

*Schon immer hat die Berechnung elastischer Verformungen von Körpern eine zentrale Rolle in den modernen Ingenieurwissenschaften gespielt. Dabei kann es sich sowohl um die Berechnung der Verschiebungen ganzer Tragwerke unter dem Einfluss äußerer und innerer Kräfte handeln, als auch die Berechnung der Verformung einzelner Bau- und Maschinenteile.*

# Zerreißen und Zusammenfügen

Gebietszerlegungsverfahren und Finite Elemente auf Parallelrechnern

Von Axel Klawonn und Oliver Rheinbach

## Finite Elemente

Die Berechnung der Verschiebungen unter dem Einfluss äußerer und innerer Kräfte ist im Allgemeinen nicht mehr analytisch möglich. Daher müssen Verfahren angewandt werden, die eine möglichst genaue approximative Lösung liefern. Im Bereich der Strukturmechanik ist das Standardverfahren dafür die Methode der Finiten Elemente. Dabei wird das Bauteil in viele, geometrisch einfache Objekte zerlegt, wie zum Beispiel Tetraeder, Hexaeder oder Prismen (Abb. 1). Diese Objekte nennt man Finite Elemente oder einfach nur Elemente. Die Lösung wird dann, im einfachsten Fall, in den Eckpunkten dieser Elemente berechnet. Allgemeiner kann man auch Punkte auf Kanten und Flächen der Elemente hinzunehmen. Hier beschränken wir uns jedoch der Einfachheit halber auf die Eckpunkte. Je genauer die Lösung sein soll, umso mehr Eckpunkte, also auch Elemente,

benötigt man. Die Eckpunkte werden als Knoten bezeichnet und man sagt, dass die Elemente bzw. Knoten ein Gitter bilden. Die Zerlegung eines Bauteils mit komplizierter Geometrie in Elemente ist, für sich genommen, schon eine anspruchsvolle Aufgabe, auf die wir hier nicht weiter eingehen werden (vergleiche zum Beispiel Abb. 1); das Gitter für die dort abgebildete Kurbelwelle ist mit dem Programm Netgen berechnet worden<sup>1</sup>.

Eine erste Einführung in die Ideen der Finiten Elemente findet man in Rannacher und Stein<sup>2</sup>, für eine ausführlichere Darstellung siehe Braess<sup>3</sup> oder Zienkiewicz und Taylor<sup>4</sup>.

## Lösung großer Gleichungssysteme

In der Praxis kommen Gitter mit bis zu hundert Millionen Knoten vor. Gitter dieser Größenordnung entstehen zum Beispiel, wenn man das elastische Verhalten dreidimensionaler Strukturen wie das eines

Automobils oder eines Flugzeugs möglichst genau berechnen möchte oder die komplizierte Geometrie eines Bauteils ohne Genauigkeitsverluste aufgelöst werden soll. Zur Berechnung der approximativen Verschiebungen löst man ein lineares oder nicht-lineares Gleichungssystem bei dem die Unbekannten die Verschiebungen an den Knoten sind, das heißt es gibt jeweils drei Unbekannte pro Knoten, entsprechend den drei Verschiebungen. Gleichungssysteme dieser Größenordnung lassen sich jedoch nicht mehr in angemessener Zeit auf einem einzelnen Rechner lösen. Es stellt sich die Frage, ob man diese Aufgabe nicht auf mehrere Rechner verteilen kann, die gleichzeitig einen Teil des Gleichungssystems berechnen, so dass man am Ende die Gesamtlösung aus den verschiedenen Teillösungen der einzelnen Rechner zusammensetzen kann. Lange Zeit waren ausgeklügelte Implementierungen direkter Lösungsverfahren, wie die Gauß-Eli-



Oliver Rheinbach, Axel Klawonn. Foto: Timo Bobert

**Der FETI-DP Algorithmus**

Für jedes Teilgebiet stellen wir eine lokale Steifigkeitsmatrix  $K^{(i)}$  und einen lokalen Lastvektor  $f^{(i)}$  auf. Weiterhin bezeichnen wir mit  $u^{(i)}$  die lokale Lösung der Verschiebungswerte an den Knoten. Jede lokale Steifigkeitsmatrix  $K^{(i)}$  kann so partitioniert werden, dass die Numerierung bei den Knoten beginnt, die zu den Ecken eines Teilgebiets gehören, gefolgt von den übrigen Variablen. Analog kann man bei jedem lokalen Lastvektor  $f^{(i)}$  und jeder lokalen Lösung  $u^{(i)}$  vorgehen. Versehen wir die Eckvariablen mit dem Index  $c$  und die übrigen mit  $r$ , so erhalten wir

$$K^{(i)} = \begin{bmatrix} K_{rr}^{(i)} & K_{rc}^{(i)} \\ K_{rc}^{(i)T} & K_{cc}^{(i)} \end{bmatrix}, \quad u^{(i)} = \begin{bmatrix} u_r^{(i)} \\ u_c^{(i)} \end{bmatrix}, \quad f^{(i)} = \begin{bmatrix} f_r^{(i)} \\ f_c^{(i)} \end{bmatrix}, \quad i = 1, \dots, N.$$

Im nächsten Schritt werden die lokalen Steifigkeitsmatrizen und Lastvektoren in den Eckknoten assembliert und wir erhalten aus den lokalen Blöcken  $K_{cc}^{(i)}$  die globale Matrix  $\tilde{K}_{cc}$  und aus den lokalen Blöcken  $K_{rc}^{(i)}$  die teillasemblierten Matrizen  $\tilde{K}_{rc}^{(i)}$ . Wir führen folgende Notation ein  $K_{rr} := \text{diag}_{i=1}^N(K_{rr}^{(i)})$  und  $\tilde{K}_{rc} := [\tilde{K}_{rc}^{(1)T} \dots \tilde{K}_{rc}^{(N)T}]^T$ . Die globalen Vektoren  $\tilde{u}_c$  und  $\tilde{f}_c$  werden analog definiert.

Um zu garantieren, dass die Lösung entlang der Teilgebietsränder keinen Sprung aufweist, müssen wir noch eine Sprungbedingung einführen. Da durch die Teillassemblierung bereits Stetigkeit in den Ecken der Teilgebiete garantiert ist, müssen wir diese Bedingung nur noch auf den übrigen Knoten fordern. Dazu führen wir eine Matrix  $B_r$  ein, die so konstruiert ist, dass  $B_r u_r = 0$  die Stetigkeit der Lösung an den Teilgebietsrändern bedeutet.

Die Lösung des Finite Elemente Problems ist dann gegeben durch

$$\begin{bmatrix} K_{rr} & \tilde{K}_{rc} & B_r^T \\ \tilde{K}_{rc}^T & \tilde{K}_{cc} & 0 \\ B_r & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r \\ \tilde{u}_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r \\ f_c \\ 0 \end{bmatrix}. \tag{1}$$

Eliminierung der Verschiebungsvariablen  $u_r$  und  $\tilde{u}_c$  führt auf ein reduziertes lineares Gleichungssystem der Form

$$F_A \lambda = d_A,$$

bei dem als Unbekannte nur noch die Lagrangeschen Multiplikatoren  $\lambda$  auftauchen. Die Matrix  $F_A$  und die rechte Seite  $d_A$  erhält man formal durch Block-Gauß-Elimination, allerdings wird  $F_A$  nie explizit berechnet, dies würde auf ein dicht-besetztes Gleichungssystem führen, sondern in jeder Iteration werden entsprechende dünnbesetzte Gleichungssysteme gelöst; vgl. Farhat, Lesoinne, Pierson (2000) oder Klawonn, Widlund, Dryja (2002a) für weitere Details.

Um bessere Konvergenzeigenschaften zu erhalten, müssen wir zusätzliche, optionale Nebenbedingungen einführen. Diese sind von der Form

$$Q_r u_r = 0. \tag{2}$$

Hierbei ist  $Q_r$  eine Matrix, die garantiert, dass die Mittelwerte der Verschiebungen auf ausgewählten Flächen und Kanten keine Sprünge haben. Flächen und Kanten mit dieser Eigenschaft nennen wir primal. Die Anzahl der Zeilen der Matrix  $Q_r$  ist gegeben durch die Anzahl primärer Kanten und Flächen.

Um die zusätzliche Nebenbedingung (2) zu erfüllen, führen wir optionale Lagrangesche Multiplikatoren  $\mu$  ein. Aus (1) erhalten wir dann das lineare Gleichungssystem

$$\begin{bmatrix} K_{rr} & \tilde{K}_{rc} & Q_r^T & B_r^T \\ \tilde{K}_{rc}^T & \tilde{K}_{cc} & 0 & 0 \\ Q_r & 0 & 0 & 0 \\ B_r & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r \\ \tilde{u}_c \\ \mu \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r \\ f_c \\ 0 \\ 0 \end{bmatrix}. \tag{3}$$

Eliminierung der Unbekannten  $u_r$ ,  $\tilde{u}_c$  und  $\mu$  führt wieder auf ein reduziertes Gleichungssystem der Form

$$F \lambda = d, \tag{4}$$

wobei die Matrix  $F$  und die rechte Seite  $d$  formal wieder durch Block-Gauß Elimination entstehen. Wir definieren nun den (Dirichlet-) Vorkonditionierer. Dazu benötigen wir einen skalierten Springoperator  $B_{D,r}$ . Diesen erhält man aus  $B_r = [O \ B_\Delta]$  durch gebietsweise Skalierung von  $B_\Delta$  mit passenden, diagonalen Skalierungsmatrizen  $D^{(i)}$ , indem wir definieren  $B_{D,\Delta} := [D^{(1)} B_\Delta^{(1)} \dots D^{(N)} B_\Delta^{(N)}]$ . Die Skalierungsmatrizen  $D^{(i)}$  werden durch Materialkoeffizienten bestimmt, vgl. Klawonn, Widlund, Dryja (2002a) für weitere Details. Schließlich fügen wir für jede Ecke an passender Stelle eine Nullspalte zu  $B_{D,r}$  hinzu. Durch Gauß-Elimination der Verschiebungen, die zu Knoten im inneren eines Teilgebiets gehören, d.h. die nicht auf dem Interface liegen, erhalten wir aus den lokalen Steifigkeitsmatrizen  $K^{(i)}$  lokale Schurkomplemente  $S^{(i)}$ . Fassen wir diese in einer Block-Diagonalmatrix  $S := \text{diag}_{i=1}^N(S^{(i)})$  zusammen, so ist der Dirichlet-Vorkonditionierer definiert durch  $M^{-1} := B_{D,r} S B_{D,r}^T$ . Der zugehörige FETI-DP Algorithmus ist das vorkonditionierte Verfahren der konjugierten Gradienten zur Lösung des vorkonditionierten Systems

$$M^{-1} F \lambda = M^{-1} d.$$

mination oder die Cholesky-Zerlegung, die meist benutzen Verfahren in der Strukturmechanik. Diese Algorithmen sind sehr robust, haben jedoch keine optimale Komplexität und lassen sich auch nicht einfach parallelisieren, also so programmieren, dass sie auf mehreren Rechnern gleichzeitig benutzt werden können. Zusätzlich haben diese Verfahren auch einen erhöhten Speicherbedarf, wenn man von zwei- zu dreidimensionalen Problemen übergeht.

Eine Alternative zu direkten Lösern sind Iterationsverfahren, die häufig schon direkt für den Einsatz auf Parallelrechnern entworfen werden. Zudem sind oft auch die

Speicheranforderungen wesentlich niedriger als die direkter Verfahren, insbesondere bei dreidimensionalen Problemen. Im Gegensatz zu direkten Verfahren, bei denen die Lösung in einem Schritt berechnet wird, nähern sich Iterationsverfahren nur schrittweise der Lösung an. Dabei wird in jedem Iterationsschritt nur die Multiplikation der Matrix des Gleichungssystems mit einem Vektor benötigt. Häufig lässt sich die Anzahl der Iterationen durch einen Vorkonditionierer verringern. Ein Vorkonditionierer ist dabei eine Approximation an die Inverse der Matrix des Gleichungssystems. Diese Approximation muss nicht als

Matrix aufgestellt werden, sondern nur die Anwendung auf einen Vektor muss leicht zu berechnen sein.

**Gebietszerlegungsverfahren**

Gebietszerlegungsverfahren sind eine Klasse von Iterationsverfahren, die sich als numerisch skalierbar und häufig auch als sehr robuste Algorithmen für Probleme der Strukturmechanik erwiesen haben. Das wohl erste Gebietszerlegungsverfahren hat H.A. Schwarz im Jahr 1869 vorgeschlagen und es für Existenzbeweise eingesetzt. In den achtziger Jahren des darauf folgenden Jahrhunderts fand die Idee der Gebietszerlegung erhöhte Interesse, als sowohl das Verfahren von Schwarz, als auch die Ideen der rekursiven Substrukturierung aus der Ingenieurspraxis von Mathematikern und Ingenieuren wieder aufgegriffen und weiterentwickelt wurden.

Gebietszerlegungsmethoden sind in diesem Zusammenhang vorkonditionierte Iterationsverfahren. In einem solchen Verfahren zerlegen wir das Gebiet, auf dem die partielle Differentialgleichung gelöst werden soll, in eine Anzahl überlappender oder nicht überlappender Teilgebiete. In der Praxis kann das Gebiet zum Beispiel eine Kurbelwelle sein und die zugehörige partielle Differentialgleichung wird durch das entsprechende Elastizitätsproblem beschrieben (vgl. Abb. 2). Die automatische Zerlegung eines Finite-Elemente-Gitters führt auf das Problem einen Graphen zu partitionieren; eine Aufgabe, die in vielen wissenschaftlichen Anwendungen auftritt. Hierfür sind in der Informatik entsprechende Algorithmen, so genannte Graphpartitionierungsalgorithmen, und Software entwickelt worden<sup>5,6</sup>.

In jedem Iterationsschritt wird für jedes Teilgebiet ein lokales Problem gelöst. Dieses Teilproblem ist normalerweise eine Approximation an das Originalproblem eingeschränkt auf das jeweilige Teilgebiet. Abhängig von dem jeweiligen Algorithmus werden diese Teilprobleme

entweder approximativ oder mit einem direkten Verfahren gelöst. Gebietszerlegungsverfahren erlauben somit eine gewisse Flexibilität in der Konstruktion robuster, aber immer noch skalierbarer Iterationsverfahren. Man kann sie auch als hybride Verfahren ansehen, die eine Brücke zwischen direkten Lösern und reinen iterativen Verfahren herstellen.

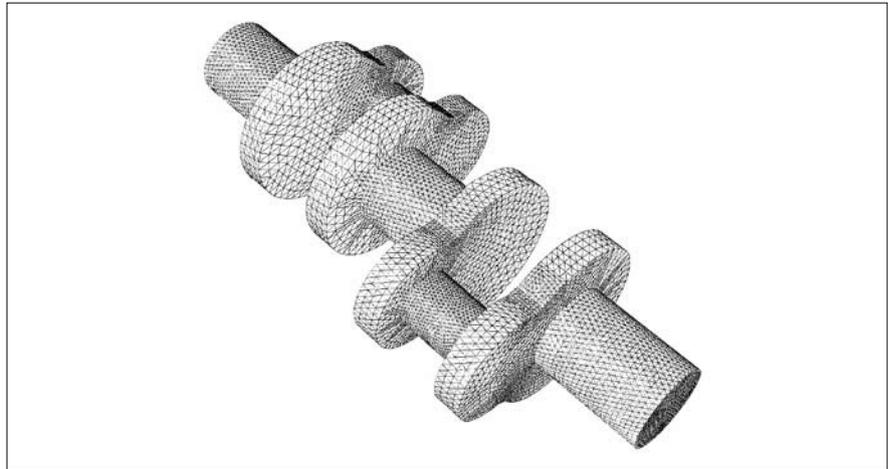
Um numerisch skalierbar zu sein, das heißt, das Verfahren konvergiert unabhängig von der Anzahl der Teilgebiete, muss in einem Gebietszerlegungsverfahren immer noch zusätzlich ein kleines globales, gebietsübergreifendes Problem gelöst werden. Ein solches Problem garantiert den globalen Informationsaustausch in jedem Iterationsschritt, es erfordert jedoch auch eine spezielle Aufmerksamkeit bei der Implementierung auf einem Parallelrechner im Hinblick auf die parallele Skalierbarkeit.

Zusammenfassend könnte man sagen, dass die Idee der Gebietszerlegungsverfahren darin besteht, in jedem Iterationsschritt eine approximative Lösung des ursprünglichen Gleichungssystems zu berechnen, die sich aus vielen, lokalen Teilgebetslösungen und der Lösung eines kleinen, globalen Problems zusammensetzt.

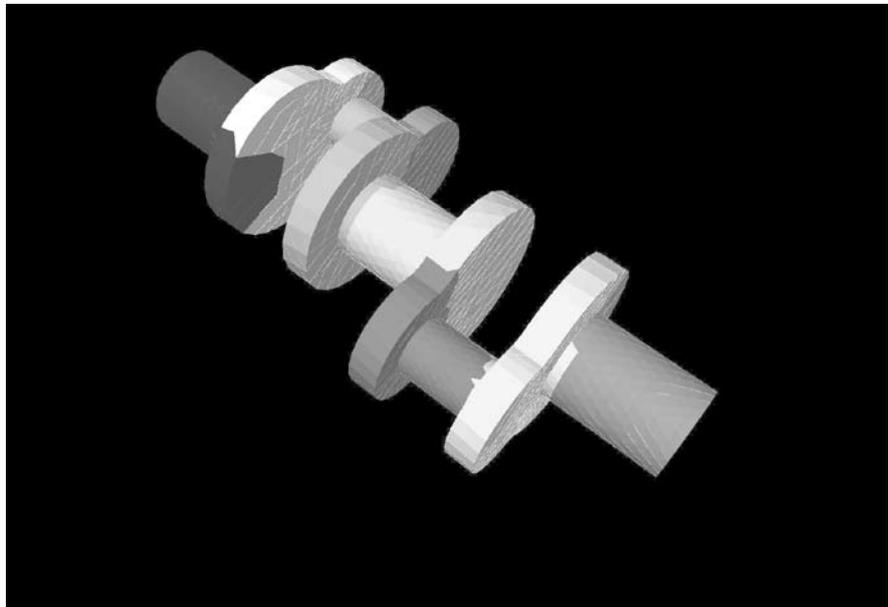
Abhängig von der Art der Zerlegung in Teilgebiete werden die Algorithmen häufig in überlappende und nicht überlappende Gebietszerlegungsverfahren eingeteilt. Die letzteren werden oft auch als iterative Substrukturierungsverfahren bezeichnet. Im Folgenden werden wir nur Verfahren der zweiten Klasse betrachten. Für eine ausführliche Beschreibung von Gebietszerlegungsverfahren vergleiche Smith, Bjørstad und Gropp<sup>7</sup>, Quarteroni und Valli<sup>8</sup> oder Toselli und Widlund<sup>9</sup>.

**FETI**

Wir betrachten hier eine spezielle Familie von nicht überlappenden

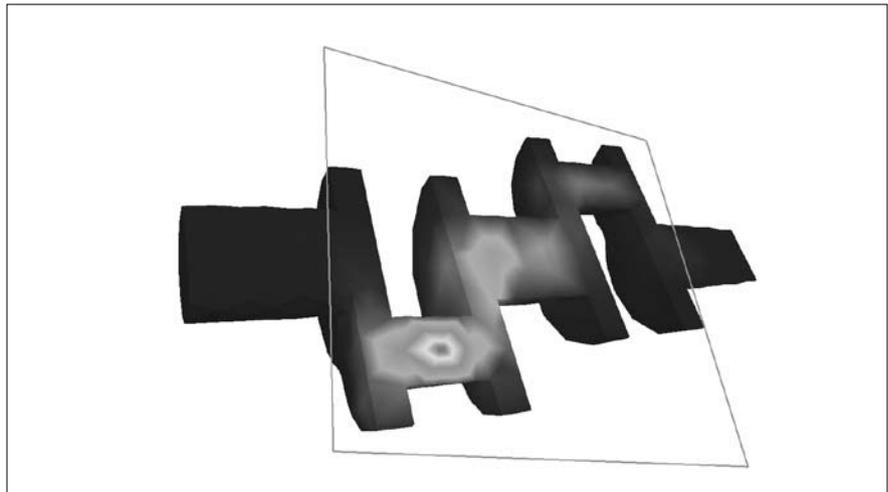


(1) Zerlegung einer Kurbelwelle in 150 000 Tetraeder.

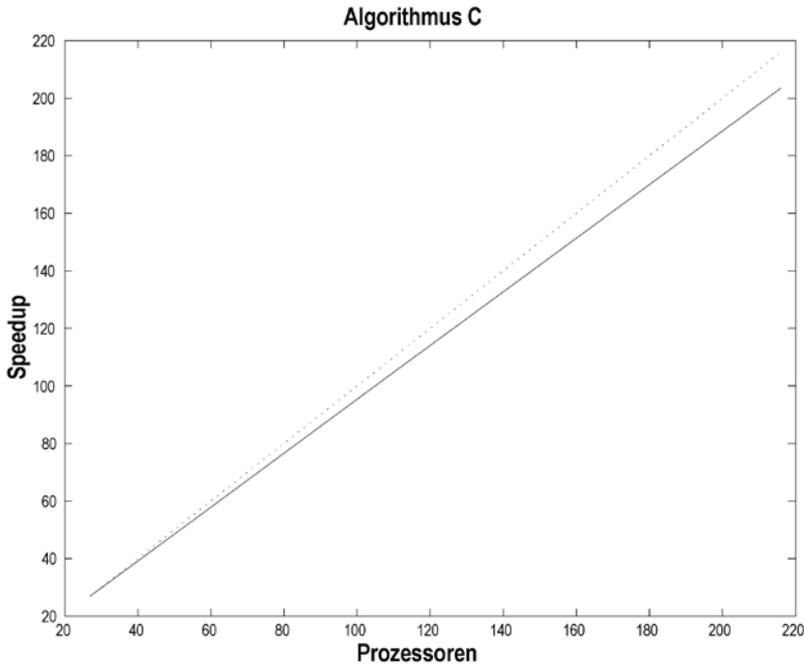


(2) Gebietszerlegung einer Kurbelwelle in acht Teilgebiete (jeder Grauton repräsentiert ein Teilgebiet).

Quellen: Gitter: Netgen, Zerlegung: Chaco, Visualisierung: Medit



(3) Spannungsberechnungen am Modell einer Kurbelwelle.



(4) Erreichter Speedup einer Variante eines FETI-DP-Algorithmus für ein Diffusionsproblem mit Sprüngen; berechnet auf 27 bis 216 Prozessoren bei 2 685 616 Unbekannten und 216 Teilgebieten.

Quellen: für Details vgl. A. Klawonn, O. Rheinbach, O. Rheinbach und B. Widlund 2004. Implementiert mit PETSc, siehe S. Balay u.a. 2001

Die Autoren bedanken sich für die Benutzung des Parallelrechners „Jazz“, einem Cluster mit 350 Prozessoren der Mathematics and Computer Science Division des Argonne National Laboratory der USA.

Gebietszerlegungsverfahren, die so genannten FETI (Finite Element Tearing and Interconnecting)-Verfahren, die von Farhat und Roux zu Beginn der neunziger Jahre des letzten Jahrhunderts eingeführt wurden<sup>10</sup>. Bei der Familie der FETI-Gebietszerlegungsverfahren geht man wie folgt vor. Zuerst zerlegt man das Gebiet in nichtüberlappende Teilgebiete und stellt für jedes Teilgebiet eine lokale Steifigkeitsmatrix und einen lokalen Lastvektor auf. Angenommen, jedes dieser Teilprobleme ist lösbar, dann erhalten wir auf jedem der Teilgebiete eine lokale Lösung, die dort das Elastizitätsproblem löst. Allerdings werden diese lokalen Lösungen im Allgemeinen von Teilgebiet zu Teilgebiet einen Sprung aufweisen (Abb. 5).

Diese Tatsache ist nicht besonders überraschend, da wir ja an den Teilgebietsrändern keine gemeinsamen Bedingungen für die Lösung vorgeschrieben hatten. Daher fordern wir zusätzlich, dass die Lösung an den Teilgebietsrändern keinen Sprung mehr hat und stellen die zugehörigen Gleichungen auf. Wenn der Sprung der Verschiebungen Null ist, haben wir eine Lösung für unser Ursprungsproblem gefunden.

Insgesamt müssen wir also ein Energieminimierungsproblem mit Nebenbedingungen lösen. Indem wir zum Erzwingen der Sprungnebenbedingungen Lagrangesche Multiplikatoren einführen, erhalten wir daraus ein gemischtes Gleichungssystem mit den lokalen Verschiebungen pro Teilgebiet und den Lagrangeschen Multiplikatoren als Variablen. Um dieses Gleichungssystem zu lösen, eliminieren wir zunächst die Verschiebungsvariablen. Diese können unabhängig voneinander durch Gauß-Elimination entfernt werden und wir erhalten ein Gleichungssystem, das nur noch die Lagrangeschen Multiplikatoren als Unbekannte enthält. Solch ein reduziertes System löst man nun mit einem iterativen Verfahren, dem Verfahren der konjugierten Gradienten und einem geeigneten Vorkonditionierer.

#### Die Finite Elemente Methode (FEM)

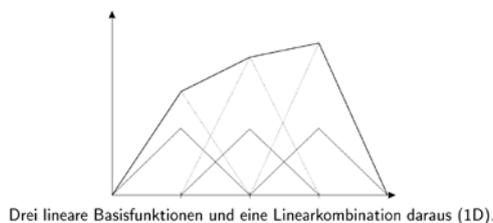
Finite Element Methoden wurden entwickelt, weil sich mathematisch exakte Lösungen von partiellen Differentialgleichungen (zum Beispiel aus der Mechanik) nur für sehr einfache Fälle angeben lassen. Um dennoch realistische Probleme aus den Ingenieurwissenschaften und der Physik lösen zu können, wird daher die Suche nach der exakten Lösung aufgegeben und nur nach einer Approximation der Lösung gesucht. Die Finite Elemente Methoden gehören zu den Techniken, die es auf diese Weise ermöglichen, auch in komplizierten Geometrien Lösungen zu finden, die die exakte Lösung beliebig genau annähern. Ein Objekt, dessen Verformung berechnet werden soll (zum Beispiel die Karosserie eines Autos bei einem Crash), wird dabei so behandelt, als bestünde es aus vielen kleinen, einfachen Teilen (Elementen), deren Verformungsverhalten man kennt. Mit einem groben Gitter erhält man eine erste Annäherung an das Verhalten des Objekts. Durch Unterteilung in immer mehr und immer kleinere Elemente kann man das gesuchte Verhalten soweit annähern, wie es für die jeweilige Aufgabenstellung erforderlich ist. Exakt beschrieben bedeutet dies, dass das Objekt (Gebiet) in einem ersten Schritt in einfache geometrische Formen wie Tetraeder, Würfel oder Prismen (Elemente) zerlegt wird. Im nächsten Schritt wird (im einfachsten Fall) jedem Eckpunkt (Knoten) eine Basisfunktion zugeordnet. Die Basisfunktion wird so gewählt, dass sie an dem zugeordneten Knoten den Wert Eins annimmt und an den anderen Knoten den Wert Null. Die ursprüngliche Differentialgleichung in so genannter schwacher Form

$$\text{Finde } u, \text{ so dass } a(u, v) = (f, v) \quad \forall v \in V \quad (5)$$

ist in einem abstrakten, unendlich-dimensionalen Funktionenraum  $V$  formuliert und daher im Allgemeinen schwer zu lösen. Die Finite Elemente Methode besteht darin, dass der unendlich-dimensionale Funktionenraum  $V$  durch einen endlich-dimensionalen Raum  $V^h$  ersetzt wird, zum Beispiel durch den Raum der stückweise linearen, stetigen Funktionen. Die Differentialgleichung erhält dadurch die Form

$$\text{Finde } u_h, \text{ so dass } a(u_h, v_h) = (f, v_h) \quad \forall v_h \in V^h. \quad (6)$$

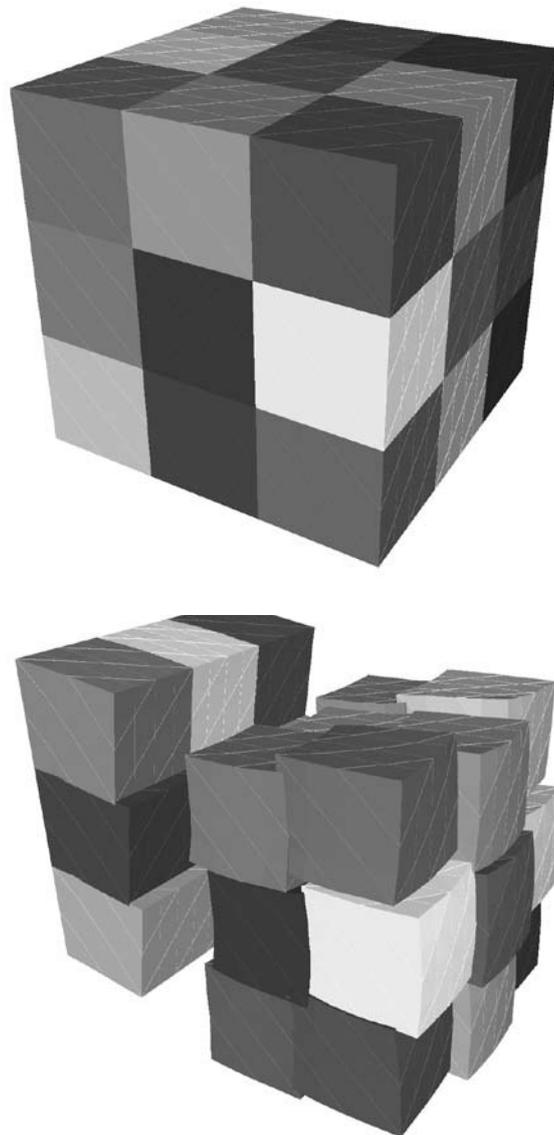
Die Darstellung von  $u_h$  durch die Basis führt dann auf ein (großes) lineares Gleichungssystem. Löst man dieses Gleichungssystem, so hat man die gesuchte Näherungslösung berechnet.



Bisher haben wir angenommen, dass die lokalen Gleichungssysteme alle lösbar sind. Bei der Eliminierung der Verschiebungen aus dem partitionierten Gleichungssystem muss jedoch beachtet werden, dass im Allgemeinen die meisten der lokalen Steifigkeitsmatrizen nur positiv semidefinit sind. Dieser Fall tritt auf, wenn nur natürliche Randbedingungen gegeben sind, wie zum Beispiel bei inneren Teilgebieten. Dann können bis zu sechs Starrkörperbewegungen als Elemente des Kerns auftreten. Das zugehörige lokale Gleichungssystem kann trotzdem gelöst werden, wenn es konsistent ist, das heißt, die rechte Seite im Bild der lokalen Steifigkeitsmatrix ist. Dies wird bei den ursprünglichen (einstufigen) FETI-Verfahren durch einen Projektor erreicht. Dieser Projektor definiert gleichzeitig ein globales Problem, welches die Skalierbarkeit des Verfahrens garantiert. Die Eliminierung der Verschiebungsvariablen entspricht dann der Lösung eines Neumann-Problems in jedem Teilgebiet. Dieser Ansatz geht ursprünglich auf Farhat und Roux<sup>11</sup> zurück. Er wurde in den letzten zehn Jahren durch intensive Forschung weiterentwickelt, für eine theoretische Analyse der Konvergenzeigenschaften vergleiche unter anderem Klawonn und Widlund<sup>12</sup>.

### FETI-DP

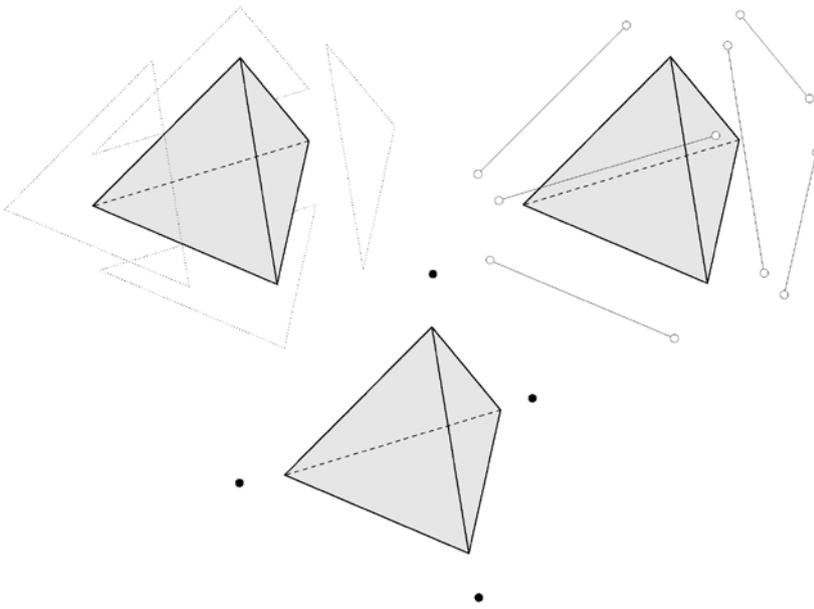
Ein anderer Ansatz, die reinen Neumann-Probleme zu lösen, führt zu der Familie der dual-primale FETI (FETI-DP)-Verfahren. Bei diesen Algorithmen teilt man die Knoten auf den Rändern der Teilgebiete in Eckknoten (Ecken) und übrige Knoten ein (Abb. 6). Um die Lösbarkeit der lokalen Gleichungssysteme zu garantieren, wird nun eine Teilassemblierung der Steifigkeitsmatrizen in den Ecken vorgenommen. Dieses Vorgehen kann man sich in etwa so vorstellen, als ob man die Lösungen auf den Teilgebieten, die im Allgemeinen Sprünge aufweisen, (Abb. 5), an den



(5) Oben: Würfel zerlegt in 27 Teilgebiete. Unten: Die 27 Lösungen des Elastizitätsproblems auf den Teilgebieten. Die Sprünge der Lösung konvergieren mit fortschreitender Iteration gegen Null, das heißt die Abstände zwischen den einzelnen Würfeln werden immer kleiner, bis sie zum Schluss zusammen den verformten Körper bilden.

Ecken „zusammenklebt“. Die zugehörigen Verschiebungen sind dann automatisch in jeder Iteration stetig. Die Stetigkeit der übrigen Variablen wird, wie bei einstufigen FETI-Verfahren, weiterhin durch Lagrange-Multiplikatoren erzwungen. Die weitere Vorgehensweise ist ähnlich wie bei den einstufigen Verfahren und führt auf einen Algorithmus, durch den die Lagrange-Multiplikatoren iterativ berechnet werden. Der Ansatz, die lokalen Matrizen in den Ecken zu assemblieren, führt zu einem skalierbaren Verfahren in zwei Dimensionen, allerdings im

Allgemeinen zu einer schlechteren Konvergenzrate in drei Dimensionen. Dazu muss ein besseres globales Problem eingeführt werden. Um es zu beschreiben, führen wir zunächst eine Unterteilung der Knoten auf den Teilgebietsrändern, dem sogenannten Interface, in Eckknoten (Ecken), Kantenknoten (Kanten) und Seitenflächenknoten (Flächen) ein (Abb. 6). Die Idee besteht darin, zusätzlich zur Assemblierung in den Ecken als weitere Nebenbedingung zu verlangen, dass bestimmte Mittelwerte der Verschiebungen auf ausgewählten Flächen oder Kanten



(6) (Seiten-)Flächen (oben links), Kanten (oben rechts) und Ecken (unten) bei einem tetraederförmigen Teilgebiet.

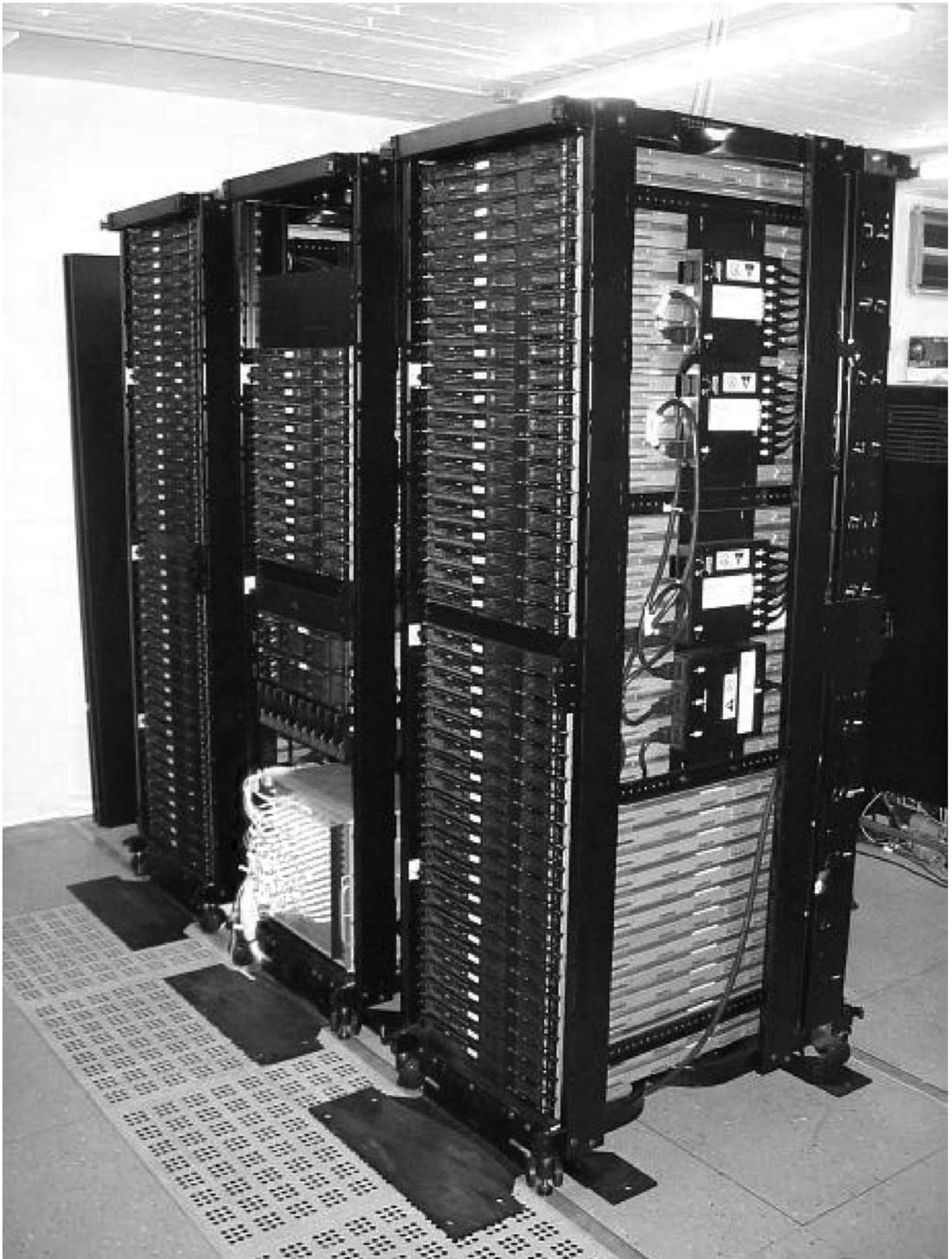
denselben Wert haben. Wenn also zwei Teilgebiete eine gemeinsame Kante oder Fläche haben, dann sollen die Mittelwerte der Verschiebungen, jeweils pro Teilgebiet gebildet, denselben Wert haben. Dies muss nicht für alle Flächen und Kanten vorgenommen werden<sup>13</sup>. Die Gesamtlösung verändert sich durch diese zusätzlichen Nebenbedingungen nicht. Denn bei einer stetigen Lösung sind auch die entsprechenden Mittelwerte auf einer Fläche oder Kante automatisch stetig. Die Sprünge sind bei Konvergenz des iterativen Verfahrens jedoch Null, die Lösung somit stetig. Die zusätzlichen Nebenbedingungen sind in diesem Sinne also optional. Eine Möglichkeit diese optionalen Nebenbedingungen zu implementieren ist die Einführung zusätzlicher (optionaler) Lagrangescher Multiplikatoren. Bei dieser Variante der FETI-DP-Verfahren werden zusätzlich zu den Verschiebungen auch die optionalen Lagrangeschen Multiplikatoren eliminiert und man erhält formal wieder ein lineares Gleichungssystem mit den ursprünglichen Lagrangeschen Multiplikatoren als Unbekannten. Dieses wird wie zuvor mit dem Verfahren der konjugierten Gradienten und einem

passenden Vorkonditionierer gelöst. Wir erhalten verschiedene FETI-DP Algorithmen, je nachdem, ob nur in den Ecken assembliert wird (Algorithmus A) oder ob zusätzlich noch die Mittelwerte auf den Kanten und Flächen (Algorithmus B), nur auf den Kanten (Algorithmus C) oder nur auf den Flächen (Algorithmus E) übereinstimmen. Dabei nehmen wir hier an, dass immer alle Ecken, Kanten oder Flächen ausgewählt werden. Man kann sich jedoch auch auf eine minimale Auswahl beschränken<sup>14</sup>. Die entsprechenden Gleichungssysteme werden jeweils durch den Index A, B, C oder E gekennzeichnet. Es ist klar, dass Algorithmus B, C und E mehr Aufwand pro Iterationsschritt benötigen, als Algorithmus A. Allerdings führt gerade dieser Mehraufwand oft zu besser skalierbaren Algorithmen.

Der (Dirichlet-)Vorkonditionierer für die FETI-Verfahren wird aus zwei Komponenten gebildet. Zum einen aus einem skalierten Sprungoperator und zum anderen aus einem Schurkomplement. Die Skalierung des Sprungoperators wird aus bestimmten Materialkoeffizienten des betrachteten Körpers bestimmt und ist notwendig, um ein Iterationsverfahren zu erhalten, das unab-

hängig von eventuellen Sprüngen in den Materialien konvergiert. Das Schurkomplement ist eine Block-Diagonalmatrix, die sich aus einzelnen Schurkomplementen zusammensetzt, die wiederum gebietsweise wie folgt gebildet werden. Für jedes Teilgebiet eliminiert man die inneren Variablen aus der lokalen Steifigkeitsmatrix, das heißt die Variablen, die nicht auf dem Rand des entsprechenden Teilgebiets liegen. Dies entspricht der Lösung eines Problems mit Dirichlet-Randdaten, daher auch der Name Dirichlet-Vorkonditionierer. Diese Lösung der Dirichlet-Probleme muss dann in jedem Iterationsschritt für jedes Teilgebiet ausgeführt werden. Da die Probleme jedoch alle entkoppelt sind, kann dies komplett parallel geschehen.

Bei den Verfahren der FETI-Familie wird in jedem Iterationsschritt eine approximative Lösung berechnet, deren Sprung mit jedem Schritt immer kleiner wird und die gegen die gesuchte Lösung konvergiert. Die Geschwindigkeit, mit der die Iterierten gegen die gesuchte Lösung konvergieren, hängt bei den FETI-Verfahren mit Dirichlet-Vorkonditionierer nur noch logarithmisch von der Anzahl der Unbekannten pro Teilgebiet ab, aber nicht mehr von der Anzahl der Teilgebiete und der Anzahl Unbekannter des Gesamtproblems. Die Algorithmen der FETI-Familie sind also numerisch skalierbar. Erste Abschätzungen dieser Art sind für die einstufigen FETI-Verfahren und für Probleme ohne große Sprünge in den Materialkoeffizienten von Mandel und Tezaur<sup>15</sup> hergeleitet worden. Die Abschätzungen wurden von Klawonn und Widlund<sup>16</sup> verschärft und erweitert auf Probleme mit beliebig großen Sprüngen in den Materialkoeffizienten. Für FETI-DP-Verfahren in zwei Raumdimensionen und für Probleme ohne Materialsprünge haben Mandel und Tezaur<sup>17</sup> ein solches Resultat bewiesen. Für den dreidimensionalen Fall und Probleme mit beliebig großen Sprüngen in den Materi-



(7) Zivcluster der Westfälischen Wilhelms-Universität Münster mit 96 Prozessoren.  
Quelle: <http://zivcluster.uni-muenster.de>

alkoeffizienten ist die Familie von Algorithmen von Klawonn, Widlund und Dryja<sup>18,19</sup> erweitert worden. In diesen Arbeiten ist jeweils auch eine Abschätzung der Konvergenzgeschwindigkeit vom oben angegebenen Typ hergeleitet worden. Alle genannten Arbeiten behandeln skalare Diffusionsprobleme, wie sie zum Beispiel bei der stationären Wärmeleitung vorkommen. Die Erweiterung auf die Verschiebungsgleichungen der linearen Elastizität ist für die einstufigen FETI-Verfahren recht einfach, wogegen bei den FETI-DP-Verfahren einige nicht-triviale Änderungen vorgenommen werden müssen. Die Entwicklung und Konvergenzanalyse entsprechender Verfahren ist inzwischen sehr weit fortgeschritten<sup>20</sup>.

Die FETI-DP-Verfahren gehen zurück auf Farhat u. a.<sup>21</sup> für Probleme in zwei Raumdimensionen und die erste theoretische Konvergenzabschätzung findet sich in Mandel und Tezaur<sup>22</sup>. Für dreidimensionale Probleme müssen erhebliche Veränderungen vorge-

nommen werden, um ein skalierbares Verfahren zu erhalten. Erste numerische Ergebnisse finden sich in Farhat, Lesoinne und Pierson<sup>23</sup>. In Klawonn, Widlund und Dryja<sup>24,25</sup> ist systematisch eine ganze Familie von Algorithmen für dreidimensionale Probleme entwickelt worden<sup>26</sup>.

### Summary

In modern engineering and natural sciences, partial differential equations play an important role in modeling complex technical problems and natural phenomena. Computer simulations of complicated models are becoming more and more important as the complexity of the problems considered increases. In order to obtain a sufficiently accurate simulation, often a large number of finite elements has to be used to discretize the partial differential equations. For stationary problems or time-implicit methods, the discretization of the partial differential equations on such a mesh

leads, either directly or after a linearization, to a large linear algebraic system of equations which is often very sparse. The development of parallel computing allows to solve very large problems, with up to hundreds of millions of degrees of freedom in structural and fluid mechanics. These problems are often so large that we need very refined numerical techniques to solve them in an acceptable time even on very fast parallel computers. For a long time, elaborate implementations of direct solvers such as Gauss elimination have been the method of choice. These solvers are very robust, but they have a non optimal complexity and are not so easily parallelized. An alternative to direct solvers are preconditioned iterative methods. One particular class of iterative methods are domain decomposition methods which have been proven to be numerically scalable while often still very robust. In such methods, the domain, where the partial differential equation has to be solved, is partitioned into a number of overlapping or nonoverlapping subdomains. In each iteration step, a local problem is solved on each subdomain; often these local problems are suitable restrictions of the original problem to that subdomain. The local problems are then solved concurrently on different processors of a parallel computer. Attention has to be given to obtain scalable algorithms. The present article is especially focused on domain decomposition algorithms of the FETI family, a class of algorithms which is central to the research of the authors.

### Anmerkungen

- 1) Schöberl 2003: Netgen
- 2) Rannacher und Stein 1999
- 3) Braess 2003
- 4) Zienkiewicz und Taylor 1989
- 5) Hendrickson und Leland 1995: Chaco
- 6) Karypis, Schloegel und Kumar 2003: ParMetis
- 7) Smith, Bjørstad und Gropp 1996
- 8) Quarteroni und Valli 1999
- 9) Toselli und Widlund 2004
- 10) Farhat und Roux 1991
- 11) ebenda

#### Parallelrechner

Obwohl heutige Rechner um Größenordnungen leistungsfähiger geworden sind als ihre Urahnen, reicht ihre Rechenleistung immer noch nicht aus, um den Anforderungen von Wissenschaft und Technik gerecht zu werden. Dies betrifft typische Anwendungen z.B. aus der Meteorologie (Wetter und Klima), Mechanik oder der Simulation von Strömungen.

Durch parallele Verarbeitung mit mehreren Prozessoren lassen sich Rechnungen beschleunigen, die einen einzelnen Rechner überfordern würden. Wenn die Prozessoren sich nicht mehr einen gemeinsamen Hauptspeicher teilen, sondern direkten Zugriff nur auf ihren eigenen Speicher besitzen, spricht man von verteilten Rechnern und verteiltern Rechnern. Ein solcher Rechner besteht aus einzelnen mehr oder weniger vollständigen Computern (Knoten) mit eigenem Hauptspeicher, die durch Kommunikationshardware miteinander verbunden sind. Heutige Supercomputer sind massiv parallele, verteilte Rechner mit mehreren Tausend Prozessoren. Das gängigste Beispiel für einen verteilten Rechner ist ein Cluster von PCs, die durch ein schnelles Netzwerk miteinander verbunden sind. Kleine Cluster mit zehn oder zwanzig Knoten finden sich heute in vielen Universitäten und Firmen. Auf der anderen Seite des Leistungsspektrums stehen Cluster mit mehreren tausend Prozessoren, die zu den schnellsten Rechnern der Welt gehören. Es ist wichtig zu betonen, dass die Motivation für den Einsatz von Parallelrechnern dieser Art nicht nur in der höheren Geschwindigkeit zu suchen ist, mit der Ergebnisse zur Verfügung stehen. Verteiltes Rechnen ermöglicht auch, Probleme zu betrachten, deren Lösung wesentlich mehr Speicher erfordert, als einem einzelnen Prozessor zur Verfügung gestellt werden kann.

Die meisten Standard-Algorithmen sind vor dem Hintergrund des klassischen, sequentiellen Rechnermodells entworfen worden und sind daher für Parallelrechner weniger geeignet. Parallele Algorithmen müssen die Aufgaben intelligent aufteilen, damit die Rechenleistung aller Prozessoren möglichst gut ausgenutzt wird. Dies sollte auch möglich sein, wenn die Anzahl der Prozessoren in die Tausende und in naher Zukunft Zehntausende steigt. Da die Kommunikation zwischen Knoten um Größenordnungen langsamer ist, als der Zugriff auf den lokalen Hauptspeicher, müssen reale Algorithmen möglichst viel Arbeit lokal, d.h. unabhängig von anderen Knoten, erledigen können. Es hat sich erwiesen, dass Gebietszerlegungsverfahren geradezu maßgeschneidert für die Architektur der verteilten Parallelrechner sind, von kleinen Clustern angefangen, bis zu Millionen Euro teuren Supercomputern (allerdings ist ihr Einsatz nicht auf verteilte Rechner beschränkt geblieben). Gebietszerlegungsverfahren erfordern viele lokale Rechnungen, die unabhängig von anderen Gebieten geschehen können. Die Kommunikation, die über das Netzwerk zwischen den Knoten geschehen muss, ist dagegen beschränkt auf die Oberfläche von Gebieten und fast vollständig beschränkt auf die nächsten Nachbarn. Nur ein kleiner Teil der Information muss zwischen allen Prozessoren ausgetauscht werden.

Die Rechenleistung von Parallelrechnern wird heute in Teraflops (Billionen Fließkomma-Operationen pro Sekunde) angegeben. Zweimal im Jahr stellen die Universität Mannheim, die Universität Tennessee und das National Energy Scientific Computing Center der USA die Liste der 500 schnellsten Rechner der Welt zusammen und veröffentlichen Sie im Internet unter <http://www.top500.org>.

- 12) Klawonn und Widlund 2001
- 13) Klawonn, Widlund und Dryja 200a, b
- 14) ebenda
- 15) Mandel und Tezaur 1996
- 16) Klawonn und Widlund 2001
- 17) Mandel und Tezaur 2001
- 18) Klawonn, Widlund und Dryja 2002a
- 19) Klawonn, Widlund, Dryja 2002b
- 20) vgl. Klawonn und Widlund 2004
- 21) Farhat, Lesoinne, LeTallec u. a. 2001
- 22) Mandel und Tezaur 2001
- 23) Farhat, Lesoinne und Pierson 2001
- 24) Klawonn, Widlund und Dryja 2002a
- 25) Klawonn, Widlund und Dryja 2002b
- 26) für numerische Ergebnisse zu diesen Algorithmen, vgl. Klawonn, Rheinbach und Widlund 2004

### Literatur

- Balay, S., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M., McInnes, L. C., Smith, B. F., Zhang, H.: PETSc home page 2001 <http://www.mcs.anl.gov/petsc>
- Braess, D.: Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie.<sup>3</sup> Berlin 2003
- Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., Rixen, D.: FETI-DP: A dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method, in: Int. J. Numer. Meth. Engrg. 50/2001, 1523–1544
- Farhat, C., Lesoinne, M., Pierson, K.: A scalable dual-primal domain decomposition method, in: Numer. Lin. Alg. Appl., 7/2000, 687–714
- Farhat, C., Roux, F.-X.: A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm, in: Int. J. Numer. Meth. Engrg. 32/1991, 1205–1227
- Frey, P.: Medit – An Interactive Mesh Visualization Software. Technical Report RT-0253, INRIA, Institut National de Recherche en Informatique et en Automatique, Dec 2001, <http://www-rocgl.inria.fr/gamma/medit/>
- Hendrickson, B., Leland, R.: The Chaco user's guide, Version 2.0. Technical Report SAND95-2344, Sandia National Laboratories, July 1995, <http://www.cs.sandia.gov/bahendr/chaco.html>
- Karypis, G., Schloegel, K., Kumar, V.: ParMetis-parallel graph partitioning and sparse matrix ordering library, Version 3.1, August 2003, User's Guide, <http://www-users.cs.umn.edu/~karypis/metis/parmetis/index.html>.
- Klawonn, A., Rheinbach, O., Widlund, O. B.: Some computational results for Dual-Primal FETI methods for three dimensional elliptic problems, in: R. Kornhuber, R.H.W. Hoppe, D.E. Keyes, J. Periaux, O. Pironneau, J. Xu (editors): Domain Decomposition Methods, Lecture Notes in Computational Science and Engineering, 2004. To appear in the proceedings of the 15<sup>th</sup> International Conference on Domain Decomposition Methods, Berlin, July 21–25, 2003
- Klawonn, A., Widlund, O.: Dual-Primal FETI methods for linear elasticity. Technical report, Courant Institute of Mathematical

### Skalierbarkeit und Speedup

Wenn ein Algorithmus sowohl auf einem, als auch auf vielen Prozessoren effizient ablaufen kann, spricht man von paralleler Skalierbarkeit. Viele Standardalgorithmen sind nicht gut skalierbar und für einige Probleme läßt sich beweisen, dass es keine skalierbaren Algorithmen geben kann.

Skalierbarkeit kann bedeuten, dass ein gegebenes Problem durch Einsatz von mehr Prozessoren immer schneller gelöst werden kann. Diese Art von Skalierbarkeit wird häufig durch den sogenannten Speedup gemessen, der zum Beispiel wie folgt definiert werden kann:

$$\text{Speedup} := \frac{t_1 \cdot p_1}{t_2}$$

Hierbei sind  $t_1$  und  $t_2$  die Zeiten, die der Algorithmus für das gleiche Problem mit  $p_1$  bzw.  $p_2$  Prozessoren benötigt. Der (theoretische) Optimalfall ist in dieser Definition gegeben, wenn der Speedup gleich  $p_2$  ist. Für ein Beispiel zum Speedup eines FETI-DP Algorithmus, vgl. Abb. (4).

Eine andere Sichtweise von paralleler Skalierbarkeit ist die Lösung immer größerer Probleme in gleicher Zeit durch Hinzunahme von mehr Prozessoren. Im optimalen Fall könnte man also ein doppelt so großes Problem (mit doppelt so vielen Gebieten bei gleich bleibender Anzahl Unbekannter pro Gebiet) durch Einsatz von doppelt so vielen Prozessoren in gleich bleibender Zeit lösen. Streng genommen gilt dies natürlich nur, solange das globale Problem klein bleibt. Dies ist ein wichtiger Bestandteil des Entwurfs paralleler skalierbarer Algorithmen.

FETI-Verfahren sind in diesem Sinne parallel skalierbar. Der tiefere Grund dafür liegt in ihrer sogenannten numerischen Skalierbarkeit, d.h. die Konvergenzgeschwindigkeit hängt bei den hier betrachteten FETI-Verfahren nur logarithmisch von der Anzahl der Unbekannten pro Teilgebiet ab, ist aber unabhängig von der Anzahl der Teilgebiete und der Größe des Gesamtproblems.

- Sciences, Department of Computer Science, 2004, in preparation
- Klawonn, A., Widlund, O.: FETI and Neumann-Neumann iterative substructuring methods: Connections and new results, in: Comm. Pure Appl. Math., 54/2001, 57–90
- Klawonn, A., Widlund, O.: Selecting constraints in Dual-Primal FETI for elasticity in three dimensions, in: R. Kornhuber, R.H.W. Hoppe, D.E. Keyes, J. Periaux, O. Pironneau, and J. Xu, editors, Domain Decomposition Methods. Lecture Notes in Computational Science and Engineering, 2004. Submitted to the proceedings of the 15<sup>th</sup> International Conference on Domain Decomposition Methods, Springer: Berlin, July 21<sup>st</sup>–25<sup>th</sup>, 2003
- Klawonn, A., Widlund, O., Dryja, M.: Dual-Primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients, in: SIAM J. Numer. Anal., 40/2002a, 159–179
- Klawonn, A., Widlund, O., Dryja, M.: Dual-Primal FETI methods with face constraints, in: Luca F. Pavarino and Andrea Toselli (editors): Recent developments in domain decomposition methods, 2002b, 27–40 (=Lecture Notes in Computational Science and Engineering, Volume 23)
- Mandel, J., Tezaur, R.: Convergence of a Substructuring Method with Lagrange Multipliers, in: Numer. Math. 73/1996, 473–487
- Mandel, J., Tezaur, R.: On the convergence of a dual-primal substructuring method, in:

- Numer. Math. 88/2001, 543–558
- Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations, Oxford Science Publications: Oxford 1999
- Rannacher, R., Stein, E.: Finite Elemente: die Ideen, in: Beste, D. (Hrsg.): Dossier Software, in: Spektrum der Wissenschaft 1999, 10–17
- Schöberl, J.: Netgen. <http://www.hp fem.jku.at/netgen>
- Barry, F., Smith, B. F., Bjørstad, P. E., Gropp, W.: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press: Cambridge 1996
- Toselli, A., Widlund, O. B.: Domain Decomposition Methods: Algorithms and Theory, Springer: Berlin 2004 (to appear)
- Zienkiewicz, O. C., Taylor, R.: The finite element method, McGraw-Hill: New York 1989

### Die Autoren

Axel Klawonn studierte Mathematik und Informatik an der Universität Münster, wo er auch promovierte. An seiner Dissertation arbeitete er etwa zu gleichen Teilen in Münster und den USA, wo er am Courant Institute of Mathematical Sciences in New York als visiting graduate Student studierte und dort insbesondere mit Olof Widlund arbeitete. Er habilitierte sich 2001 an der Universität zu Köln und erhielt 2002 einen Ruf auf eine Professur für Numerische Mathematik an die Universität Essen. Während seiner Habilitation war er als Wissenschaftlicher Assistent an der Universität Münster und als Nachwuchswissenschaftler am Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen (SCAI) in Sankt Augustin tätig. Axel Klawonn arbeitet an verschiedenen Forschungsvorhaben im Bereich des wissenschaftlichen Rechnens und der Numerik partieller Differentialgleichungen zusammen mit nationalen und internationalen Partnern. Er hielt sich zu längeren Forschungsaufenthalten im Ausland auf, unter anderem am Courant Institute in New York, an der University of Colorado at Boulder, USA und am Argonne National Laboratory, USA.

Oliver Rheinbach studierte Mathematik und Informatik sowie Wirtschaftsinformatik an der Universität zu Köln und Betriebswirtschaft an der McGill University, Montreal, Kanada. Er fertigte seine Diplomarbeit am Fraunhofer-Institut SCAI über FETI-Verfahren an und ist seit 2002 wissenschaftlicher Mitarbeiter im Fachbereich Mathematik der Universität Duisburg-Essen, Campus Essen. Im Jahr 2003 war er zu einem Forschungsaufenthalt an der Mathematics and Computer Science Division des Argonne National Laboratory, USA.

## **Zerreißen und Zusammenfügen**

Klawonn, Axel; Rheinbach, Oliver

In: UNIKATE: Berichte aus Forschung und Lehre / Heft 23 (2004)

Dieser Text wird über DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt.

Die hier veröffentlichte Version der E-Publikation kann von einer eventuell ebenfalls veröffentlichten Verlagsversion abweichen.

URN: [urn:nbn:de:hbz:464-20190301-122420-3](https://nbn-resolving.org/urn:nbn:de:hbz:464-20190301-122420-3)

Link: <https://duepublico.uni-duisburg-essen.de:443/servlets/DocumentServlet?id=48271>

Rechtliche Vermerke:

Sofern nicht im Inhalt ausdrücklich anders gekennzeichnet, liegen alle Nutzungsrechte bei den Urhebern bzw. Herausgebern. Nutzung - ausgenommen anwendbare Schrankenregelungen des Urheberrechts - nur mit deren Genehmigung.

Quelle: Druckausg. erschienen bei ESSENER UNIKATE 23, 2004, ISBN 3-934359-23-1