hierarchie rein algebraisch aus der Koeffizientenmatrix, und zeichnen sich somit durch einen Black-Box-Charakter aus. Kürzlich hat NVIDIA die amgX-Bibliothek veröffentlicht, die verschiedene AMG-Implementierungen für GPUs beinhaltet. Nach einer kurzen Einführung in allgemeine AMG-Prinzipien kann die Arbeit verschiedene Schwerpunkte setzen. Eher theoretisch interessierte Studierende können verschiedene Konstruktionsprinzipien für Galerkin-Produkte gegenüberstellen, und eher praktisch interessierte Studierende vergleichen geometrische und algebraische Mehrgitterverfahren auf CPUs und GPUs.

Coderahmen: eigene Wrapper um externe Bibliotheken

Literatur: Übersicht und Einführung in AMG: Stüben [44], AMG-Implementierungen: <http://www.netlib.org/utk/people/JackDongarra/la-sw.html>, amgX: <https://developer.nvidia.com/amgx>, non-smoothed aggregation: Napov und Notay [35], Notay [36]

3.15 Umsortierung von unstrukturierten Gittern für die FEM

Voraussetzung: Vorlesungen *Praktische Einführung in FEM* und/oder *Finite Elemente I*, Vorlesung *Schnelle Löser*

Schwerpunkt: Finite Elemente, effiziente Implementierung, Graphentheorie

Zusammenfassung: Mit der Finite-Elemente-Methode werden PDEs unter Hinzunahme eines (unstrukturierten) Gitters in ein lineares Gleichungssystem übersetzt. Das Besetzungsmuster der Matrix ergibt sich hierbei zum Einen aus der Wahl des FE-Ansatzraumes und zum Anderen aus der Nummerierung der Gitter-Entitäten, d.h. der Knoten, Kanten und Flächen. Zur Beschleunigung von iterativen Lösern wird das Gleichungssystem hierbei häufig mit dem Cuthill-McKee-Algorithmus umsortiert. Aktuelle Tests haben aber gezeigt, dass auch die Performance des Matrix-Aufbaus signifikant von der Nummerierung des Gitters abhängt. Die Kernidee dieser Arbeit besteht darin, den von Cuthill und McKee vorgeschlagenen Algorithmus zur Umsortierung von Matrizen in einer modifizierten Form auf das dem Problem zugrunde liegenden Gitter anzuwenden, und erst anschliessend das Gleichungssystem mit der FEM aufzubauen. Auf diesem Wege kann so einerseits die Assemblierung des LGS deutlich beschleunigt werden, wobei sich dabei automatisch eine für den Löser günstige Sortierung der Matrix ergibt. Der Implementierungsaufwand beschränkt sich hierbei auf den modifizierten Cuthill-McKee-Algorithmus; die für die Beispielrechnungen notwendige FEM-Funktionalität wird von der Lehrstuhlsoftware bereitgestellt.

Coderahmen: Lehrstuhl-Software (C++)

Literatur: Umsortierungs-Algorithmus: Cuthill und McKee [11], eigene Recherche

3.16 Mathematik der Visualisierung zeitabhängiger Daten

Voraussetzung: Numerik II

Schwerpunkt: bunte Bilder, gewöhnliche Differentialgleichungen

Zusammenfassung: Die Visualisierung zeitabhängiger Daten (Vektorfelder) ist ein hochgradig komplexes Thema, aus einer hochdimensionalen Menge vektorwertiger Daten (Geschwindigkeiten etc.) muss eine zwei- oder dreidimensionale Darstellung generiert werden um in der visuellen Analyse eine Chance zu haben, Interpretationen des Verhaltens finden zu können. Erschwerend kommt die zeitabhängige Dynamik typischer Datensätze hinzu. In dieser Arbeit soll die Mathematik hinter typischen Visualisierungstechniken wie Stream- und Streaklines herausgearbeitet werden, und es soll auf die Linienintegral-Faltung eingegangen werden. Volumen-Rendering-Techniken können vernachlässigt werden, es reicht der Fokus auf den zweidimensionalen Fall. Spannend sind insbesondere Betrachtungen, die die Stabilität des Zeitintegrators balancieren mit der Antwortzeit des Visualisierungssystems: Die Darstellung soll ja nicht länger brauchen als die eigentliche Rechnung.

Coderahmen: Matlab

Literatur: kommt noch

3.17 Effiziente Realisierung von Stencilcodes auf GPUs am Beispiel eines Strömungslösers

Voraussetzung: Studienprojekt Modellbildung und Simulation TMABA

Schwerpunkt: effiziente Implementierung, Hardware-orientierte Numerik

Zusammenfassung: Das Ziel dieser Arbeit besteht darin, den im Studienprojekt gemeinsam entwickelten Strömungslöser auf GPUs zu portieren und deutliche Laufzeitgewinne zu erzielen. Insbesondere soll der Fokus auf der Beschreibung (und Erweiterung) effizienter Realisierungen für Stencils und Gitterfunktionen liegen.

Coderahmen: Studienprojekt

Literatur: diverses Vorlesungsmaterial von mir für GPU-Programmierung allgemein, speziell zu Stencil-Codes auf GPUs: [9, 33]

3.18 Vergleich unterschiedlicher Zeitschrittverfahren für Operator-Splitting Strömungslöser

Voraussetzung: Studienprojekt Modellbildung und Simulation TMABA

Schwerpunkt: Algorithmen- und Theorie

Zusammenfassung: Der im Studienprojekt realisierte Strömungslöser arbeitet voll explizit in der Zeit, die numerische Leistungsfähigkeit ist also durch eine CFL-Bedingung beschränkt: Anschaulich gesprochen muss die Zeitschrittweite immer kleiner gewählt werden, je genauer zeitabhängige Phänomene im Ort aufgelöst werden sollen. In dieser Arbeit sollen verschiedene alternative Zeitschrittverfahren vorgestellt und experimentell analysiert werden, sowohl solche mit höherer Ordnung als auch implizite Verfahren, die die Lösercharakteristik ändern.

Coderahmen: Studienprojekt

Literatur: Vorlesungsskripte zur Numerik gewöhnlicher Differentialgleichungen, Vorlesungen von Turek

3.19 Mehrgitterverfahren für reine Neumann-Probleme mit Anwendungen in der Strömungssimulation

Voraussetzung: Studienprojekt Modellbildung und Simulation TMABA

Schwerpunkt: Theorie und Praxis

Zusammenfassung: Der bisher im Studienprojekt verwendete iterative Löser für das Poisson-Problem im Druck hat sich als ein großer Flaschenhals herausgestellt, und soll in dieser Arbeit durch ein besseres Verfahren ersetzt werden. Insbesondere bietet sich ein Mehrgitterverfahren an. Die Herausforderung besteht nun darin, dass das Laplace-Problem mit reinem Neumann-Rand versehen ist. Die Lösung ist also nur bis auf eine Konstante eindeutig bestimmt, und spezielle Filtertechniken sind notwendig, um Verfahren zu konstruieren, die zur (im starken Sinne) gesuchten Lösung konvergieren unter Beibehaltung der Konvergenzeigenschaften, die man bspw. von einem Mehrgitterverfahren für Laplace-Probleme erwartet.

Coderahmen: Studienprojekt

Literatur: Bochev und Lehoucq [5] (ohne den Teil mit finiten Elementen) und Referenzen darin, eigene Recherche

3.20 Erweiterung des Strömungslösers um Energietransport

Voraussetzung: Studienprojekt Modellbildung und Simulation TMABA

Schwerpunkt: Modellierung und Praxis

Zusammenfassung: In vielen praxisrelevanten Szenarien ist der Einfluss der Temperatur auf die Strömung von großer Bedeutung, sowie der Wärmetransport durch die Strömung. Ein elementares Beispiel ist ein Heizkörper, ein etwas spannenderes die Meteorologie. In dieser Arbeit sollen die Navier-Stokes-Gleichungen um entsprechende Modelle erweitert werden. Zur numerischen Behandlung sind analog zur Simulation von „reinen“ Fluiden wieder Vereinfachungen nötig, hierbei soll sich auf die sogenannte Boussinesq-Approximation konzentriert werden. Ein mögliches Simulationsszenario ist die Entlüftung eines Tunnels, in dem stehende Fahrzeuge Abgase produzieren.

Coderahmen: Studienprojekt

Literatur: Griebel et al. [19] (Kapitel 9) und Referenzen darin, eigene Recherche

3.21 Erweiterung des Strömungslösers für Turbulenzmodelle

Voraussetzung: Studienprojekt Modellbildung und Simulation TMABA

Schwerpunkt: Modellierung und Praxis

Zusammenfassung: Am Beispiel der Zylinderumströmung kann eine starke Abhängigkeit der Strömung von der Reynoldszahl beobachtet werden. Mit zunehmender Reynoldszahl wechselt die Strömung von stationärem zu laminarem Verhalten, um schließlich periodisches bzw. quasi-periodisches Verhalten aufzuweisen. An dieser Stelle bricht das im Studienprojekt bisher betrachtete Modell zusammen, und um das in der Praxis zu beobachtende chaotisch-turbulente Verhalten bei noch höheren Reynoldszahlen simulieren zu können, müssen die Navier-Stokes-Gleichungen um ein sogenanntes Turbulenzmodell ergänzt werden. Ziel dieser Arbeit ist die Beschreibung des $k - \epsilon$ Turbulenzmodells und seine Integration in den existierenden Strömungslöser.

Coderahmen: Studienprojekt

Literatur: Griebel et al. [19] (Kapitel 10) und Referenzen darin, eigene Recherche

4 Masterarbeiten

In einer Masterarbeit sollen Sie deutlich machen, dass Sie eine *vertiefte* Fragestellung mit dem Kenntnisstand des Masterstudiums selbstständig bearbeiten können. In meinem „Weltbild“ beinhalten Masterarbeiten daher einen deutlich stärkeren Forschungsbezug als Bachelorarbeiten, und das Themenspektrum meiner im Folgenden skizzierten Vorschläge folgt diesem Grundgedanken. Deshalb beschreibe ich auch nur grob das *Thema*, aber nicht das *Ziel* einer Arbeit.

4.1 Mehrgitterverfahren für konvektionsdominante Probleme

Geometrische Mehrgitterverfahren liefern typischerweise eher schlechte Konvergenzraten für transportdominante Probleme wie z.B. die Modellgleichung $-\varepsilon \Delta u + \mathbf{w} \cdot \nabla u = f$ mit $\varepsilon \ll 1$. Sogar die h -unabhängige Konvergenz kann verloren gehen, wenn man sich ungeschickt anstellt. Der Grund ist im Prinzip, dass geometrische Mehrgitterverfahren für primär elliptische PDEs gedacht sind, während die Konvektions-Diffusions-Modellgleichung für $\varepsilon \rightarrow 0$ hyperbolisch wird.

Typische Gegenmaßnahmen zur „Rettung“ der Glättungseigenschaft sind Relaxationsverfahren die eine (implizite) Umnummerierung vornehmen. Die Idee ist hier, dass ein handelsübliches Gauß-Seidel-Verfahren ein beinahe exakter Löser ist, wenn die zugrundeliegende Nummerierung gerade zur Transportrichtung $\mathbf{w}$ „passt“, wenn also relevante Beiträge immer in „upwind-Richtung“, d.h. im rechten unteren Dreiecksteil der Koeffizientenmatrix liegen. Wenn allerdings $\mathbf{w} = \mathbf{w}(x, y)$, so ist dieser Ansatz nicht mehr praktikabel, und man muss in 2D zu sogenannten „linelet“-Glättern [43] (in 3D „planelet“) übergehen, die lokal die Transportrichtung verfolgen. Ein in meiner aktuellen Forschung verfolgter Ansatz ist die Generalisierung von Färbungstechniken hin zu etwas, was wir als „Farbschattierungen“ bezeichnen, unsere Idee besteht im Prinzip darin, die Ausnutzung lokaler Transportrichtungen so durchzuführen, dass genug feingranulare Parallelität „übrig“ bleibt um das Verfahren auf GPUs anzuwenden. Ein orthogonaler Ansatz ist das MSPAI-Probing [21, 22, 26], s. auch das mögliche Masterseminar 3.6: Hier wird ein passender von $\mathbf{w}$ abhängiger Raum definiert, so dass die SPAI-Konstruktionsvorschrift [18, 20] (direkte Approximation der Inversen unter Nebenbedingungen an die Besetzungsstruktur) den Transportterm berücksichtigt in der inkrementellen Konstruktion des Vorkonditionierers bzw. des Glätters.

Typische Gegenmaßnahmen zur „Rettung“ der Grobgitterkorrektur bestehen darin, die Gittertransfer-Operatoren an die Transportrichtung anzupassen, bspw. in Form eines sogenannten *semi-coarsenings*, bei dem für Transporterme mit dominierender Richtung eben gerade nur in die relevante Richtung vergrößert wird.

Das Thema dieser Arbeit ist also in erster Näherung, für geometrische Mehrgitterverfahren herauszufinden, welchen Einfluss welche Strategie hat, und wie welche Strategie möglicherweise verbessert werden kann. Dazu soll zunächst das Konvektions-Diffusions-Modellproblem betrachtet werden, eine spätere Erweiterung bspw. auf die Navier-Stokes-Gleichungen ist explizit nicht ausgeschlossen.

Die Implementierung soll im Rahmen der Lehrstuhl-eigenen FEAST2-Software (C++) erfolgen, hier sind für SHK mit FEAST2-Hintergrund zahlreiche Synergie-Effekte möglich, insbesondere bzgl. der FEM-Assemblierung der nicht-Standard Glättungs- und Transferoperatoren. Aus FEM-Sicht gibt es viele Schnittpunkte [24].

Eine mögliche Kür ist die vergleichende Betrachtung von *algebraischen Mehrgitterverfahren*, die sich gerne genau umgekehrt verhalten: Schon bei einfachen Testfällen geht im diffusionsdominanten Fall die h -unabhängige Konvergenz verloren [32, 35, 36].

4.2 Lösung von Anfangswertaufgaben parallel in der Zeit

Das Parareal-Konzept (s. das mögliche Masterseminar 3.8) erstellt die initiale Partitionierung, d.h. die Übertragung einer AWA in eine Folge gekoppelter RWA, vollkommen sequentiell. Damit ist das Verfahren prinzipiell nicht skalierbar: Man kann zeigen, dass der maximale Speedup durch die Parallelisierung

asymptotisch durch einen Faktor zwei beschränkt ist. In der Arbeit sollen schlaunere Zeit-parallele Ansätze beschrieben, evaluiert und erweitert werden, indem die Zeitparallelität von einem Zweigitter- in ein Mehrgitterverfahren übertragen wird, wie man sich leicht überlegt, stellt Parareal ja ein Zweigitterverfahren in der Zeit dar. Eine mögliche Lösung ist der PFASST-Algorithmus, der die Mehrgitteridee des FAS-Zugangs (full approximation scheme) auf die Parallelisierung in der Zeit überträgt. Ein hervorragender Startpunkt für eine eigene Recherche zum Thema ist die Webseite <http://www.unc.edu/~mwemmett/pfasst> bzw. das die Python-Software unter <http://pypfasst.readthedocs.org/en/latest/>. Der Coderahmen für dieses Thema ist noch etwas unklar, Prof. Turek wird bei einem Fokus auf die Navier-Stokes-Gleichungen die Co-Betreuung übernehmen.

4.3 Inverse Probleme und Wavelet-Kompressionen am Beispiel der Leitfähigkeits-Tomographie

Bei der Leitfähigkeits-Tomographie wird versucht anhand von diskreten Messdaten zu rekonstruieren, wie bspw. eine mehrphasige Strömung zusammengesetzt ist. Das klassische Beispiel ist ein Pumprohr in einer Ölquelle, bei dem die Betreiberfirma möglichst in Echtzeit und ohne die Strömung zu unterbrechen wissen muss, wie hoch der Anteil an Gasbläschen im geförderten Öl ist (dies beeinflusst typischerweise den weiteren Raffinationsprozess). Dazu wird um das Rohr ein Messinstrument mit reihum angeordneten Sensoren angebracht, welches in einer gewissen Frequenz Messdaten liefert indem Leitfähigkeitsprofile des beobachteten mehrphasigen Fluids zwischen den Sensoren erstellt werden [37–39]. Die Simulation der Erstellung eines solchen Profils kann als Lösung eines Poisson-Problems modelliert werden, man spricht vom sogenannten Vorwärtsproblem. Viele solcher Messpunkte und Simulationen werden integriert, man erhält ein dünn besetztes überbestimmtes Gleichungssystem impraktikabler Dimension, dessen Lösung eine Bildgebung des beobachteten Fluids darstellt. In der Arbeit sollen nun mehrere Möglichkeiten betrachtet werden, dieses *inverse Problem* effizient zu lösen. Aus dem großen Zoo an Möglichkeiten soll der Fokus zweigeteilt werden: Wavelet-Kompressionen zur Modellreduktion erlauben die Kompression der Ausgangsdaten mit kontrollierbarem Informationsverlust (s. [8] für einen analogen Zugang in der Geophysik), und iterative Verfahren lassen sich auf den Fall überbestimmter Gleichungssysteme umformulieren (s. z.B. LSQR). Die Arbeit erlaubt zwei mögliche Ausgestaltungen, entweder mathematisch-methodisch oder im Hinblick auf die quasi-Echtzeit Anforderung der Bildgebung.

4.4 Asynchrone numerische Verfahren

Zukünftige Exascale-Systeme, d.h. Supercomputer mit einer nominelle Rechenleistung von 10^{18} (!) Gleitkomma-Operationen pro Sekunde, werden für das Jahr 2018 erwartet. Der Zunahme an Rechenleistung um den Faktor 100-1000 verglichen mit aktuellen Höchstleistungsrechnern steht jedoch die Erwartung massiver Probleme im Bereich der Synchronität gegenüber [14]. Dies ist kein reines „exascale“-Problem. Schon heute beobachten KMUs (kleine mittelständische Unternehmen, „Ingenieurbüros“ als Ihre potentiellen zukünftigen Arbeitgeber) variables Verhalten, wenn Simulationsrechnungen in „die Cloud“ ausgelagert werden. Man könnte nun die Numerik so implementieren, dass auf solchen Rechenressourcen immer brutal synchronisiert wird, in der Konsequenz wartet also der Teil der Rechnung der „in den USA“ durchgeführt wird auf den Teil der Rechnung, der „in Irland“ stattfindet. Dies kostet Geld, bzw. ist numerisch unbefriedigend.

Man könnte aber auch schlaunere sein, und das globale Verfahren *asynchron* definieren. Genau das ist das Thema dieser Arbeit: Was passiert, wenn Teile der Grobgitterkorrektur in Mehrgitterverfahren erst verspätet eintreffen [23], analog für Beiträge in Krylov-Unterraum-Verfahren? Für das Jacobi-Verfahren kann man (als Löser) tatsächlich präzise Konvergenz nachweisen [3, 4, 15], sofern Beiträge nicht „ewig“ verspätet sind. Für leistungsfähigere Lösungsverfahren ist die Situation unklar, auf theoretischer Ebene bietet das Konzept von *inner-outer-Methoden* Erfolgsaussichten [6, 10, 34, 42]. Berücksichtigungswert sind auch Entwicklungen in der „grid“-community, dem wissenschaftlichen Vorläufer der Cloud.

Der Fokus dieser Arbeit ist beliebig gestaltbar im Spannungsfeld zwischen Theorie, exa-Vorbereitung, cloud-computing und Anwendungen – Details überlegen wir uns bei Interesse an diesem Thema zusammen.

4.5 Feingranulare Assemblierung der Galerkin-Grobgitterkorrektur in algebraischen Mehrgitterverfahren

In algebraischen Mehrgitterverfahren wird typischerweise nicht das Grobgitter vorgegeben, sondern ein Transferoperator zwischen dem Fein- und Grobgitter. Der Grobgitteroperator ergibt sich hier als Galerkin-Produkt, in die Definition geht der Feingitteroperator ein, sowie die Transferoperatoren. Wenn wir also einen Transfer repräsentieren können als (rechteckige) Matrix, so ist das Galerkin-Produkt „auf dem Papier“ eine einfache Matrixmultiplikation. In der Praxis macht diese Vorgehensweise natürlich keinen Sinn, so dass in einer Masterarbeit untersucht werden soll, wie die Operatorkonstruktion in algebraischen Mehrgitterverfahren feingranular parallel (OpenMP, CUDA, TBB) umgesetzt werden kann.

5 Abgeschlossene Arbeiten (Auswahl)

In diesem Abschnitt werden einige bereits absolvierte Arbeiten zusammengefasst. Dies dient zum einen zur Inspiration, und zum anderen zur Verdeutlichung der großen thematischen Breite von Abschlussarbeiten bei mir: Wie eingangs bereits erwähnt, stehe ich vielen Themenvorschlägen sehr offen gegenüber. Die Reihenfolge dieser Liste ist invers chronologisch und nicht thematisch.

5.1 Effiziente Sparse-Matrix-Vektor-Multiplikation auf moderner Hardware

Typ: Master-Seminar Technomathematik 2013

Zusammenfassung: Matrix-Vektor-Multiplikationen mit dünnbesetzten Matrizen sind Kernkomponente vieler numerischer Verfahren, und sie dominieren oft die Laufzeit. Während für auf gewisse Weise strukturierte Matrizen (Bandstruktur, Blockstruktur) durchaus viele effiziente Ansätze existieren, ist der Fall beliebig unstrukturierter und damit allgemeiner Matrizen prinzipiell ungeeignet für moderne Hardware. Dennoch (oder gerade deshalb) wird seit langem an Datenstrukturen jenseits des klassischen CSR-Formats geforscht, um diese Basisoperation „Hardware-freundlich“ realisieren zu können, insbesondere im Bezug auf fein-granulare Parallelität (Multicore, GPUs, Xeon Phi). Kürzlich haben Kreutzer et al. [29] ein Format vorgestellt, dass für die von ihnen verwendeten Testprobleme konkurrenzfähige Ausführungsgeschwindigkeit und Hardwareeffizienz auf CPUs, GPUs und der MIC-Architektur verspricht. In dieser Arbeit soll zunächst ein Überblick über verschiedene Speicherformate gegeben werden (CSR, ELL, JD) bevor das SELL-C- σ Format ausführlich vorgestellt wird. Wesentlicher Teil der Arbeit ist die Evaluierung anhand von Matrixstrukturen die bei der Diskretisierung partieller Differentialgleichungen mittels Gitter-basierter Methoden auftreten.

5.2 IDR(s) und QMRCGStab: Krylov-Unterraum-Verfahren für nicht-SPD-Systeme

Typ: Master-Seminar Technomathematik 2013

Zusammenfassung: Das CG-Verfahren ist ein effizientes und elegant theoretisch untermauertes iteratives Verfahren für symmetrische, positiv definite dünn besetzte lineare Gleichungssysteme. Ist die SPD-Voraussetzung nicht gegeben, so kann das BiCGStab-Verfahren verwendet werden, welches jedoch nicht immer robust ist: Der Konvergenzverlauf weist typischerweise Oszillationen auf, und es kann zu vorzeitigen Verfahrensabbrüchen kommen. In der Literatur existieren viele Verbesserungsmöglichkeiten, zwei Vertreter sollen in dieser Arbeit beschrieben, theoretisch analysiert und für Modellprobleme untersucht werden, namentlich das IDR(s)- und das QMRCGStab-Verfahren. Beide basieren auf einer Quasi-Minimierung der lokalen Suchrichtung, was Oszillationen reduziert oder sogar verhindert.

5.3 Algebraische Mehrgitter-Verfahren

Typ: Bachelorarbeit Technomathematik 2013

Zusammenfassung: Die aus den Vorlesungen HPC-PARNUM und Numerik 2 bekannten Lösungsverfahren für dünn besetzte lineare Gleichungssysteme haben den Nachteil, nur h -abhängig zu konvergieren, d.h. je feiner eine Diskretisierung ist, desto mehr Iterationen werden benötigt. Mehrgitter-Verfahren konvergieren optimal, unabhängig von h und mit einem Rechenaufwand, der linear in der Zahl der Unbekannten ist. Die Grundidee ist es hierbei, eine Hierarchie immer größerer diskreter Repräsentationen des Problems zu betrachten; und auf jeder Gitterebene Fehlerfrequenzen zu reduzieren: Hochfrequente Fehlerkomponenten können durch einfache Relaxationsverfahren (Jacobi, Gauß-Seidel etc.) effizient eliminiert werden, niederfrequente hingegen nicht. Diese Fehlerkomponenten sind jedoch hochfrequent auf gröberen Gittern. Mehrgitterverfahren arbeiten also rekursiv, zunächst werden Fehlerkomponenten im jeweiligen Defekt eliminiert, und auf dem Rückweg werden so gewonnene Korrekturen in die jeweilige feinere Lösung integriert. Im Gegensatz zu den evtl. aus anderen VL bekannten *geometrischen* Mehrgitterverfahren sollen in dieser Arbeit sog. *algebraische* Mehrgitterverfahren betrachtet werden, bei denen die Eingabe keine Gitterhierarchie ist, sondern lediglich das diskrete Problem auf dem feinsten Gitter. AMG-Verfahren konstruieren dann die Hierarchie rein algebraisch aus der Koeffizientenmatrix, und zeichnen sich somit durch einen Black-Box-Charakter aus. Der Schwerpunkt der Arbeit soll auf der theoretischen Konstruktion dieser Gittertransfer-Operatoren liegen. In einem kurzen Praxisteil soll für ausgewählte Testprobleme (Poisson, anisotrope Diffusion, Konvektion-Diffusion) die Leistungsfähigkeit der AMG-Techniken mit der von Krylov-Unterraum-Verfahren verglichen werden, dazu soll eine geeignete Open-Source-Implementierung von AMG verwendet werden.

5.4 Verbesserte (Block-) Vorkonditionierungstechniken für Probleme der linearisierten Elastizität

Typ: Bachelorarbeit Technomathematik 2013

Zusammenfassung: Im Studienprojekt wurden verschiedene Vorkonditionierungstechniken betrachtet, für globale Probleme als auch für lokale Probleme in einem SDO-artigen äußeren (iterativen) Block-Zugang. Die Untersuchungen beschränkten sich jedoch auf Bandmatrizen und darauf zugeschnittene elementare Vorkonditionierer. In dieser Arbeit sollen alternative und verbesserte Vorkonditionierer betrachtet werden: Ein Teil der Arbeit soll sich mit der maximalen Ausnutzung der Bandstruktur im SDO-Fall beschäftigen, indem parallele Färbungs- und Linien-/Flächenvorkonditionierer analog zur Vorlesung entwickelt und analysiert werden (GS, ADI-3D). Der zweite/alternative Teil dieser Arbeit soll sich zunächst einer Verbesserung der CSR-Datenstruktur widmen, bei der in einem Gitterpunkt alle Komponenten des Verschiebungsfelds gespeichert werden und somit die Verwaltungsstrukturen nicht für jeden Nicht-Null-Eintrag der Koeffizientenmatrix gespeichert werden müssen. Für einen entsprechend modifizierten PDO-artigen Zugang sollen dann darauf zugeschnittene leistungsfähige Vorkonditionierer evaluiert werden. Die Arbeit schliesst mit einer gemeinsamen Gegenüberstellung der Ergebnisse.

5.5 Finite-Elemente-Assemblierung auf GPUs

Typ: Bachelorarbeit Technomathematik 2013

Zusammenfassung: In dieser Arbeit sollen Assemblierungstechniken für konforme Vierecks- und Hexaederelemente erster und zweiter Ordnung (bilinear, trilinear, bi-quadratisch, triquadratisch) auf GPUs untersucht werden. Die zugrunde liegenden Gitter sollen unstrukturiert sein, als exemplarischer Testfall dient das Poisson-Problem. Verschiedene Techniken sollen gegenübergestellt und verglichen werden. Demonstriert werden sollen am Ende Effizienzsteigerungen durch die GPU gegenüber einer konventionell mit OpenMP parallelisierten CPU-Referenzimplementierung aus einem am Lehrstuhl entwickelten Softwarepaket.

5.6 Image Stitching: Mathematische Modellierung der Berechnung von Panoramafotos

Typ: Bachelorarbeit Technomathematik 2012

Zusammenfassung: Panoramafotos werden oft aus vielen Einzelbildern zusammengesetzt. In der professionellen Bildbearbeitung existieren schon lange Softwarelösungen, die diese Funktionalität bereitstellen. Seit kurzem haben auch Smartphones genug Rechenkraft für diese Aufgabe, so gibt es bspw. eine iPhone-App. Mathematisch kann die Berechnung optimaler Übergänge zwischen den Teilbildern als eine elliptische partielle Differentialgleichung, also eine PDE vom Typ des Poisson-Problems beschrieben werden. Die inexakten, potentiell überlappenden Übergänge zwischen den Einzelbildern werden dabei durch nicht-Standard interne Randbedingungen realisiert. In dieser Arbeit soll genau diese Modellierung mathematisch fundiert beschrieben werden, und es sollen Lösungsverfahren vorgeschlagen und prototypisch implementiert werden.

5.7 Strömungssimulation in Echtzeit auf Smartphones und Tablet-PCs

Typ: Bachelorarbeit Technomathematik 2012

Zusammenfassung: Die Prozessoren in aktuellen Smartphones und Tablets werden immer mächtiger, was sie auch für numerische Simulationen spannend erscheinen lässt. In dieser Arbeit soll eine App erstellt werden, die interaktive 2D-Strömungssimulation in Echtzeit auf einem solchen Endgerät erlaubt. Insbesondere soll es möglich sein, das Verhalten des Fluids über den Touchscreen zu beeinflussen, und eine ansprechende Echtzeit-Visualisierung ist zu realisieren. Als Basis für die Lösung der Navier-Stokes Gleichungen soll das „Stable Fluids“-Verfahren von Stam dienen.

5.8 Die Spektrale Elemente Methode für die Simulation der Ausbreitung seismischer Wellen

Typ: Bachelorarbeit Technomathematik 2012

Zusammenfassung: In der Geophysik wird die Ausbreitung seismischer Wellen unter anderem zur Vorhersage von Erdbeben simuliert und mit den elastischen Wellengleichungen für heterogene Medien modelliert. Zur Diskretisierung kann die Spektrale Elemente Methode (SEM) verwendet werden, die die Flexibilität der klassischen FEM mit der Genauigkeit spektraler Methoden kombiniert. In dieser Arbeit soll die SEM ausführlich vorgestellt, analysiert und diskutiert werden.

5.9 Kommunikations-vermeidende Algorithmen im Höchstleistungsrechnen

Typ: Masterarbeit Mathematik 2012

Zusammenfassung: Zukünftige Exascale-Systeme, d.h. Supercomputer mit einer nominelle Rechenleistung von 10^{18} (!) Gleitkomma-Operationen pro Sekunde, werden für das Jahr 2018 erwartet. Der Zunahme an Rechenleistung um den Faktor 100-1000 verglichen mit aktuellen Höchstleistungsrechnern steht jedoch eine deutlich geringere Zunahme der Kommunikationsgeschwindigkeit gegenüber. Es besteht also ein starker Bedarf an kommunikationsvermeidenden bzw. kommunikationsminimierenden Algorithmen. Hier kommt die Numerik ins Spiel: Für viele Krylov-Unterraum-Verfahren ist es mit einigem Aufwand möglich, Abhängigkeiten zwischen Matrix-Vektor-Multiplikationen und Vektor-Vektor-Operationen (Skalarprodukte) aufzubrechen, so dass die Kommunikationskosten bei s Iterationen nur $1 + O(1)$ betragen, statt $O(s)$ in klassischen Formulierungen. In dieser Arbeit sollen solche kommunikationsminimierende und kommunikationsvermeidende Verfahren beschrieben, analysiert, exemplarisch implementiert und für Poisson-artige Testprobleme evaluiert werden.

5.10 Optimale Färbungen für parallele Algorithmen

Typ: Bachelorarbeit Technomathematik 2012

Zusammenfassung: Färbungsstrategien stellen ein im parallelen Rechnen oft verwendetes Universalhilfsmittel dar, und finden sich in so verschiedenen Bereichen wie der Lastverteilung, der Entkopplung sequentieller Abhängigkeiten, der Umordnung von Berechnungen zur Erhöhung des Parallelitätsgrades etc. Man unterscheidet zwischen exakter (bspw. FEM-Assemblierung auf unstrukturierten Gittern) und inexakter Parallelisierung (bspw. Gauß-Seidel-Vorkonditionierung. Die Bestimmung einer optimalen Färbung ist NP-hart, d.h. für Nichtinformatiker pragmatisch, dass kein sequentieller Algorithmus bekannt ist, der weniger als exponentielle Laufzeit in der Größe der Eingabe benötigt. In dieser Arbeit sollen verschiedene Färbungsstrategien beschrieben und exemplarisch evaluiert werden. Je nach persönlichem Interesse kann dabei der Fokus auf graphentheoretischen Grundlagen, der Balancierung zwischen möglichst optimaler Färbung und der Entkopplung von Abhängigkeiten (bspw. bei der effizienten Anwendung von Mehrfarben-Vorkonditionierern auf CPUs und GPUs auf unstrukturierten Gittern) oder bei Färbungsstrategien für die exakte FEM-Assemblierung auf CPUs liegen.

Literatur

- [1] Matthias Augustin, Alfonso Caiazzo, Aandré Fiebach, Jürgen Fuhrmann, Volker John, Alexander Linke, und Rudolf Umla. An assessment of discretizations for convection-dominated convection-diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 200(47–48):3395–3409, November 2011. DOI: 10.1016/j.cma.2011.08.012.
- [2] Guillaume Bal und Yvon Maday. A parareal time discretization for non-linear PDEs with application to the pricing of an american put. In Luca F. Pavarino und Andrea Toselli (Hrsg.), *Recent Developments in Domain Decomposition Methods*, volume 23 of *Lecture Notes in Computational Science and Engineering*, Seite 189–202. Springer, 2002. DOI: 10.1007/978-3-642-56118-4_12.
- [3] R. H. Barlow und D. J. Evans. Parallel algorithms for the iterative solution to linear systems. *The Computer Journal*, 25(1):56–60, November 1982. DOI: 10.1093/comjnl/25.1.56.
- [4] Kostas Blathras, Daniel B. Szyld, und Yuan Shi. Parallel processing of linear systems using asynchronous iterative algorithms. Technischer Bericht, Temple University, Philadelphia, April 1997. http://spartan.cis.temple.edu/shi/public_html/super96/kostas/super96.html.
- [5] Pavel Bochev und Richard B. Lehoucq. On the finite element solution of the pure Neumann problem. *SIAM Review*, 47(1):50–66, Januar 2005. DOI: 10.1137/S0036144503426074.
- [6] Amina Bouras und Valérie Frayssé. Inexact matrix-vector products in Krylov methods for solving linear systems: A relaxation strategy. *SIAM Journal Matrix Analysis and Applications*, 26(3):660–678, März 2005. DOI: 10.1137/S0895479801384743.
- [7] Li-Wen Chang, John A. Stratton, Hee-Seok Kim, und Wen-mei W. Hwu. A scalable, numerically stable, high-performance tridiagonal solver using GPUs. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC’12)*, Seite 27:1–27:11, November 2012.
- [8] Sébastien Chevrot, Roland Martin, und Dimitri Komatitsch. Optimized discrete wavelet transforms in the cubed sphere with the lifting scheme—implications for global finite-frequency tomography. *Geophysical Journal International*, 191(3):1391–1402, Dezember 2012. DOI: 10.1111/j.1365-246X.2012.05686.x.
- [9] Matthias Christen, Olaf Schenk, und Yifeng Cui. Patus for convenient high-performance stencils: evaluation in earthquake simulations. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC’12)*, Seite 11:1–11:10, November 2012.
- [10] Tijmen P. Collignon und Martin B. van Gijzen. Fast iterative solution of large sparse linear systems on geographically separated clusters. *International Journal of High Performance Computing Applications*, 25(4):440–450, November 2011. DOI: 10.1177/1094342010388541.
- [11] Elizabeth Cuthill und J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, ACM ’69, Seite 157–172, 1969. DOI: 10.1145/800195.805928.
- [12] Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, Juni 2004. DOI: 10.1145/992200.992205.

- [13] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, und Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3):720–755, Juli 1999. DOI: 10.1137/S0895479895291765.
- [14] Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfo Hoisie, Koh Hotta, Zhong Jin, Yutaka Ishikawa, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr, Matthias S. Mueller, Wolfgang E. Nagel, Hiroshi Nakashima, Michael E. Papka, Dan Reed, Mitsuhsa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski, und Kathy Yelick. The international exascale software project roadmap. *International Journal of High Performance Computing Applications*, 25(1):3–60, Februar 2011. DOI: 10.1177/1094342010391989.
- [15] Andreas Frommer und Daniel B. Szyld. On asynchronous iterations. *Journal of Computational and Applied Mathematics*, 123(1–2):201–216, November 2000. DOI: 10.1016/S0377-0427(00)00409-X.
- [16] Dominik Götdeke. Gebietszerlegungsverfahren, Juli 2012. Vorlesungsskript, Fakultät für Mathematik, TU Dortmund.
- [17] Dominik Götdeke. High performance computing und parallele Numerik, Februar 2012. Vorlesungsskript, Fakultät für Mathematik, TU Dortmund.
- [18] Dominik Götdeke. Schnelle Löser, Dezember 2013. Vorlesungsskript, Fakultät für Mathematik, TU Dortmund.
- [19] Michael Griebel, Thomas Dornseifer, und Tilman Neunhoffer. *Numerische Simulation in der Strömungsmechanik*. Vieweg, 1995.
- [20] Marcus J. Grote und Thomas Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing*, 18(3):838–853, Mai 1997. DOI: 10.1137/S1064827594276552.
- [21] Thomas Huckle und Alexander Kallischko. Frobenius norm minimization and probing for preconditioning. *International Journal of Computer Mathematics*, 84(8):1225–1248, August 2007. DOI: 10.1080/00207160701396387.
- [22] Thomas Huckle, Alexander Kallischko, Andreas Roy, Matous Sedlacek, und Tobias Weinzierl. An efficient parallel implementation of the MSPAI preconditioner. *Parallel Computing*, 36(5–6):273–284, 2010. DOI: 10.1016/j.parco.2009.12.007.
- [23] Peter K. Jimack und Mark A. Walkley. Asynchronous parallel solvers for linear systems arising in computational engineering. *Computational Technology Reviews*, 3:1–20, 2011. DOI: 10.4203/ctr.3.1.
- [24] Volker John und Petr Knobloch. On the choice of parameters in stabilization methods for convection–diffusion equations. In Karl Kunisch, Günther Of, und Olaf Steinbach (Hrsg.), *Numerical Mathematics and Advanced Applications – Proceedings of ENUMATH 2007*, Seite 297–304. Springer, September 2008. DOI: 10.1007/978-3-540-69777-0_35.

- [25] Jonathan M. Kaldor, Doug L. James, und Steve Marschner. Simulating knitted cloth at the yarn level. *ACM Transactions on Graphics*, 27(3):65:1–65:9, August 2008. DOI: 10.1145/1360612.1360664.
- [26] Alexander Kallischko. *Modified Sparse Approximate Inverses (MSPAI) for Parallel Preconditioning*. Doktorarbeit, Fakultät für Mathematik, Technische Universität München, März 2008. <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20071114-632977-1-5>.
- [27] Robert C. Kirby, Matthew Knepley, Anders Logg, und L. Ridgway Scott. Optimizing the evaluation of finite element matrices. *SIAM Journal on Scientific Computing*, 27(3):741–758, Oktober 2005. DOI: 10.1137/040607824.
- [28] Liliya Yurievna Kolotilina und A. Yu. Yereimin. Factorized sparse approximate inverse preconditionings I. Theory. *SIAM Journal on Matrix Analysis and Applications*, 14(1):45–58, 1993. DOI: 10.1137/0614004.
- [29] Moritz Kreutzer, Georg Hager, Gerhard Wellein, Holger Fehske, und Alan R. Bishop. A unified sparse matrix data format for modern processors with wide SIMD units. Technischer Bericht 1307.6209, arXiv, Juli 2013.
- [30] Xiaoye S. Li und Meiyue Shao. A supernodal approach to incomplete LU factorization with partial pivoting. *ACM Transactions on Mathematical Software*, 37(4):43:1–43:20, Februar 2011. DOI: 10.1145/1916461.1916467.
- [31] Anders Logg und Garth N. Wells. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2):20:1–20:28, April 2010. DOI: 10.1145/1731022.1731030.
- [32] Scott P. MacLachlan, J. David Moulton, und Tim P. Chartier. Robust and adaptive multigrid methods: comparing structured and algebraic approaches. *Numerical Linear Algebra with Applications*, 19(2):389–413, März 2012. DOI: 10.1002/nla.837.
- [33] Paulius Micikevicius. 3D finite-difference computation on GPUs using CUDA. In *GPGPU-2: Proceedings of the 2nd Workshop on General Purpose Processing on Graphics Processing Units*, Seite 79–84, März 2009. DOI: 10.1145/1513895.1513905.
- [34] Reinhard Nabben und Cornelis Vuik. A comparison of deflation and the balancing preconditioner. *SIAM Journal on Scientific Computing*, 27(5):1742–1759, 2006. DOI: 10.1137/040608246.
- [35] Artem Napov und Yvan Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM Journal on Scientific Computing*, 34(2):A1079–A1109, 2012. DOI: 10.1137/100818509.
- [36] Yvan Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis*, 37:123–146, 2010.
- [37] Carlos Ortiz-Alemán und Roland Martin. Inversion of electrical capacitance tomography data by simulated annealing: Application to real two-phase gas–oil flow imaging. *Flow Measurement and Instrumentation*, 16(2–3):157–162, April 2005. DOI: 10.1016/j.flowmeasinst.2005.02.014.
- [38] Carlos Ortiz-Alemán und Roland Martin. Two-phase flow oil-gas pipe flow imaging by simulated annealing. *Journal of Geophysics and Engineering*, 2(1):32–37, März 2005. DOI: 10.1088/1742-2132/2/1/005.
- [39] Carlos Ortiz-Alemán, Roland Martin, und José Carlos Gamio. Reconstruction of permittivity images from capacitance tomography data by using very fast simulated annealing. *Measurement and Science Technology*, 15(7):1382–1390, Juli 2004. DOI: 10.1088/0957-0233/15/7/022.

- [40] Eric Polizzi und Ahmed H. Sameh. A parallel hybrid banded system solver: the SPIKE algorithm. *Parallel Computing*, 32(2):177–194, Februar 2006. DOI: 10.1016/j.parco.2005.07.005.
- [41] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 3. Auflage, 2003.
- [42] Valeria Simoncini und Daniel B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25(2):454–477, Februar 2003. DOI: 10.1137/S1064827502406415.
- [43] Orlando Soto, Rainald Löhner, und Fernando Camelli. A linelet preconditioner for incompressible flow solvers. *International Journal of Numerical Methods for Heat & Fluid Flow*, 13(1):133–147, 2003. DOI: 10.1108/09615530310456796.
- [44] Klaus Stüben. Algebraic multigrid (AMG): An introduction with applications. Technischer Bericht 70, German National Research Center for Information Technology (GMD), Institute for Algorithms and Scientific Computing (SCAI), November 1999.