TECHNISCHE UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:
Methoden und Werkzeuge für die Nutzung
paralleler Rechnerarchitekturen

Hierarchische Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik

Hans-Joachim Bungartz, Ralf Ebner und Stefan Schulte

TUM-I9708
SFB-Bericht Nr. 342/05/97 A
April 97

TUM–INFO–04-I9708-200/1.–FI

Anforderungen an: Prof. Dr. A. Bode
Sprecher SFB 342
Institut für Informatik
Technische Universität München
D-80290 München, Germany

Druck: Fakultät für Informatik der
Technischen Universität München

Hierarchische Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik

Hans-Joachim Bungartz[1], Ralf Ebner[1] und Stefan Schulte[2]

[1] Institut für Informatik der TU München, 80290 München

[2] Siemens AG Erlangen, ZT AN 2, 91050 Erlangen

Zusammenfassung

Bei der numerischen Simulation gekoppelter Probleme haben sich modulare Ansätze als vorteilhaft erwiesen. Hier erfolgt die Lösung der betreffenden Teilprobleme verteilt und getrennt von der Behandlung der koppelnden Bedingungen. Als ein einfaches Beispiel gekoppelter Probleme können mittels eines Gebietszerlegungs- oder Substrukturierungsansatzes behandelte Probleme der Strukturmechanik angesehen werden, bei denen auf jedem Interface als Koppelbedingung ein Kräftegleichgewicht hergestellt werden muß.

In der vorliegenden Arbeit wird ein entsprechender Substrukturierungsansatz untersucht, wobei durch die strikt modulare Formulierung auf den jeweiligen Teilgebieten auch unterschiedliche Diskretisierungsstrategien bzw. Löser verteilt eingesetzt werden können. Zur Beschleunigung der Konvergenz des Koppelalgorithmus' werden die koppelnden Gleichungen mittels hierarchischer Basen diskretisiert.

1 Einleitung

Mit den gestiegenen Möglichkeiten der numerischen Simulation in Technik und Naturwissenschaften wachsen auch die Anforderungen – sowohl hinsichtlich der Genauigkeit der Rechnungen als auch im Hinblick auf die Komplexität der zu behandelnden Probleme. Immer mehr rücken dabei nicht nur einzelne Bauteile, sondern ganze Systeme in den Vordergrund, die ihrerseits wieder aus Komponenten oder Subsystemen bestehen. Diese Entwicklung kann vor allem im Bereich der Mikrotechnologie beobachtet werden, wo das Zusammenspiel unterschiedlicher physikalischer Effekte wie Strömungsmechanik, Strukturmechanik, Wärmeleitung oder Elektrostatik gezielt zur Konstruktion von Sensoren und Aktoren herangezogen wird. U. a. dadurch gewinnt die numerische Simulation solcher Wechselwirkungen immer mehr an Bedeutung. Aus Gründen der besseren Ausnutzung von Expertenwissen, der schnelleren und flexibleren Modellierung, des möglichen Austauschs einzelner Simulationsbausteine sowie des leichteren Zugangs zu einer verteilten Bearbeitung erweisen sich hierbei modulare Ansätze oftmals als vorteilhaft.

Die Zerlegung eines komplexen Problems in leichter behandelbare Teilprobleme, die bei Wechselwirkungen unterschiedlicher physikalischer Disziplinen bisher nur sporadisch eingesetzt und erst in jüngster Zeit eingehend untersucht wird [3], hat bei der Bearbeitung komplexer strukturmechanischer Aufgabenstellungen eine lange Tradition. Die grundlegende Technik der Gebietszerlegung geht auf H. A. Schwarz (1870) zurück und wird heute in zahlreichen Varianten eingesetzt, etwa in Form der Schurkomplement- und Kapazitätsmatrixmethode [6] oder als Methode der (rekursiven) Substrukturierung [11, 12]. Für einen Überblick über Gebietszerlegungsverfahren sei auf [13] verwiesen. Für die Realisierung solcher Ansätze ist der Zugang zu der im Raum diskretisierten Form der Probleme (d. h. Steifigkeits-, Dämpfungs- und Massenmatrix) notwendig. Dies wiederum erfordert eine Eingriffsmöglichkeit in den Quellcode vorhandener (kommerzieller) Programme bzw. die Existenz entsprechender Schnittstellen, welche aber nur von wenigen Programmsystemen in der benötigten Form angeboten werden (z. B. MSC/NASTRAN in Form der DMAP, [5]).

Ein modularer Substrukturierungsansatz bietet nun die Möglichkeit, diese Nachteile zu überwinden und zudem auch zu einem einfachen Aufbau hybrider Verfahren, z. B. Kopplungen zwischen Finite-Elemente-Methoden (FEM) und Randelemente-Methoden (BEM), welche für spezielle Anwendungsgebiete wie etwa die Modellierung von Gebäude-Boden-Wechselwirkungen [9] entscheidende Vorteile bieten, zu gelangen. Bei entsprechender algorithmischer Realisierung kann ein solches hybrides Verfahren auch als eine Methode zur Kopplung beliebiger Teilproblemlöser aufgefaßt werden. Diese Teilproblemlöser können nun so ausgewählt werden, daß sie optimal an die Behandlung der speziellen Aufgaben angepaßt sind, wodurch eine Verbesserung der Effizienz der Gesamtproblemlösung erzielt werden kann gegenüber der Verwendung eines homogenen Lösungsverfahrens für das Gesamtproblem.

Neben der Möglichkeit des Einsatzes optimaler Teilproblemlöser bieten Substrukturierungsansätze aber auch ein großes Potential hinsichtlich einerseits einer verteilten Bearbeitung bei der Modellierung und andererseits einer verteilten Berechnung bei anschließenden Simulationsläufen. In [11] wurde die Substrukturtechnik unter dem Aspekt betrachtet, daß sie die Abbildung der Teamarbeit im Entwicklungsprozeß komplexer Anlagen und Komponenten auf ein entsprechendes Rechnernetz („Team von Rechnern") ermöglicht.

Um die Vorteile der verteilten Berechnung nutzen und hybride Verfahren aufbauen zu können, welche es ermöglichen, beliebige, beim jeweiligen Anwender vorhandene Programme zur Teilproblemlösung einzusetzen, ist die Bereitstellung eines strikt modular aufgebauten Kopplungsverfahrens erforderlich, das ohne Eingriffe in den Quellcode der Programme auskommt. Dies ist möglich, wenn zur Kopplung nur solche Informationen genutzt werden, die von den Programmen standardmäßig geliefert werden. Der im folgenden vorgestellte Algorithmus benutzt für die Kopplung der Teilproblemlöser nur die Ergebnisse dieser Löser, wie z. B. Kräfte und Verschiebungen auf dem gemeinsamen Gebietsrand, und ist somit vom speziellen Typ des Teilproblemlösers völlig unabhängig. Insbesondere ergibt sich somit auch die Möglichkeit, analytische und numerische Lösungsansätze zu koppeln.

Durch die Trennung von Teilproblemlösung und Behandlung der eigentlichen Kopplung fällt letzterer besonderes Gewicht zu. Dies gilt insbesondere aufgrund der erwünschten Austauschbarkeit von Teilproblemlösern, wodurch Effizienzsteigerungen nur bei der Diskretisierung der Kopplungsbedingungen möglich werden. Bei FEM-Diskretisierungen spielt dabei neben dem Iterationsverfahren zur Lösung der entstehenden Gleichungssysteme die Wahl einer geeigneten Basis eine entscheidende Rolle. Hier haben sich hierarchische Basen als vorteilhaft erwiesen [2, 4, 14, 15], da sie neben Vorteilen bei der Konvergenzgeschwindigkeit auch die einfache Realisierung von Adaptivität gestatten. Erste Untersuchungen zur expliziten Verwendung hierarchischer Basen bei der Methode der Substrukturierung wurden in [7] vorgestellt.

Im folgenden besprechen wir in Abschnitt 2 kurz das Prinzip des modularen Kopplungsverfahrens aus [1]. Die Hierarchisierung der koppelnden Gleichungen wird in Abschnitt 3 behandelt. Die Vorteile der Hierarchisierung werden anschließend in Abschnitt 4 anhand eines numerischen Beispiels studiert. Ein Ausblick auf das Anwendungspotential der in dieser Studie eingesetzten Technik und zusammenfassende Bemerkungen schließen die Arbeit ab.

2 Ein modulares Kopplungsverfahren

Im folgenden soll das Grundprinzip des im Rahmen der Diplomarbeit [1] implementierten und analysierten Gebietszerlegungsverfahrens vorgestellt werden. Der vorliegende Bericht beschränkt sich auf lineare Gesamtprobleme, welche mittels eines Gebietszerlegungsansatzes in zwei Teilprobleme aufgespalten werden. Das vorgestellte Verfahren ist jedoch nicht auf diese Problemklasse beschränkt und kann, wie in [1] dargestellt, auf die Kopplung von linearen und nichtlinearen Teilproblemen sowie auf die Zerlegung in eine beliebige Anzahl von Teilproblemen ausgedehnt werden.

Ausgangspunkt des hier vorgeschlagenen Verfahrens ist eine zur Schurkomplementmethode [13] analoge Zerlegung des Gesamtproblems in Teilprobleme gemäß Abb. 1. Die Zerlegung kann dabei sowohl auf natürliche Weise (z. B. bei aus mehreren Komponenten zusammengesetzten Bauteilen) als auch rein synthetisch erfolgen (etwa beim Einsatz der Gebietszerlegung als bloße Parallelisierungsstrategie).

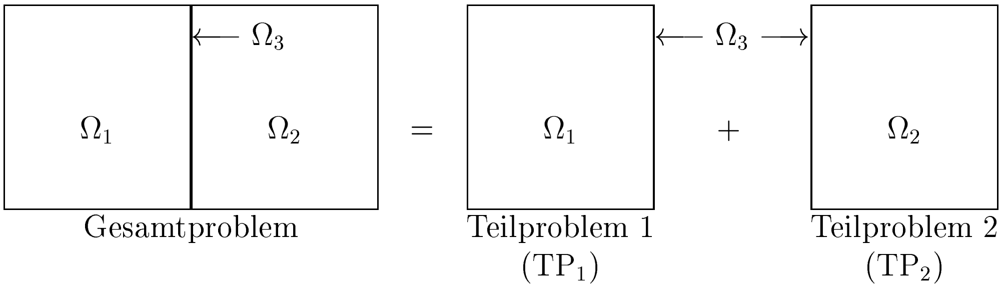


Abb.1: Gebietszerlegungsansatz zur Zerlegung des Gesamtproblems in Teilprobleme

Nach Anwendung einer Diskretisierungstechnik, z. B. der Finite-Elemente-Methode (FEM) oder der Randintegralelemente-Methode (BEM), haben die beiden Teilprobleme dann jeweils die Gestalt eines blockstrukturierten linearen Gleichungssystems:

$$\mathrm{TP}_1\colon \quad \begin{pmatrix} A_{11} & A_{13} \\ A_{31} & (A_{33})_1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ (b_3)_1 \end{pmatrix}, \tag{1}$$

$$\mathrm{TP}_2\colon \quad \begin{pmatrix} (A_{33})_2 & A_{32} \\ A_{23} & A_{22} \end{pmatrix} \begin{pmatrix} u_3 \\ u_2 \end{pmatrix} = \begin{pmatrix} (b_3)_2 \\ b_2 \end{pmatrix}. \tag{2}$$

Bei Vorgabe des Verschiebungsvektors u_3 auf dem Interface Ω_3 lassen sich die Teilprobleme im Inneren der jeweiligen Gebiete lösen:

$$u_1 := A_{11}^{-1}(b_1 - A_{13}u_3), \tag{3}$$

$$u_2 := A_{22}^{-1}(b_2 - A_{23}u_3). \tag{4}$$

Der wesentliche Rechenaufwand bei der Bestimmung von u_1 und u_2 gemäß (3) und (4) entsteht durch die Invertierung der Matrizen A_{11} und A_{22}, d. h. durch die Lösung der Teilprobleme, wobei die jeweilige Belastung aus einer Kombination aus externen Lasten (repräsentiert durch b_1 bzw. b_2) und aus der Vorgabe der Verschiebung am Interface u_3 (der sog. Verschiebungslast) resultiert.

Die Reaktionskräfte auf dem Interface ergeben sich nun unter Verwendung der Lösungen u_1 und u_2 sowie der vorgegebenen Verschiebung u_3:

$$f_1 := (b_3)_1 - A_{31}\, u_1 - (A_{33})_1\, u_3, \tag{5}$$

$$f_2 := (b_3)_2 - A_{32}\, u_2 - (A_{33})_2\, u_3. \tag{6}$$

Die gesuchte Lösung des Gesamtproblems ist dadurch gekennzeichnet, daß sich die aus (5) bzw. (6) berechneten Reaktionskräfte f_1 und f_2 zu Null addieren müssen, um die Forderung nach einem Kräftegleichgewicht zwischen linkem und rechtem Schnittufer zu erfüllen. Die Summe der Reaktionskräfte soll im folgenden als *Residuum der Reaktionskräfte* bzw. kurz als *Residuum* bezeichnet werden. Formal läßt sich das Residuum also als eine Funktion der angenommenen Verschiebungen u_3 auf dem Interface auffassen:

$$r(u_3) := f_1(u_3) + f_2(u_3). \tag{7}$$

Bei dem betrachteten linearen Problem ergibt sich aus den Gleichungen (3) bis (7) für das Residuum eine lineare Funktion

$$r(u_3) =: Ku_3 - c \tag{8}$$

bzw., mit $v := u_3$, die Residuengleichung

$$r(v) = Kv - c. \tag{9}$$

Die Matrix K, deren Größe durch die Anzahl der Freiheitsgrade auf dem Interface Ω_3 bestimmt ist, ist dabei gerade das Schurkomplement.Somit kann K aus den Matrizen A_{ij} berechnet werden (vgl. [1, 6]) und ist insbesondere symmetrisch und positiv definit.

Die gesuchte Lösung v wird nun aus der Forderung

$$r(v) = 0 \tag{10}$$

ermittelt, indem ein geeignetes Iterationsverfahren zur Anwendung kommt. Damit ist die Kopplung der Teilprobleme TP_1 und TP_2 aus (1) bzw. (2) auf ein Interface-Problem für v reduziert. Gemäß dem Substrukturierungsansatz (vgl. [7, 11]) ist nun anstelle des Gesamtproblems nur die Berechnung der Werte auf dem Interface erforderlich. Die vollständigen Lösungen der Teilprobleme geschehen unabhängig voneinander und können bei Bedarf auf mehrere Rechner verteilt werden. In Abb. 2 ist die Vorgehensweise bei dem untersuchten Algorithmus schematisch zusammengefaßt.

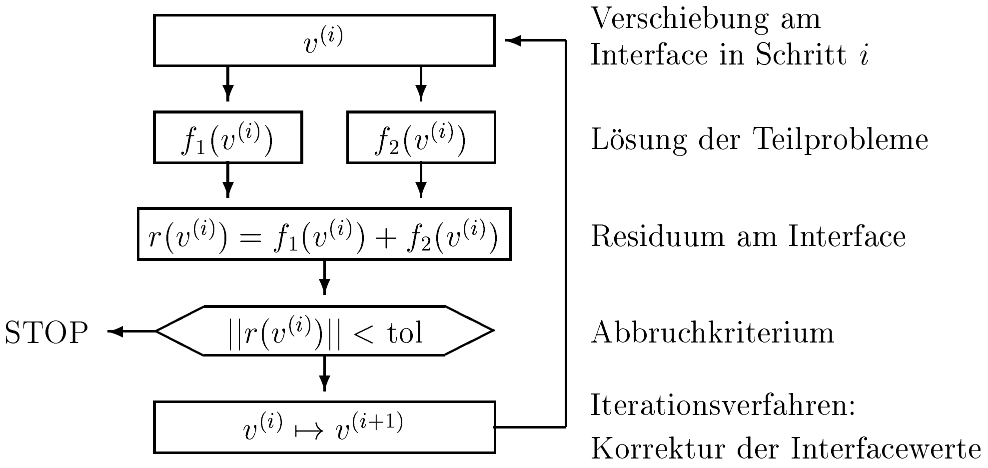


Abb.2: Flußdiagramm des modularen Kopplungsverfahrens

Da die Matrizen $A_{ij}, i \neq j$ und damit die Matrix K nicht explizit bekannt sind, kann zur Lösung der Residuengleichung (9) z. B. ein konjugiertes Gradientenverfahren verwenden. Dies ist wegen der speziellen Form der Residuengleichung (9) möglich. Die mit einem derartigen Verfahren erzielten Ergebnisse sind in [1] dargestellt und sollen im im folgenden kurz zusammengestellt werden.

Der vorgeschlagene Ansatz läßt sich vollkommen modular realisieren, d. h., es können für die Berechnung der Reaktionskräfte auf dem Interface gemäß (5) bzw. (6) bei der Bestimmung von u_1 und u_2 in (3) und (4) beliebige Teilproblemlöser verwendet werden. Insbesondere ist damit die Möglichkeit zur Realisierung hybrider Verfahren, d. h. zur Kopplung unterschiedlicher Diskretisierungstechniken bzw. unterschiedlicher Programme, gegeben.

Die Gesamteffizienz des Verfahrens hängt im wesentlichen von zwei Punkten ab:

1. vom Mehraufwand bei den jeweiligen Programmneustarts zur Berechnung der Reaktionskräfte und
2. von der benötigten Anzahl der Iterationen bis zum Erreichen des Abbruchkriteriums.

Die Programmneustarts sind in erster Linie ein Problem der verfügbaren Software-Techniken. Ein gewisser Teil des Mehraufwands, z. B. die mehrfache Invertierung der Matrizen A_{11} und A_{22}, läßt sich durch Ausnutzung von Neustarttechniken reduzieren, mit deren Hilfe die Gleichungen (3) und (4) für mehrere Verschiebungen u_3 gelöst werden. Derartige Möglichkeiten bieten heute bereits viele kommerzielle Programme, auf jeden Fall sollte von seiten der jeweiligen Programmanbieter die Bereitstellung entsprechender Schnittstellen möglich sein.

Zur Effizienzsteigerung des beschriebenen Ansatzes sind zwei Maßnahmen erforderlich. Zum einen sollte ein gegebenes Gesamtproblem derart unterteilt werden, daß die resultierenden Schnittflächen (Interfaces) möglichst wenig Knoten aufweisen, um so die Dimension des Interface-Problems niedrig zu halten. Zum anderen muß das zur Lösung der Residuengleichung (9) eingesetzte Iterationsverfahren beschleunigt werden, etwa durch den Übergang zu (multiplikativen) Multilevel-Techniken oder durch eine entsprechende Vorkonditionierung.

Schließlich ist in [1] ausgeführt, wie das hier vorgeschlagene Verfahren auch auf die Behandlung instationärer Probleme angewendet werden kann. Ebenso ist eine Kombination von linearen und nichtlinearen Teilproblemen möglich.

3 Effizienzsteigerung durch Transformation auf hierarchische Basen

Unter der Voraussetzung, daß ein expliziter Zugang zu den Matrizen A_{ij} aus Gleichung (1) und (2) möglich ist, lassen sich aus den Inversen von A_{11} und A_{22} effiziente Vorkonditionierer für das Schurkomplement gewinnen [6].

Eine allgemeine Möglichkeit zur substantiellen Verbesserung des Konvergenzverhaltens iterativer Löser für Finite-Elemente-Probleme ist durch den Wechsel von den standardmäßig verwendeten Knotenpunktbasen auf hierarchische Basen gegeben [2, 4, 7, 8, 14, 15]. In [7] etwa wird durch eine explizite Diskretisierung mittels hierarchischer Basen eine deutliche Konvergenzbeschleunigung für eine iterative Variante des in [11] vorgeschlagenen Verfahrens der rekursiven Substrukturierung erzielt. Außerdem bietet ein solcher Ansatz Möglichkeiten zu einem direkten Einbau von (multiplikativen) Multilevel-Algorithmen, zum Einsatz selektiver Relaxationstechniken (Gauß-Southwell) sowie zu einer einfachen adaptiven Verfeinerung der zugrundeliegenden Gitter. Will man auf den expliziten Einsatz hierarchischer Basen verzichten, so eröffnet eine hierarchische Vorkonditionierung den Zugang zu (additiven) Multilevel-Techniken. Da die Herleitung solcher Vorkonditionierer auch für das Schurkomplement K aus dem Interface-Problem (9) leicht und ohne explizite Verwendung der Teilmatrizen aus (1) und (2) möglich ist, wurde für die vorliegende Arbeit dieser Weg gewählt.

Im folgenden wird die Technik der Hierarchisierung (vgl. [2, 4, 14, 15]) von eindimensionalen Funktionen kurz eingeführt und gezeigt, wie sie für unser Beispielproblem aus Abb. 1 auf die iterative Lösung der Residuengleichung (9) angewendet werden kann.

3.1 Funktionen in hierarchischer Darstellung

Es seien gegeben ein Grundintervall Ω sowie ein diskretes Gitter Ω_l auf Ω aus $2^l + 1$ äquidistanten Punkten. Zur näherungsweisen Darstellung einer Funktion $f : \Omega \mapsto \mathbb{R}$ mittels eines bzgl. Ω_l stückweise linearen Interpolanten f_l verwendet man üblicherweise die *Knotenpunktbasis* B_l. Sie besteht aus den $2^l + 1$ bzgl. Ω_l stückweise linearen Funktionen φ_i, $i = 0, \ldots 2^l$, die jeweils an genau einem Stützpunkt den Wert 1 und an allen anderen Stützpunkten den

Wert 0 annehmen. Entscheidend ist nun, daß der von B_l aufgespannte Raum auch mit Hilfe einer *hierarchischen Basis* H_l erzeugt werden kann. Diese ist wie folgt rekursiv definiert:

$$H_0 := V_0 := B_0, \qquad\qquad H_l := H_{l-1} \cup V_l = \bigcup_{i=0}^{l} V_i. \tag{11}$$

Dabei ist $V_l \subset B_l$ die Menge der Basisfunktionen, die auf den zu B_{l-1} gehörenden Gitterpunkten verschwinden. In Abb. 3 ist links die Knotenpunktbasis B_3 und rechts die hierarchische Basis H_3 dargestellt.

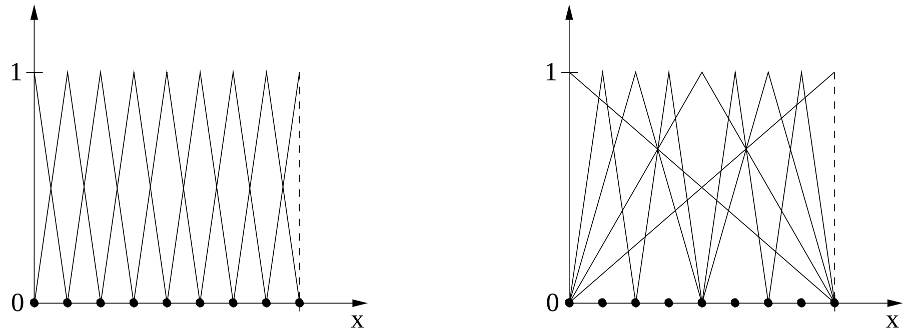


Abb.3: Knotenpunktbasis B_3 (links) und hierarchische Basis H_3 (rechts)

Eine Funktion f_l aus dem von B_l aufgespannten Raum kann nun als Linearkombination der Knotenbasisfunktionen $\varphi_i \in B_l$,

$$f_l(x) = \sum_{i=0}^{2^l} v_i \varphi_i(x), \quad v_i \in \mathbb{R}, \tag{12}$$

oder gleichwertig als Linearkombination von hierarchischen Basisfunktionen $\tilde{\varphi}_i \in H_l$,

$$f_l(x) = \sum_{i=0}^{2^l} \tilde{v}_i \tilde{\varphi}_i(x), \quad \tilde{v}_i \in \mathbb{R}, \tag{13}$$

eindeutig dargestellt werden. Für die Interpolierende einer Parabel und $l = 3$ ergeben sich die Koeffizienten v_i bzw. $\tilde{v}_i$ aus Abb. 4.

Während die Koeffizienten v_i der Knotenbasisdarstellung in der gleichen Größenordnung liegen, fallen die Koeffizienten $\tilde{v}_i$ für genügend glatte Funktionen geometrisch ab.

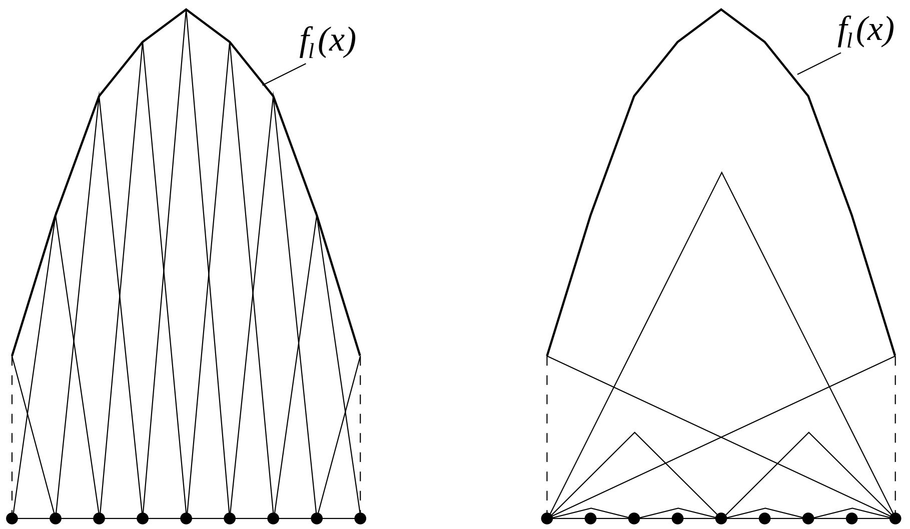


Abb.4: Interpolierende einer Parabel in Knotenpunkt-Basis und in hierarchischer Basis

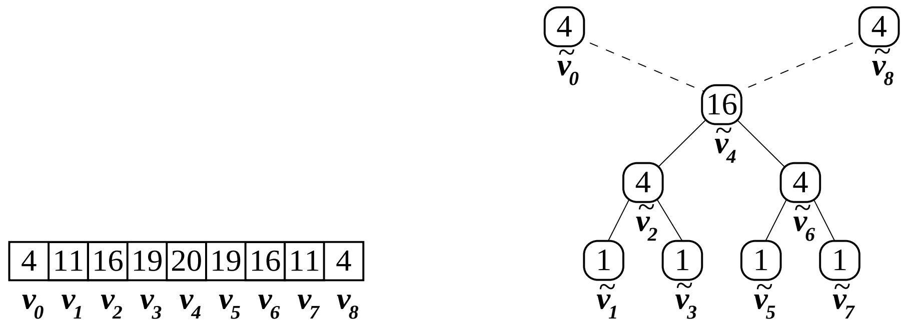


Abb.5: Datenstrukturen für die Speicherung von v bzw. $\tilde{v}$ $(L = 3)$

Als Datenstruktur für den Koeffizientenvektor $(v_0, \ldots, v_n)$ wird üblicherweise ein Array verwendet. Um der Hierarchie der Basisfunktionen Rechnung zu tragen, hält man die *hierarchischen Überschüsse* $(\tilde{v}_0, \ldots, \tilde{v}_n)$ dagegen in einem Binärbaum gespeichert, siehe Abb. 5.

Da aber Löser aus numerischen Bibliotheken in der Regel nur auf Arrays arbeiten, kann man alternativ die $\tilde{v}_i$ in einem Array lexikographisch gespeichert belassen und Baumdurchläufe — wie sie z. B. bei der im nächsten Abschnitt dargestellten hierarchischen Transformation nötig sind — durch geeignete Indexrechnung nachbilden. Sei L die Tiefe das Baums. Alle Knoten der Ebene l des Baums werden besucht, indem man mit dem Startindex 2^{L-l} beginnt und die Schrittweite 2^{L-l+1} wählt.

3.2 Hierarchisierung und Dehierarchisierung

Für das modulare Kopplungsverfahren sind auf dem Interface die hierarchischen Transformationen *Hierarchisierung* und *Dehierarchisierung* erforderlich, d. h. also die Transformationen von $v = (v_0, \dots, v_n)^T$ nach $\tilde{v} = (\tilde{v}_0, \dots, \tilde{v}_n)^T$ und umgekehrt. Denn einerseits rechnen kommerzielle Finite-Elemente-Programme nur in der Knotenpunktbasis, andererseits konvergiert aber das umfassende Iterationsverfahren für das Residuengleichungssystem (9) auf dem Interface schneller, wenn es durch Hierarchisierung vorkonditioniert wird [13].

Der hierarchische Überschuß $\tilde{v}_i$ an einem Knoten ergibt sich durch Abziehen der gemittelten Werte $v_{n_1(i)}$ und $v_{n_2(i)}$ dessen *hierarchischer Nachbarn*:

$$\tilde{v}_i = v_i - (v_{n_1(i)} + v_{n_2(i)})/2. \tag{14}$$

Die Indizes der hierarchischen Nachbarn eines Knotens errechnen sich aus der halben Schrittweite des Levels l (vgl. den vorangehenden Abschnitt), in dem sich der Knoten befindet:

$$\begin{aligned} n_1(i) &:= i - 2^{L-l}, \\ n_2(i) &:= i + 2^{L-l}. \end{aligned} \tag{15}$$

Die lineare Abbildung S_i^{-1}, die diese hierarchische Transformation für den Koeffizienten v_i in $(v_0, \dots, v_n)^T$ leistet, ist dann bestimmt durch

$$S_i^{-1} \begin{pmatrix} \vdots \\ v_{n_1(i)} \\ \vdots \\ v_i \\ \vdots \\ v_{n_2(i)} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ v_{n_1(i)} \\ \vdots \\ v_i - (v_{n_1(i)} + v_{n_2(i)})/2 \\ \vdots \\ v_{n_2(i)} \\ \vdots \end{pmatrix}. \tag{16}$$

Wenn die hierarchische Transformation S^{-1} von $v = (v_0, \ldots, v_n)^T$ *in situ* erfolgen soll, muß (14) bezüglich der Baumdarstellung von v levelweise von unten nach oben, also bei den Blättern des Baums beginnend, angewendet werden. Im Fall von $l = 2$ bedeutet das

$$\tilde{v} = S^{-1}v = S_1^{-1}(S_2^{-1}(S_3^{-1}v)). \tag{17}$$

Allgemein ist die Transformation dann auszuführen als

$$\tilde{v} = (\prod_{l=1}^{L} \prod_{\substack{i=2^{L-l} \\ \text{step } 2^{L-l+1}}}^{2^l} S_i^{-1})\ v. \tag{18}$$

Zur Dehierarchisierung S überlegt man sich

$$\begin{aligned} &S = (S^{-1})^{-1} = (S_1^{-1}S_2^{-1}S_3^{-1})^{-1} = \\ &(S_3^{-1})^{-1}(S_2^{-1})^{-1}(S_1^{-1})^{-1} = S_3S_2S_1. \end{aligned} \tag{19}$$

Hier muß die Transformation an der Wurzel des Baums begonnen werden, und zwar nicht nur bei Rechnungen am Platz, da sonst die dehierarchisierten Koeffizienten $v_{n_1(i)}$ und $v_{n_2(i)}$ an den hierarchischen Nachbarn noch nicht zur Verfügung stehen.

Nach der Hierarchisierung ist anstelle des Residuengleichungssystems (9) das äquivalente System

$$S^TKS\tilde{v} = S^Tc, \tag{20}$$

zu lösen, dessen Matrix S^TKS besser konditioniert ist als K (vgl. [7, 8, 13]). Die Transformation S^T muß — bezüglich der Reihenfolge der bearbeiteten Knoten — umgekehrt wie S erfolgen, also von den Blättern des Baums hinauf zur Wurzel.

3.3 Das Kopplungsprogramm mit hierarchischen Basen

Zur Integration des Hierarchisierungsverfahren wurde das Konzept des Fortran-Programms aus [1] in einen neuen Prototyp in der Programmiersprache C umgesetzt. Anstelle des CG-Verfahrens wird jetzt das Krylov-Unterraum-Verfahren GMRES [10] für die Lösung der Residuengleichung eingesetzt. Dies hat den Vorteil, daß zum einen das Programm auch auf Probleme angewendet werden kann, die auf nichtsymmetrische Gleichungssysteme auf dem Interface führen, und daß zum anderen größtmöglicher Nutzen aus einer modernen, optimierten Numerik-Bibliothek gezogen wird.

Wichtig bei der Auswahl des GMRES-Lösers war, daß er eine benutzerdefinierte Vorkonditionierung des zu lösenden Gleichungssystems erlaubt. Damit kann die Hierarchisierung ohne Probleme integriert werden. GMRES verlangt dazu die Bereitstellung einer Funktion `msolve`, die die Hierarchisierung der rechten Seite bzw. des Residuums vornimmt:

$$\texttt{msolve}(r) = \tilde{r} = S^T r. \tag{21}$$

Weiterhin ist der Einsatz von GMRES für das neue hierarchisierte Kopplungsverfahren erst möglich, weil GMRES die Matrix des Gleichungssystems nicht explizit benötigt. Wie beim CG-Verfahren ist nur eine Funktion `matvec` für die Berechnung des Matrix-Vektor-Produkts notwendig:

$$\texttt{matvec}(v) = \tilde{K}\tilde{v} = S^T K S \tilde{v} =: \tilde{y}. \tag{22}$$

Dabei muß $\tilde{v}$ also zuerst dehierarchisiert werden ($v = S\tilde{v}$). Für die Berechnung von Kv kommen gemäß (7) und (9) die Teilproblemlöser zum Einsatz, welche die Reaktionskräfte f_1 und f_2 für die Punkte des Interface liefern:

$$Kv = c + f_1 + f_2 =: y. \tag{23}$$

Da diese Berechnung in Knotenbasis-Darstellung erfolgt, ergeben sich durch das hierarchisierte Kopplungsverfahren keine Einschränkungen für die Verwendung der Teilproblemlöser. Schließlich ist noch die Hierarchisierung des Matrix-Vektor-Produkts erforderlich:

$$\tilde{y} = S^T y \tag{24}$$

Als Flußdiagramm ist das hierarchisierte Kopplungsverfahren in Abb. 6 dargestellt.

Mit dieser Vorgehensweise ist also weder die Kenntnis der Matrix K noch die explizite Aufstellung der Hierarchisierungs-Matrizen erforderlich, was sich entscheidend auf die Speichereffizienz und die möglichst universelle Einsetzbarkeit des vorgestellten Kopplungsverfahrens auswirkt.

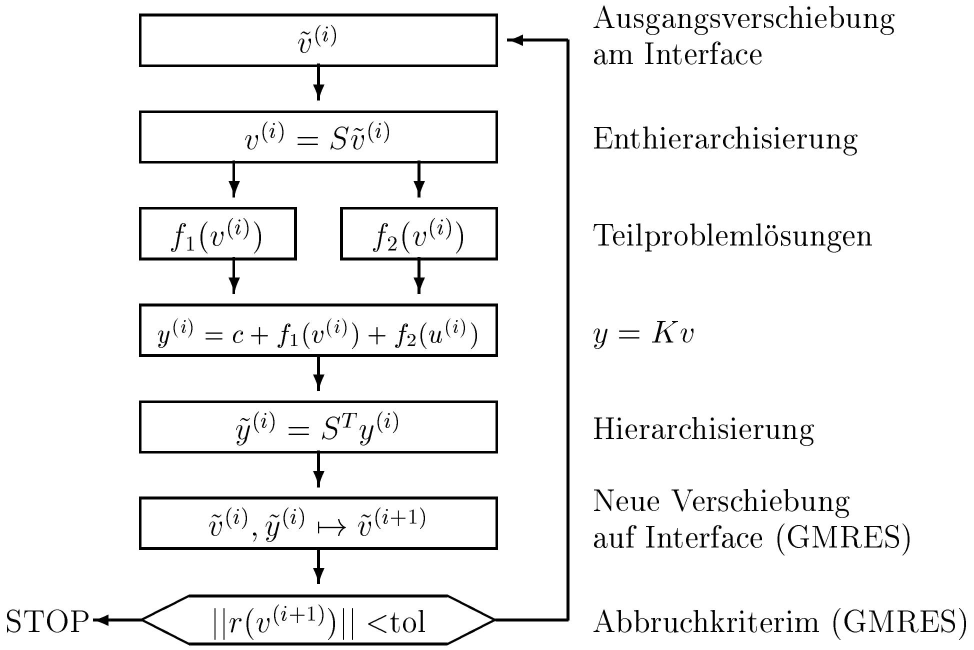


Abb.6: Flußdiagramm des hierarchisierten modularen Kopplungsverfahrens

4 Numerische Beispielrechnungen

Als Beispielproblem wurde eine Kragscheibe mit den Abmessungen[1] $y \in [0,2]$ und $z \in [0,1]$ gewählt (Abb. 7). Es interessiert die Verformung der Kragscheibe, wenn auf deren oberer Kante eine Last aufgebracht wird.

Der Zusammenhang zwischen dem Level l und der Anzahl der Knoten auf dem Interface n_{int} im Beispiel von Abb. 1 ist gegeben durch die Beziehung

$$n_{int} = 2^l + 1 \tag{25}$$

Da in jedem Knoten zwei Werte (horizontale und vertikale Verschiebung) gehalten werden, gilt dann für die Anzahl N der Unbekannten auf dem Interface:

$$N = 2 \cdot (2^l + 1) \tag{26}$$

In der Implementierung des beschriebenen Kopplungsverfahrens wurde zur

[1] Das verwendete FE-Programm behandelt 2-dim. Probleme grundsätzlich in der y-z-Ebene, weshalb hier auch y und z gewählt wurden.

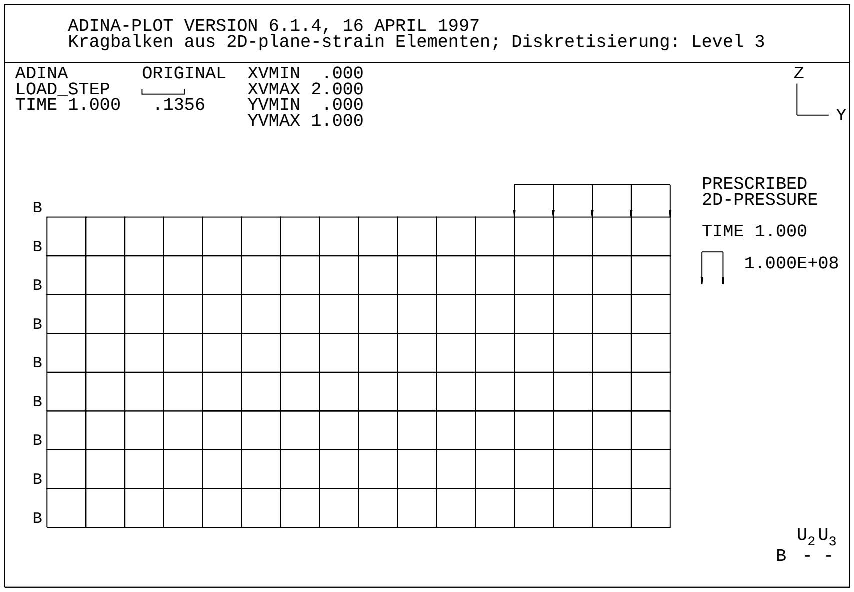


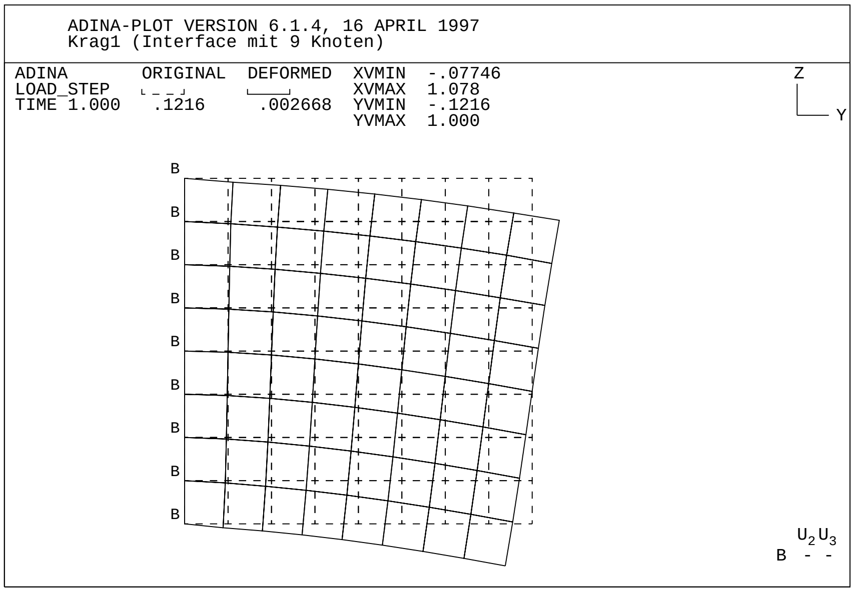
Abb. 7: Die zu untersuchende Kragscheibe ohne Last

Lösung der Interface-Gleichung (9) der GMRES-Löser `DGMRES` von P. Brown, A. Hindmarsh und M.K. Seager, Lawrence Livermore National Laboratory, Livermore (CA) verwendet. Als Teilproblemlöser kam das Finite-Elemente-Programmpaket ADINA der Firma ADINA R & D, Watertown (NA), zum Einsatz.

Als Interface wird $y = 1$ festgelegt. Für den Fall $l = 3$ ($n_{int} = 9$) sind die Ergebnisse in Abb. 8 durch das in ADINA integrierte Plot-Programm dargestellt.

Die gestrichelten Linien in Abb. 8 zeigen die Kragscheibe ohne Belastung, die durchgezogenen die Verformung unter Einwirkung der Last auf die obere Kante $z = 1$, $y \in [1.5, 2]$. Die unterschiedliche Lage des Interface in den Teilberechnungen ist durch die automatische Skalierung des Plot-Programms bedingt.

Nach Verschieben des Interface des rechten Teils auf die Koordinaten des linken Teils ergibt sich insgesamt auch bildlich die korrekte Lösung, die man in einer Kontrollrechnung ohne Zerlegung in Teilprobleme (Abb. 9) erhält.



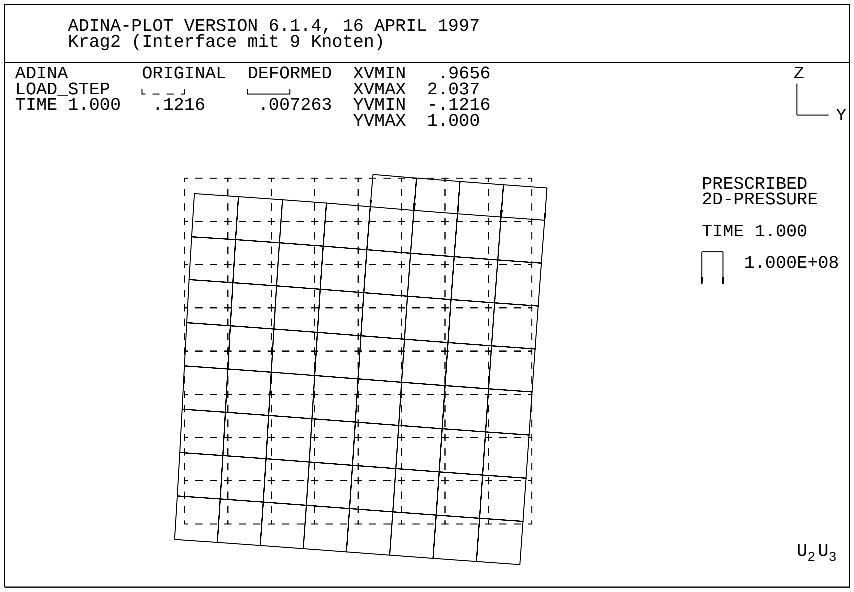


Abb.8: Ergebnisse der beiden Teilprobleme für die Simulation der Kragscheiben-Verformung unter Last

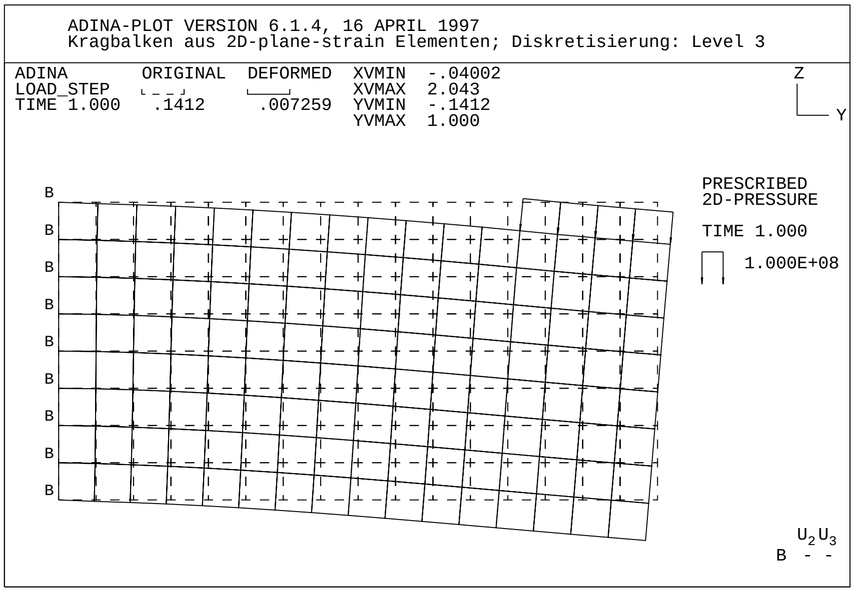


Abb.9: Ergebnis der ADINA-Berechnung ohne Zerlegung in Teilprobleme

Mit dieser Implementierung des Kopplungsverfahrens wurde nun die Anzahl der benötigten GMRES-Iterationen einmal unter Verwendung hierarchischer Basen für die Verschiebungen auf dem Interface (HB) und ohne Hierarchisierung (KPB) berechnet:

Level l	#Interface-Knoten N	#Iterationen HB	KPB
1	6	6	6
2	10	10	10
3	18	13	17
4	34	16	25
5	66	17	38
6	130	18	52
7	258	20	75

Ab Level $l = 5$ steigt die Anzahl der GMRES-Iterationen bei Verwendung der Knotenpunktbasis für die Residuengleichung (9) linear mit der Anzahl der Interfacepunkte an. Das schon in [7] durch Hierarchisierung erreichte logarithmische Anwachsen der Iterationsanzahl für einen einfachen Gauß-Seidel-

Löser ist auch bei dem hier vorgestellten Verfahren mit Vorkonditionierung und GMRES-Löser in Abb. 10 zu beobachten.

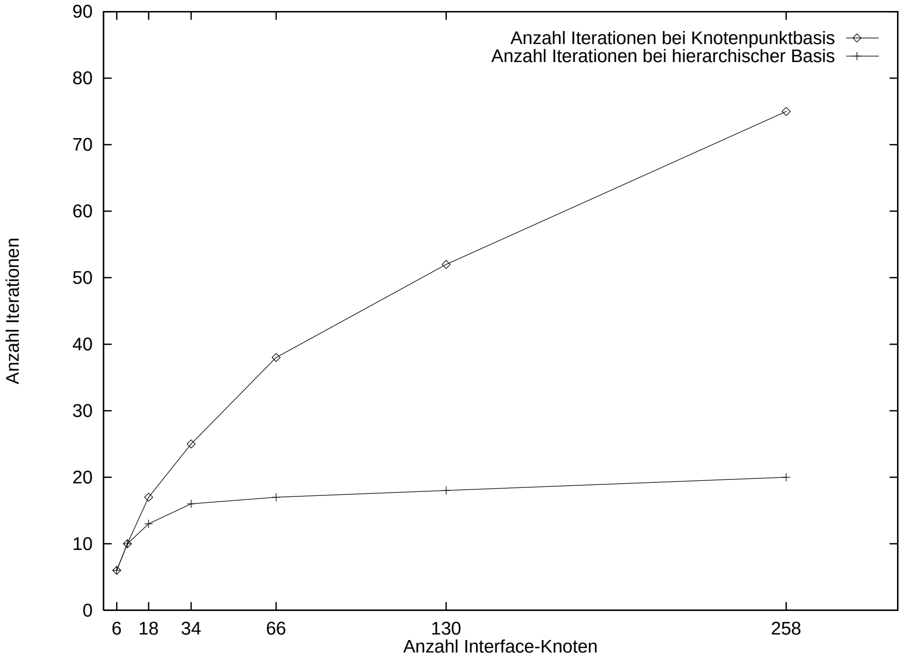


Abb.10: Anzahl der GMRES-Iterationen in Abhängigkeit von N

Es bleibt noch anzumerken, daß das GMRES-Programm ferner die Möglichkeit bietet, eine Skalierung der Diagonale vorzunehmen. Im allgemeinen ist jedoch die Diagonale des Schurkomplements nicht bekannt, so daß Techniken zur Approximation der Diagonale entwickelt werden müßten.

5 Ausblick

Auf der algorithmischen Seite wäre zu prüfen, ob die Transformation auf semidefinite Systeme, wie sie in [4] ausführlich untersucht werden, auch in der Kombination mit GMRES gegenüber den hierarchischen Basen eine nochmalige Reduktion der Anzahl der benötigten Iterationen bringt. In [4] erreicht der CG-Löser für des semidefinite System eine (optimale) konstante Anzahl von Iterationsschritten, wie sie auch mit Multilevel-Verfahren erreichbar ist.

Eine interessante Erweiterung des beschriebenen Ansatzes ist die Anwendung auf sogenannte gekoppelte Probleme [16, 17]. Ein derartiges Problem entsteht, wenn die beiden Teilprobleme aus Abb. 1 unterschiedliche physikalische Phänomene beschreiben. Beispielsweise würde im Fall einer Fluid-Struktur-Kopplung das Teilproblem 1 die Strömungsmechanik repräsentieren, und das Teilproblem 2 die Strukturmechanik.

Bei einer Fluid-Struktur-Kopplung kann völlig analog wie in Abb. 2 dargestellt vorgegangen werden. Wird also die Verschiebung des Interfaces als unabhängige Variable gewählt, so bedeutet dies aus Sicht der Strukturmechanik die Vorgabe einer Verschiebungslast am Interface; berechnet wird dann entsprechend die Kraft an diesem Interface (analog zu Gleichung (5) oder (6)). Für die Fluidmechanik folgt aus der Festlegung der Position des Interfaces die Geometrie des Strömungsgebiets, womit das fluiddynamische Problem bei Stellung entsprechender strömungsmechanischer Randbedingungen eindeutig festgelegt ist. Als Ergebnis der Strömungsanalyse wird dann die Druckverteilung am Interface geliefert, welche sich in Reaktionskräfte an den Knoten umrechnen läßt. Als koppelnde Bedingung kann nun wieder das Kraftgleichgewicht am Interface verwendet werden.

Die sich für ein Fluid-Struktur-Problem ergebende Residuengleichung ist eine nichtlineare Funktion der vorgegebenen Interface-Verschiebung, so daß das beschriebene Kopplungsverfahren entsprechend erweitert werden muß, beispielsweise durch Verwendung eines Newton-Krylov-Algorithmus'.

Ebenso ist auch eine Elekrostatik-Struktur-Kopplung realisierbar. Dann fließen die durch elektrische Felder verursachten Kräfte auf dem Interface das Kraftgleichgewicht mit ein.

Um bei Problemen der Struktur-Struktur-Wechselwirkung eine weitere Leistungssteigerung zu erzielen, wäre eine Kombination des hier beschriebenen Ansatzes mit den ARESO-Techniken aus [7, 11] denkbar. Die grobgranulare Aufteilung des Gesamtproblems auf unterschiedliche Teilproblemlöser (z. B. FEM, BEM) würde mit Hilfe des hier beschriebenen modularen Kopplungsverfahrens realisiert. Wenn der jeweilige Teilproblemlöser den Zugriff auf die Matrizen A_{ij} erlaubt, könnte das Teilproblem mit ARESO-Techniken weiter zerlegt und verteilt werden, z. B. auf Workstation-Netze oder Parallelrechner.

6 Abschließende Bemerkungen

Der in [1] untersuchte Ansatz ermöglicht die Verteilung strukturdynamischer Probleme auf beliebige Teilproblemlöser. Er basiert auf dem klassischen Gebietszerlegungsansatz und kann als eine modularisierbare Variante der Schurkomplementmethode angesehen werden; seine generelle Realisierbarkeit ist in [1] nachgewiesen.

Für eine Effizienzsteigerung ist eine Vorkonditionierungsmethode notwendig, die ohne expliziten Zugriff auf die Matrizen A_{ij} auskommt. Wir haben hier ein entsprechendes Verfahren mit Hierarchisierung des Interface-Problems entwickelt, das im Gegensatz zu [7, 11] auf die Kenntnis der Matrizen A_{ij} der Teilprobleme verzichten kann.

Das beschriebene Verfahren läßt sich auf weitere gekoppelte Probleme wie z.B. Fluid-Struktur- oder Elektrostatik-Struktur-Wechselwirkungen übertragen, wobei das zum Einsatz kommende Lösungsverfahren für die Interface-Gleichung dem jeweiligen Problem angepaßt werden muß.

Literatur

[1] Ahrens, S.: *Entwicklung eines Ansatzes zur verteilten Berechnung strukturdynamischer Probleme unter Verwendung kommerziell verfügbarer Programmsysteme.* Diplomarbeit, Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen, Universität Stuttgart, 1995. Die Diplomarbeit wurde in Kooperation mit der Siemens AG, ZFE T SE 3, durchgeführt und ist dort verfügbar.

[2] Bungartz, H.-J.: *Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung.* Dissertation, Institut für Informatik, Technische Universität München, 1992.

[3] Bungartz, H.-J., Schulte, S: *Coupled Problems in Microsystem Technology.* SFB-Bericht 342/07/95, Technische Universität München, 1995.

[4] Griebel, M.: *Multilevelmethoden als Iterationsverfahren über Erzeugendensystemen.* B. G. Teubner, Stuttgart, 1994.

[5] Cifuentes, A. O.: *Using MSC/NASTRAN.* Springer Verlag, New York, 1989.

[6] Hackbusch, W.: *Iterative Lösung großer schwach besetzter Gleichungssysteme.* Leitfäden der angewandten Mathematik und Mechanik, Band 69, B. G. Teubner, Stuttgart, 1993.

[7] Hüttl, R.: *Ein iteratives Lösungsverfahren bei der Finite-Element-Methode unter Verwendung von rekursiver Substrukturierung und hierachischen Basen.* Dissertation, Institut für Informatik, Technische Universität München, 1996.

[8] Jung, M.; Langer, U.: *Applications of Multilevel Methods to Practical Problems.* Surv. Math. Ind. (1991) 1, pp. 217–257.

[9] Löbel, G.: *Berechnungsmethoden zur Simulation der Bauwerk-Boden-Wechselwirkungen bei Erdbeben.* Laborbericht der Siemens AG, Aktenzeichen BTSE55/B11, ZFE BT SE 55, Erlangen, 1993. (nicht öffentlich)

[10] Saad, Y.; Schultz, M.: *GMRES, a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems.* SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[11] Schneider, M.: *Verteilte adaptive numerische Simulation auf der Basis der Finite-Elemente-Methode, Technische Universität München.* Dissertation, 1995.

[12] Schwarz, H. R.: *Methode der Finiten Elemente.* Teubner Studienbücher Mathematik, Stuttgart 1984.

[13] Smith, B. F.; Bjørstad, P. E.; Gropp, W. D.: *Domain Decomposition, Parallel Multilevel Methods For Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[14] Yserentant, H.: *On the Multi-Level Splitting of Finite Element Spaces.* Numer. Math., Vol. 49, pp. 379–412, 1986.

[15] Zenger, C.: *Sparse Grids.* SFB-Bericht 342/18/90, Technische Universität München, 1990.

[16] Zienkiewicz, O. C.: *Coupled Problems and their Numerical Solution.* In Numerical Methods in Coupled Systems (eds. Lewis, R. W.; Bettess, P.; Hinton, E.), chap. 1, pp. 35–68, Wiley, Chinchester, 1984.

[17] Zienkiewicz, O. C.; Chan, A. H. C.: *Coupled Problems and their Numerical Solution*. In Advances in Computational Non-Linear Mechanics (ed. L. S. Doltsinis), chap. 4, pp. 139–176, Springer Verlag, Berlin, 1988.

SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

342/1/90 A	Robert Gold, Walter Vogler: Quality Criteria for Partial Order Semantics of Place/Transition-Nets, Januar 1990
342/2/90 A	Reinhard Fößmeier: Die Rolle der Lastverteilung bei der numerischen Parallelprogrammierung, Februar 1990
342/3/90 A	Klaus-Jörn Lange, Peter Rossmanith: Two Results on Unambiguous Circuits, Februar 1990
342/4/90 A	Michael Griebel: Zur Lösung von Finite-Differenzen- und Finite-Element-Gleichungen mittels der Hierarchischen Transformations-Mehrgitter-Methode
342/5/90 A	Reinhold Letz, Johann Schumann, Stephan Bayerl, Wolfgang Bibel: SETHEO: A High-Performance Theorem Prover
342/6/90 A	Johann Schumann, Reinhold Letz: PARTHEO: A High Performance Parallel Theorem Prover
342/7/90 A	Johann Schumann, Norbert Trapp, Martin van der Koelen: SETHEO/PARTHEO Users Manual
342/8/90 A	Christian Suttner, Wolfgang Ertel: Using Connectionist Networks for Guiding the Search of a Theorem Prover
342/9/90 A	Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Jaroslav Kremenek, Robert Lindhof, Thomas Ludwig, Peter Luksch, Thomas Treml: TOPSYS, Tools for Parallel Systems (Artikelsammlung)
342/10/90 A	Walter Vogler: Bisimulation and Action Refinement
342/11/90 A	Jörg Desel, Javier Esparza: Reachability in Reversible Free- Choice Systems

Reihe A

342/12/90 A	Rob van Glabbeek, Ursula Goltz: Equivalences and Refinement
342/13/90 A	Rob van Glabbeek: The Linear Time - Branching Time Spectrum
342/14/90 A	Johannes Bauer, Thomas Bemmerl, Thomas Treml: Leistungsanalyse von verteilten Beobachtungs- und Bewertungswerkzeugen
342/15/90 A	Peter Rossmanith: The Owner Concept for PRAMs
342/16/90 A	G. Böckle, S. Trosch: A Simulator for VLIW-Architectures
342/17/90 A	P. Slavkovsky, U. Rüde: Schnellere Berechnung klassischer Matrix-Multiplikationen
342/18/90 A	Christoph Zenger: Sparse Grids
342/19/90 A	Michael Griebel, Michael Schneider, Christoph Zenger: A combination technique for the solution of sparse grid problems
342/20/90 A	Michael Griebel: A Parallelizable and Vectorizable Multi- Level-Algorithm on Sparse Grids
342/21/90 A	V. Diekert, E. Ochmanski, K. Reinhardt: On confluent semi-commutations-decidability and complexity results
342/22/90 A	Manfred Broy, Claus Dendorfer: Functional Modelling of Operating System Structures by Timed Higher Order Stream Processing Functions
342/23/90 A	Rob van Glabbeek, Ursula Goltz: A Deadlock-sensitive Congruence for Action Refinement
342/24/90 A	Manfred Broy: On the Design and Verification of a Simple Distributed Spanning Tree Algorithm
342/25/90 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Peter Luksch, Roland Wismüller: TOPSYS - Tools for Parallel Systems (User's Overview and User's Manuals)
342/26/90 A	Thomas Bemmerl, Arndt Bode, Thomas Ludwig, Stefan Tritscher: MMK - Multiprocessor Multitasking Kernel (User's Guide and User's Reference Manual)
342/27/90 A	Wolfgang Ertel: Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems
342/28/90 A	Rob van Glabbeek, Frits Vaandrager: Modular Specification of Process Algebras

342/29/90 A	Rob van Glabbeek, Peter Weijland: Branching Time and Abstraction in Bisimulation Semantics
342/30/90 A	Michael Griebel: Parallel Multigrid Methods on Sparse Grids
342/31/90 A	Rolf Niedermeier, Peter Rossmanith: Unambiguous Simulations of Auxiliary Pushdown Automata and Circuits
342/32/90 A	Inga Niepel, Peter Rossmanith: Uniform Circuits and Exclusive Read PRAMs
342/33/90 A	Dr. Hermann Hellwagner: A Survey of Virtually Shared Memory Schemes
342/1/91 A	Walter Vogler: Is Partial Order Semantics Necessary for Action Refinement?
342/2/91 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Rainer Weber: Characterizing the Behaviour of Reactive Systems by Trace Sets
342/3/91 A	Ulrich Furbach, Christian Suttner, Bertram Fronhöfer: Massively Parallel Inference Systems
342/4/91 A	Rudolf Bayer: Non-deterministic Computing, Transactions and Recursive Atomicity
342/5/91 A	Robert Gold: Dataflow semantics for Petri nets
342/6/91 A	A. Heise; C. Dimitrovici: Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften
342/7/91 A	Walter Vogler: Asynchronous Communication of Petri Nets and the Refinement of Transitions
342/8/91 A	Walter Vogler: Generalized OM-Bisimulation
342/9/91 A	Christoph Zenger, Klaus Hallatschek: Fouriertransformation auf dünnen Gittern mit hierarchischen Basen
342/10/91 A	Erwin Loibl, Hans Obermaier, Markus Pawlowski: Towards Parallelism in a Relational Database System
342/11/91 A	Michael Werner: Implementierung von Algorithmen zur Kompaktifizierung von Programmen für VLIW-Architekturen
342/12/91 A	Reiner Müller: Implementierung von Algorithmen zur Optimierung von Schleifen mit Hilfe von Software-Pipelining Techniken
342/13/91 A	Sally Baker, Hans-Jörg Beier, Thomas Bemmerl, Arndt Bode, Hubert Ertl, Udo Graf, Olav Hansen, Josef Haunerdinger, Paul Hofstetter, Rainer Knödlseder, Jaroslav Kremenek, Siegfried Langenbuch, Robert Lindhof, Thomas Ludwig, Peter Luksch, Roy Milner, Bernhard Ries, Thomas Treml: TOPSYS - Tools for Parallel Systems (Artikelsammlung); 2., erweiterte Auflage

Reihe A

342/14/91 A	Michael Griebel: The combination technique for the sparse grid solution of PDE's on multiprocessor machines
342/15/91 A	Thomas F. Gritzner, Manfred Broy: A Link Between Process Algebras and Abstract Relation Algebras?
342/16/91 A	Thomas Bemmerl, Arndt Bode, Peter Braun, Olav Hansen, Thomas Treml, Roland Wismüller: The Design and Implementation of TOPSYS
342/17/91 A	Ulrich Furbach: Answers for disjunctive logic programs
342/18/91 A	Ulrich Furbach: Splitting as a source of parallelism in disjunctive logic programs
342/19/91 A	Gerhard W. Zumbusch: Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme
342/20/91 A	M. Jobmann, J. Schumann: Modelling and Performance Analysis of a Parallel Theorem Prover
342/21/91 A	Hans-Joachim Bungartz: An Adaptive Poisson Solver Using Hierarchical Bases and Sparse Grids
342/22/91 A	Wolfgang Ertel, Theodor Gemenis, Johann M. Ph. Schumann, Christian B. Suttner, Rainer Weber, Zongyan Qiu: Formalisms and Languages for Specifying Parallel Inference Systems
342/23/91 A	Astrid Kiehn: Local and Global Causes
342/24/91 A	Johann M.Ph. Schumann: Parallelization of Inference Systems by using an Abstract Machine
342/25/91 A	Eike Jessen: Speedup Analysis by Hierarchical Load Decomposition
342/26/91 A	Thomas F. Gritzner: A Simple Toy Example of a Distributed System: On the Design of a Connecting Switch
342/27/91 A	Thomas Schnekenburger, Andreas Weininger, Michael Friedrich: Introduction to the Parallel and Distributed Programming Language ParMod-C
342/28/91 A	Claus Dendorfer: Funktionale Modellierung eines Postsystems
342/29/91 A	Michael Griebel: Multilevel algorithms considered as iterative methods on indefinite systems

Reihe A

342/30/91 A	W. Reisig: Parallel Composition of Liveness
342/31/91 A	Thomas Bemmerl, Christian Kasperbauer, Martin Mairandres, Bernhard Ries: Programming Tools for Distributed Multiprocessor Computing Environments
342/32/91 A	Frank Leßke: On constructive specifications of abstract data types using temporal logic
342/1/92 A	L. Kanal, C.B. Suttner (Editors): Informal Proceedings of the Workshop on Parallel Processing for AI
342/2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS
342/2-2/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS - Revised Version (erschienen im Januar 1993)
342/3/92 A	Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/4/92 A	Claus Dendorfer, Rainer Weber: Development and Implementation of a Communication Protocol - An Exercise in FOCUS
342/5/92 A	Michael Friedrich: Sprachmittel und Werkzeuge zur Unterstüt- zung paralleler und verteilter Programmierung
342/6/92 A	Thomas F. Gritzner: The Action Graph Model as a Link between Abstract Relation Algebras and Process-Algebraic Specifications
342/7/92 A	Sergei Gorlatch: Parallel Program Development for a Recursive Numerical Algorithm: a Case Study
342/8/92 A	Henning Spruth, Georg Sigl, Frank Johannes: Parallel Algorithms for Slicing Based Final Placement
342/9/92 A	Herbert Bauer, Christian Sporrer, Thomas Krodel: On Distributed Logic Simulation Using Time Warp
342/10/92 A	H. Bungartz, M. Griebel, U. Rüde: Extrapolation, Combination and Sparse Grid Techniques for Elliptic Boundary Value Problems

Reihe A

342/11/92 A	M. Griebel, W. Huber, U. Rüde, T. Störtkuhl: The Combination Technique for Parallel Sparse-Grid-Preconditioning and -Solution of PDEs on Multiprocessor Machines and Workstation Networks
342/12/92 A	Rolf Niedermeier, Peter Rossmanith: Optimal Parallel Algorithms for Computing Recursively Defined Functions
342/13/92 A	Rainer Weber: Eine Methodik für die formale Anforderungsspezifkation verteilter Systeme
342/14/92 A	Michael Griebel: Grid– and point–oriented multilevel algorithms
342/15/92 A	M. Griebel, C. Zenger, S. Zimmer: Improved multilevel algorithms for full and sparse grid problems
342/16/92 A	J. Desel, D. Gomm, E. Kindler, B. Paech, R. Walter: Bausteine eines kompositionalen Beweiskalküls für netzmodellierte Systeme
342/17/92 A	Frank Dederichs: Transformation verteilter Systeme: Von applikativen zu prozeduralen Darstellungen
342/18/92 A	Andreas Listl, Markus Pawlowski: Parallel Cache Management of a RDBMS
342/19/92 A	Erwin Loibl, Markus Pawlowski, Christian Roth: PART: A Parallel Relational Toolbox as Basis for the Optimization and Interpretation of Parallel Queries
342/20/92 A	Jörg Desel, Wolfgang Reisig: The Synthesis Problem of Petri Nets
342/21/92 A	Robert Balder, Christoph Zenger: The d-dimensional Helmholtz equation on sparse Grids
342/22/92 A	Ilko Michler: Neuronale Netzwerk-Paradigmen zum Erlernen von Heuristiken
342/23/92 A	Wolfgang Reisig: Elements of a Temporal Logic. Coping with Concurrency
342/24/92 A	T. Störtkuhl, Chr. Zenger, S. Zimmer: An asymptotic solution for the singularity at the angular point of the lid driven cavity
342/25/92 A	Ekkart Kindler: Invariants, Compositionality and Substitution
342/26/92 A	Thomas Bonk, Ulrich Rüde: Performance Analysis and Optimization of Numerically Intensive Programs
342/1/93 A	M. Griebel, V. Thurner: The Efficient Solution of Fluid Dynamics Problems by the Combination Technique

Reihe A

342/2/93 A	Ketil Stølen, Frank Dederichs, Rainer Weber: Assumption / Commit ment Rules for Networks of Asynchronously Communicating Agents
342/3/93 A	Thomas Schnekenburger: A Definition of Efficiency of Parallel Programs in Multi-Tasking Environments
342/4/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: A Proof of Convergence for the Combination Technique for the Laplace Equation Using Tools of Symbolic Computation
342/5/93 A	Manfred Kunde, Rolf Niedermeier, Peter Rossmanith: Faster Sorting and Routing on Grids with Diagonals
342/6/93 A	Michael Griebel, Peter Oswald: Remarks on the Abstract Theory of Additive and Multiplicative Schwarz Algorithms
342/7/93 A	Christian Sporrer, Herbert Bauer: Corolla Partitioning for Distributed Logic Simulation of VLSI Circuits
342/8/93 A	Herbert Bauer, Christian Sporrer: Reducing Rollback Overhead in Time-Warp Based Distributed Simulation with Optimized Incremental State Saving
342/9/93 A	Peter Slavkovsky: The Visibility Problem for Single-Valued Surface (z = f(x,y)): The Analysis and the Parallelization of Algorithms
342/10/93 A	Ulrich Rüde: Multilevel, Extrapolation, and Sparse Grid Methods
342/11/93 A	Hans Regler, Ulrich Rüde: Layout Optimization with Algebraic Multigrid Methods
342/12/93 A	Dieter Barnard, Angelika Mader: Model Checking for the Modal Mu-Calculus using Gauß Elimination
342/13/93 A	Christoph Pflaum, Ulrich Rüde: Gauß' Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids
342/14/93 A	Christoph Pflaum: Convergence of the Combination Technique for the Finite Element Solution of Poisson's Equation
342/15/93 A	Michael Luby, Wolfgang Ertel: Optimal Parallelization of Las Vegas Algorithms

Reihe A

342/16/93 A	Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger: Pointwise Convergence of the Combination Technique for Laplace's Equation
342/17/93 A	Georg Stellner, Matthias Schumann, Stefan Lamberts, Thomas Ludwig, Arndt Bode, Martin Kiehl und Rainer Mehlhorn: Developing Multicomputer Applications on Networks of Workstations Using NXLib
342/18/93 A	Max Fuchs, Ketil Stølen: Development of a Distributed Min/Max Component
342/19/93 A	Johann K. Obermaier: Recovery and Transaction Management in Write-optimized Database Systems
342/20/93 A	Sergej Gorlatch: Deriving Efficient Parallel Programs by Systemating Coarsing Specification Parallelism
342/01/94 A	Reiner Hüttl, Michael Schneider: Parallel Adaptive Numerical Simulation
342/02/94 A	Henning Spruth, Frank Johannes: Parallel Routing of VLSI Circuits Based on Net Independency
342/03/94 A	Henning Spruth, Frank Johannes, Kurt Antreich: PHIroute: A Parallel Hierarchical Sea-of-Gates Router
342/04/94 A	Martin Kiehl, Rainer Mehlhorn, Matthias Schumann: Parallel Multiple Shooting for Optimal Control Problems Under NX/2
342/05/94 A	Christian Suttner, Christoph Goller, Peter Krauss, Klaus-Jörn Lange, Ludwig Thomas, Thomas Schnekenburger: Heuristic Optimization of Parallel Computations
342/06/94 A	Andreas Listl: Using Subpages for Cache Coherency Control in Parallel Database Systems
342/07/94 A	Manfred Broy, Ketil Stølen: Specification and Refinement of Finite Dataflow Networks - a Relational Approach
342/08/94 A	Katharina Spies: Funktionale Spezifikation eines Kommunikationsprotokolls
342/09/94 A	Peter A. Krauss: Applying a New Search Space Partitioning Method to Parallel Test Generation for Sequential Circuits

Reihe A

342/10/94 A	Manfred Broy: A Functional Rephrasing of the Assumption/Commitment Specification Style
342/11/94 A	Eckhardt Holz, Ketil Stølen: An Attempt to Embed a Restricted Version of SDL as a Target Language in Focus
342/12/94 A	Christoph Pflaum: A Multi-Level-Algorithm for the Finite-Element-Solution of General Second Order Elliptic Differential Equations on Adaptive Sparse Grids
342/13/94 A	Manfred Broy, Max Fuchs, Thomas F. Gritzner, Bernhard Schätz, Katharina Spies, Ketil Stølen: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/14/94 A	Maximilian Fuchs: Technologieabhängigkeit von Spezifikationen digitaler Hardware
342/15/94 A	M. Griebel, P. Oswald: Tensor Product Type Subspace Splittings And Multilevel Iterative Methods For Anisotropic Problems
342/16/94 A	Gheorghe Ştefănescu: Algebra of Flownomials
342/17/94 A	Ketil Stølen: A Refinement Relation Supporting the Transition from Unbounded to Bounded Communication Buffers
342/18/94 A	Michael Griebel, Tilman Neuhoeffer: A Domain-Oriented Multilevel Algorithm-Implementation and Parallelization
342/19/94 A	Michael Griebel, Walter Huber: Turbulence Simulation on Sparse Grids Using the Combination Method
342/20/94 A	Johann Schumann: Using the Theorem Prover SETHEO for verifying the development of a Communication Protocol in FOCUS - A Case Study -
342/01/95 A	Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A	Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law
342/03/95 A	Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space
342/04/95 A	Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes

Reihe A

342/05/95 A	Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems
342/06/95 A	Maximilian Fuchs: Formal Design of a Model-N Counter
342/07/95 A	Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology
342/08/95 A	Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies
342/09/95 A	Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness
342/10/95 A	Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design
342/11/95 A	Thomas Schnekenburger: The ALDY Load Distribution System
342/12/95 A	Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm
342/13/95 A	Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming
342/14/95 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks
342/15/95 A	Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs
342/16/95 A	Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik
342/17/95 A	Georg Stellner: Using CoCheck on a Network of Workstations
342/18/95 A	Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications
342/19/95 A	Thomas Schnekenburger: Integration of Load Distribution into ParMod-C
342/20/95 A	Ketil Stølen: Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication
342/21/95 A	Andreas Listl, Giannis Bozas: Performance Gains Using Subpages for Cache Coherency Control

Reihe A

342/22/95 A	Volker Heun, Ernst W. Mayr: Embedding Graphs with Bounded Treewidth into Optimal Hypercubes
342/23/95 A	Petr Jančar, Javier Esparza: Deciding Finiteness of Petri Nets up to Bisimulation
342/24/95 A	M. Jung, U. Rüde: Implicit Extrapolation Methods for Variable Coefficient Problems
342/01/96 A	Michael Griebel, Tilman Neunhoeffer, Hans Regler: Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries
342/02/96 A	Thomas Grauschopf, Michael Griebel, Hans Regler: Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic-Multigrid Coarsening for Second Order Elliptic PDEs
342/03/96 A	Volker Heun, Ernst W. Mayr: Optimal Dynamic Edge-Disjoint Embeddings of Complete Binary Trees into Hypercubes
342/04/96 A	Thomas Huckle: Efficient Computation of Sparse Approximate Inverses
342/05/96 A	Thomas Ludwig, Roland Wismüller, Vaidy Sunderam, Arndt Bode: OMIS — On-line Monitoring Interface Specification
342/06/96 A	Ekkart Kindler: A Compositional Partial Order Semantics for Petri Net Components
342/07/96 A	Richard Mayr: Some Results on Basic Parallel Processes
342/08/96 A	Ralph Radermacher, Frank Weimer: INSEL Syntax-Bericht
342/09/96 A	P.P. Spies, C. Eckert, M. Lange, D. Marek, R. Radermacher, F. Weimer, H.-M. Windisch: Sprachkonzepte zur Konstruktion verteilter Systeme
342/10/96 A	Stefan Lamberts, Thomas Ludwig, Christian Röder, Arndt Bode: PFSLib – A File System for Parallel Programming Environments
342/11/96 A	Manfred Broy, Gheorghe Ştefănescu: The Algebra of Stream Processing Functions
342/12/96 A	Javier Esparza: Reachability in Live and Safe Free-Choice Petri Nets is NP-complete

Reihe A

342/13/96 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Many-to-Many Data-flow Networks
342/14/96 A	Giannis Bozas, Michael Jaedicke, Andreas Listl, Bernhard Mitschang, Angelika Reiser, Stephan Zimmermann: On Transforming a Sequential SQL-DBMS into a Parallel One: First Results and Experiences of the MIDAS Project
342/15/96 A	Richard Mayr: A Tableau System for Model Checking Petri Nets with a Fragment of the Linear Time μ -Calculus
342/16/96 A	Ursula Hinkel, Katharina Spies: Anleitung zur Spezifikation von mobilen, dynamischen Focus-Netzen
342/17/96 A	Richard Mayr: Model Checking PA-Processes
342/18/96 A	Michaela Huhn: Put your Model Checker on Diet: Verification on Local States
342/01/97 A	Tobias Müller, Stefan Lamberts, Ursula Maier, Georg Stellner: Evaluierung der Leistungsfähigkeit eines ATM-Netzes mit parallelen Programmierbibliotheken
342/02/97 A	Hans-Joachim Bungartz, Thomas Dornseifer: Sparse Grids: Recent Developments For Elliptic Partial Differential Equations
342/03/97 A	Bernhard Mitschang: Technologie für Parallele Datenbanken — Bericht zum Workshop
342/04/97 A	Manfred Broy: Abstract Semantics of Synchronous Languages: The Example Esterel
342/05/97 A	Hans-Joachim Bungartz, Ralf Ebner, Stefan Schulte: Hierarchische Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik
342/06/97 A	Hans-Joachim Bungartz, Anton Frank, Florian Meier, Tilman Neunhoeffer, Stefan Schulte: Fluid Structure Interaction: 3 D Numerical Simulation and Visualization of a Micropump

SFB 342 : Methoden und Werkzeuge für die Nutz ung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B	Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B	Jörg Desel: On Abstraction of Nets
342/3/90 B	Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B	Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B	Barbara Paech: Concurrency as a Modality
342/2/91 B	Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier- Toolbox -Anwenderbeschreibung
342/3/91 B	Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B	Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B	Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B	Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B	W. Reisig: Concurrent Temporal Logic
342/1/92 B	Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B	Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B	Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung

Reihe B

342/2/93 B	Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B	Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS