

CFD-Berechnung eines 13,5 m langen Lastwagens bei einer Geschwindigkeit von 90 km/h mit inkompressibler Strömung und κ - ϵ -Turbulenzmodell. Die Farbe der Strömungslinien macht Aussagen über deren Geschwindigkeit in m/s.

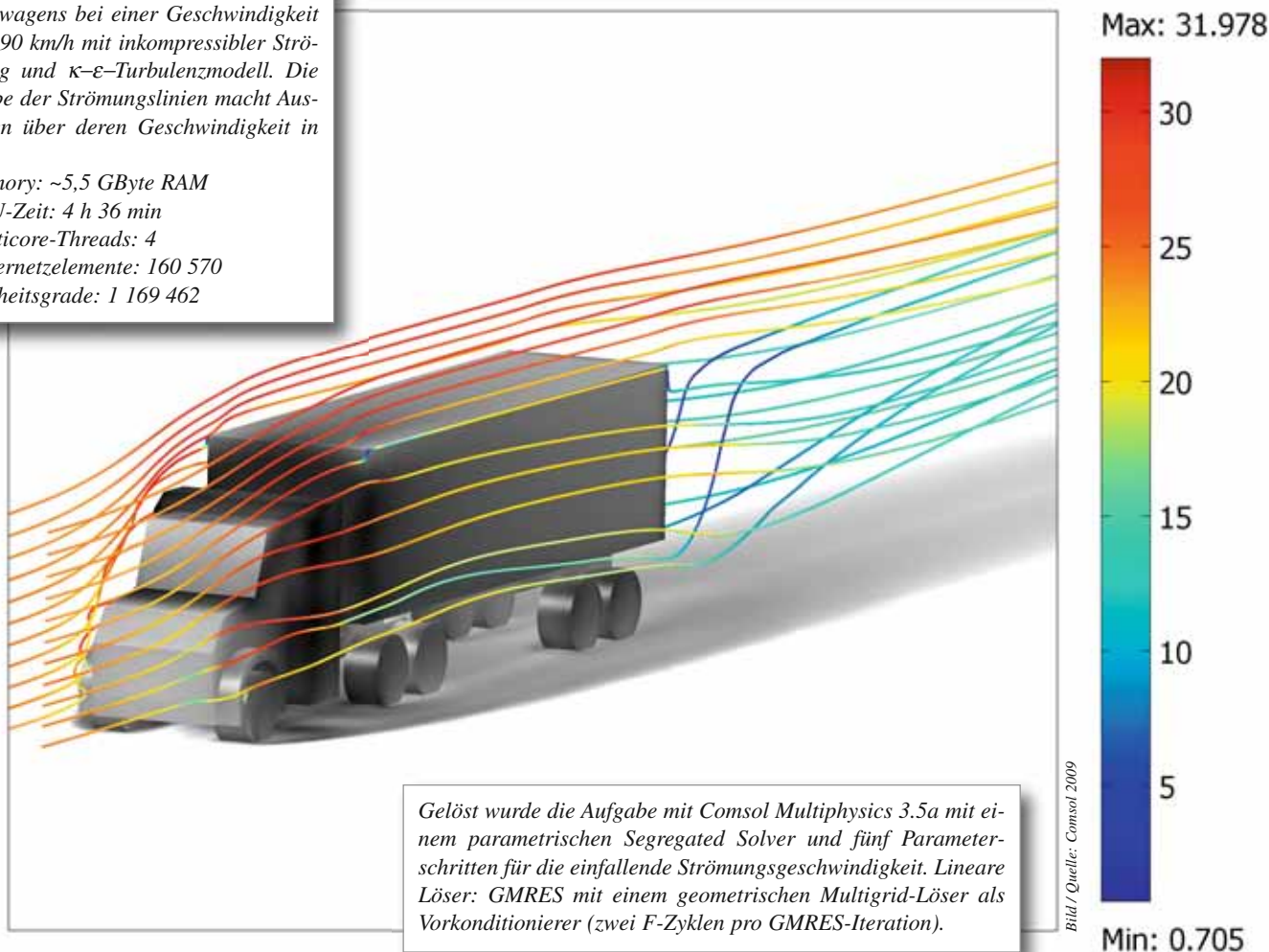
Memory: ~5,5 GByte RAM

CPU-Zeit: 4 h 36 min

Multicore-Threads: 4

Gitternetzelemente: 160 570

Freiheitsgrade: 1 169 462



Gelöst wurde die Aufgabe mit Comsol Multiphysics 3.5a mit einem parametrischen Segregated Solver und fünf Parameterschritten für die einfallende Strömungsgeschwindigkeit. Lineare Löser: GMRES mit einem geometrischen Multigrid-Löser als Vorkonditionierer (zwei F-Zyklen pro GMRES-Iteration).

Bild / Quelle: Comsol 2009

Starkes Duo vereint durch Best-in-Class Solver

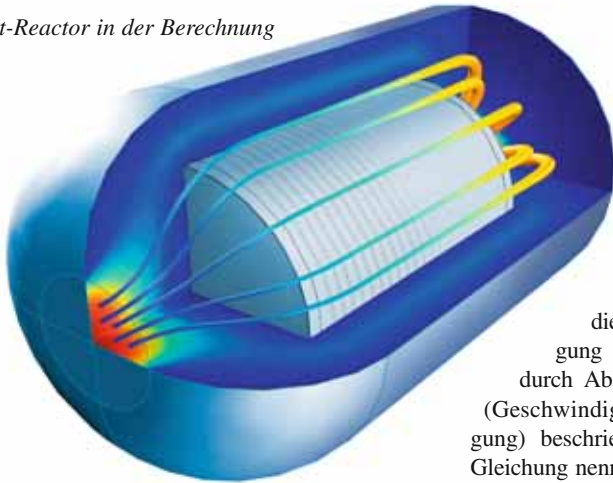
Ein kurzweiliger Ausflug in die Numerik zeigt auf, wo die Stärken von Comsol Multiphysics liegen. Hierzu ist ein Abriss über partielle Differentialgleichungen notwendig, dienen sie doch dazu, die physikalischen Phänomene korrekt mathematisch zu beschreiben. Aber auch ohne einen Überblick über die Finite-Elemente-Methode geht es nicht, schließlich ist sie der Garant dafür, dass man zu einem Analyseergebnis per Computer kommt. Perfekt vereint findet sich beides in Comsol Multiphysics.

Die Simulationsumgebung Comsol Multiphysics von Comsol AB mit Sitz in Stockholm unterstützt den gesamten Modellierungsprozess: von der Definition der Geometrie und der Spezifizierung der physikalischen Phänomene

über die Vernetzung und Löser-Technologien bis zur grafischen Aufbereitung der Ergebnisse. Dank der mitgelieferten Anwendungen, beispielsweise für die Analyse von Strömungsverhalten, strukturmechanischen oder elektro-

magnetischen Phänomenen geht der Modellaufbau und der eigentliche Analyseprozess zügig voran. Dem Anwender wird große Freiheit bei der Wahl von Materialeigenschaften, Quelltermen oder Randbedingungen gewährt. Natürlich gehört die intuitive Benutzerführung zu den erwähnenswerten Eigenschaften des CAE-Tools; uns freilich hat die Möglichkeit, selbst partielle Differentialgleichungen („partial differential equations“, PDEs) zu spezifizieren und sie mit anderen Gleichungen, etwa zur Beschreibung von ganz speziellen Materialeigenschaften, zu ergänzen, besonders gut gefallen. Herausragend an Comsol Multiphysics ist, dass der Analyseprozess über die PDEs auch

Boat-Reactor in der Berechnung



nach Beginn, quasi „on the fly“, noch beeinflusst werden kann – etwa um zusätzliche physikalische Effekte mitzuberechnen –, ohne dass die Berechnung komplett neu gestartet werden muss. Die Kopplung kann durch einfache mathematische Ausdrücke erfolgen, genau so, wie wenn man auf dem Blatt Papier einer Formel einen weiteren Term hinzufügt. Für diese intuitive Benutzerführung hat der Systemanbieter vom US-amerikanischen Patentamt sogar ein Patent erhalten (1). Der folgende Beitrag will eine Übersicht über die Kombination von Solver-Technologien geben, die das CAE-Tool in besonderem Maße auszeichnen.

Physik steht am Anfang

Lassen wir die Geometrie einmal außen vor, dann steht am Anfang jeder physikalischen Fragestellung die partielle Differenzialgleichung. Physiker lieben PDEs, zeigen sie doch sehr übersichtlich alle relevanten Phänomene der zu lösenden Aufgabenstellung. Das CAE-Tool muss nun effizient in Hinsicht auf Hardware-Ressourcen die richtigen Lösungen für die PDEs finden. Wie kommen PDEs zustande? Viele physikalische Prozesse hängen sowohl vom Ort als auch von der Zeit ab. Die

Veränderung bezüglich der Zeit ist jedoch in der Praxis nicht immer wichtig, da sich oftmals sehr schnell ein Gleichgewichtszustand einstellt, der unabhängig von der Zeit ist. In diesem Fall wird die Bewegung eines Massenpunkts nur durch Ableitungen nach der Zeit (Geschwindigkeit und Beschleunigung) beschrieben. So eine Art von Gleichung nennt man gewöhnliche Differenzialgleichung. Anders liegt der Fall beispielsweise bei Wellen, die durch einen Tropfen, der auf die Wasseroberfläche fällt, verursacht werden. Hier hängt die korrekte physikalische Beschreibung sowohl von der Zeitableitung (Geschwindigkeit der Welle) als auch von der Raum-ableitung (Profil der Welle) ab. Bei Ableitungen nach mehreren Variablen spricht man von PDEs (2). PDEs haben im Allgemeinen mehrere Lösungen. Um eine eindeutige Lösung zu erhalten, bedarf es gewisser Zusatzbedingungen, nämlich Rand- und/oder Anfangsbedingungen. Diese Situation ist vergleichbar mit gewöhnlichen Differenzialgleichungen, wo man ja auch Anfangsbedingungen in einem Punkt benötigt.

Bei PDEs gibt es auch einen Haken. Sie lassen sich – wenn überhaupt – analytisch (mit mathematischer Formelsammlung und Taschenrechner) nur bei geometrischen Grundkörpern wie Kreis oder Rechteck lösen. Bei anspruchsvolleren Geometrien müssen sie in ein lineares Gleichungssystem übergeführt werden, um vom Computer gelöst werden zu können. Hierzu wird das untersuchte Raumgebiet oder die Geometrie zunächst in Teilgebiete (finite Elemente) aufgeteilt. Auf diese Weise entstehen in 2D Dreiecke, in 3D im einfachsten Fall Tetraeder der betrachteten Geo-

metrie. Innerhalb des finiten Elements wird für die gesuchte Lösung eine feste Anzahl von sogenannten Ansatzfunktionen definiert, die nur in wenigen Teilgebieten ungleich Null sind. Durch eine Linearkombination dieser Ansatzfunktionen innerhalb des Elements werden die möglichen Lösungen der numerischen Näherung festgelegt und für jedes Element wird eine Elementmatrix aufgestellt. Die Lösungen können zum Beispiel durch Materialeigenschaften bestimmt sein.

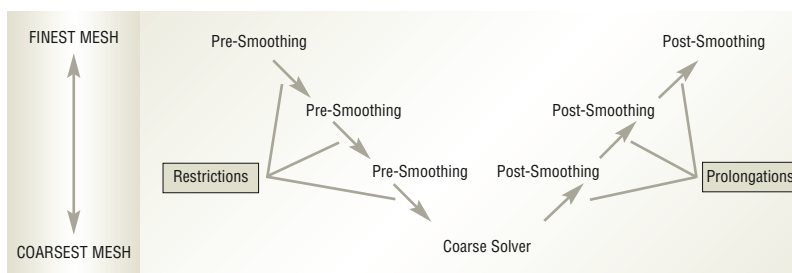
PDE und FEM

Die PDEs und die Randbedingungen werden mit Testfunktionen multipliziert und über das Lösungsgebiet integriert (schwache Form). Das Integral wird durch eine Summe über einzelne Integrale der finiten Elemente ersetzt (3). Da der Wirkungsbereich der Ansatzfunktionen nur auf wenige der Elemente beschränkt ist, ergibt sich ein dünnbesetztes, häufig sehr großes lineares Gleichungssystem, bei dem die Faktoren der Linearkombination unbekannt sind. Diese große Gesamtmatrix, auch Jacobi-Matrix genannt, bildet man, indem die Elementmatrizen aufaddiert werden. Die Gesamtmatrix ist quadratisch, hat also genauso viele Zeilen wie Spalten. Die Dimension der Matrix wird auch als Anzahl der Freiheitsgrade bezeichnet. Weil jedes Element nur mit wenigen benachbarten Elementen verbunden ist, sind die meisten Werte der Gesamtmatrix Null, so dass sie „dünnbesetzt“ ist. In den allermeisten Anwendungsfällen werden die gleichen Funktionen als Ansatz- und Testfunktionen verwendet. Ist die schwache Form zudem noch symmetrisch, so ist in diesem Fall auch die Matrix symmetrisch.

Es kommt zu folgendem linearem Gleichungssystem:

$$\begin{pmatrix} K & N^T \\ N & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

- K: Steifigkeitsmatrix, Jacobi-Matrix
- N: Constraint-Jacobi-Matrix, Nebenbedingung
- u: gesuchte Lösung
- λ: Langrange Multiplikator
- f: Residuum, Lastvektor
- g: Constraint-Residual-Vektor



Der Multigrid-V-Zyklus ist der Kern vieler Multigrid-Methoden, hier am Beispiel von vier Gitternetz-Feinheiten gezeigt.

Ist die Anzahl der Freiheitsgrade nicht zu groß (bis etwa 500 000), lässt sich dieses Gleichungssystem am effizientesten mittels eines direkten Verfahrens lösen. Hierzu gibt es eine Vielzahl von kommerziell verfügbaren Lösern, zum Beispiel auf Basis des Gauß'schen Eliminationsverfahrens. In Comsol Multiphysics stehen hierzu die direkten Löser UMFPACK (4), SPOOLES (5), TAUCS (6) und PARADISO (7) zur Verfügung. Sie sind einfach zu bedienen und robust, haben aber einen hohen Speicherbedarf. Sie eignen sich für die Berechnung von kleineren, stark nichtlinearen und Multiphysicsproblemen.

Für mehr als 500 000 Unbekannte bereitet der „Speicherhunger“ der direkten Löser zunehmend Schwierigkeiten, so dass man dann auf iterative Löser zurückgreift, die schrittweise die Rechnung verbessern. In Comsol Multiphysics sind dies GMRES, FGMRES, CG, BiCGStab und GMG (8,9).

Löser-Technologien

„Wir würden nie auf die Idee kommen, einen eigenen Löser zu entwickeln. Für ein derartiges Projekt wären hunderte von Entwicklern notwendig“, erläutert Ed Fontes, Vice President of Applications bei Comsol, im Gespräch mit der Redaktion. So verwundert es bei dieser Aussage nicht, dass auch iterative Löser von Comsol Multiphysics von extern übernommen wurden. Der Clou liegt ja auch ganz woanders: Fontes betont, dass es darauf ankomme, wie die Programmbausteine zusammengefügt würden – zum Beispiel bei der geometrischen Multigrid-Technologie GMG für die Strömungssimulation. „Oftmals sind die Methoden, eben wie bei GMG, in der Literatur nur rudimentär dokumentiert. Entscheidend ist, wie der Algorithmus programmiert wird. Da geht eine Menge an Erfahrung für Konditionierung und Stabilisierung ein“, sagt der Anwendungsexperte. Dass die Comsol-Experten um Jacob Yström hierbei ein gutes „Händchen“ bewiesen haben, zeigt die große Beachtung, die die Implementierungsform¹ von GMG und GMRES in Comsol Multiphysics gefunden hat.

Beim GMG wird eine Hierarchie von Gitternetzen durchlaufen, um die Lö-

sung für das am besten geeignete Gitternetz zu finden. Im sogenannten GMG-V-Zyklus wird sukzessive die Komplexität des ursprünglichen Gitternetzes reduziert, um mit einem Löser auf Basis eines vergrößerten Netzes effizient in Hinsicht auf RAM und CPU-Zeit die Aufgabenstellung zu berechnen. Im Anschluss daran erfolgt über einen „Glättungsalgorithmus“ des Residuums eine schrittweise Annäherung an die Ausgangsgeschwindigkeit des Gitternetzes, so dass die geforderte Genauigkeit in der Berechnung erreicht wird.

Der in Comsol Multiphysics implementierte GMG-Algorithmus kann als Vorkonditionierer mit einer festen Anzahl an Zyklen angewendet werden oder als separater linearer Löser, bei dem die GMG-Zyklen so oft angewendet werden, bis ein bestimmtes Konvergenzkriterium erreicht ist.

Der GMG-Algorithmus läuft entweder gemäß den Voreinstellungen automatisch ab, oder aber der Anwender passt ihn durch eine vielfältige Auswahl an Optionen an. So kann er auswählen, welcher Löser für die unterste Stufe der Vernetzung (das grobmaschigste Netz, also der tiefste Punkt im V-Zyklus) zum Einsatz kommen soll.

Bereits mit den GMG-Standard Einstellungen lässt sich ein breites Spektrum an Problemen lösen. Darüber hinaus gibt es aber Aufgabenstellungen, für die die konventionellen Pre-Konditionierer nicht geeignet sind. Daher stellt Comsol Multiphysics zwei weitere zur Verfügung: den sogenannten SOR-Vektor- und den Vanka-Präkonditionierer. Ersterer dient zur Lösung der Vektor-Helmholtz-Gleichung, da hier die Diskretisierung mit Vektorelementen durchgeführt wird. Die Vanka-Methode kommt bei bestimmten Formulierungen (Geschwindigkeitsdivergenz) der Navier-Stokes-Gleichung für inkompressible Flüssigkeiten zum Einsatz.

Auf dem Prüfstand

GMG beweist seine Stärken insbesondere beim sparsamen Umgang mit Hardware-Ressourcen. Dies lässt sich am Beispiel der Ausbreitung einer elektromagnetischen Welle in 3D zeigen. Wir zitieren hierzu eine Arbeit von Jacob Yström von Comsol (10), bei der die Helmholtz-Gleichung für den kubischen Reflektor mit drei verschiedenen

Algorithmen gelöst ist:

- direkter Löser SPOOLES
 - direkter Löser UMFPACK
 - iterative Methode GMRES mit GMG als Präkonditionierer (GMRES/MG).
- Der Vergleich zeigt, dass SPOOLES sich die im System vorhandene Symmetrie zunutze machen kann und daher nur die eine Hälfte der Jacobi-Matrix bearbeitet, während UMFPACK dies nicht kann (Figure 3). Die CPU-Zeit wird weiter verkürzt, wenn in Comsol Multiphysics über die Option „GMRES/MG“ die SOR- und SORU-Vektor-Algorithmen für die Prä- und Postglättung aktiviert werden (siehe V-Cycle-Diagramm). Die GMRES/MG-Methode verlangt keine weitere Anpassung und konvergiert typischerweise nach 20 bis 25 Iterationen. Die Ergebnisse sind in den Grafiken („Figure 3 und 4“) gezeigt. Interessant ist der Break-even, ab dem der CPU-Zeit- und RAM-Bedarf für GMRES/MG geringer ausfällt als für die direkten Löser. Im gewählten Beispiel liegt die Zahl bei ungefähr 50 000 komplexen Freiheitsgraden. Hier gibt es inzwischen weitere Verbesserungen, weil dies den Stand der Entwicklung von 2006 darstellt.

BERNHARD D. VALNION

INFOCORNER

- (1) Langemyr, L. et al., Patentnummer US 7 519 518 B2
- (2) http://de.wikipedia.org/wiki/Partielle_Differentialgleichung
- (3) <http://de.wikipedia.org/wiki/Finite-Elemente-Methode>
- (4) www.cise.ufl.edu/research/sparse/umfpack
- (5) www.netlib.org/linalg/spooles
- (6) www.tau.ac.il/~stoledo/taucs
- (7) www.computational.unibas.ch/cs/scicomp/software/pardiso
- (8) Stüben, K., Trottenberg, U., „Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications“, Lecture Notes in Mathematics, 960, 1982
- (9) Saad, Y., Schultz, M.H., „GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,“ SIAM J. Sci. Statist. Comput., Vol. 7, Seite 856ff., 1986
- (10) Yström, J., Comsol News 1/2006, Seite 32ff.

¹) Lesen Sie hierzu das Interview auf der nächsten Seite.

„Stets alles unter Kontrolle“

Jacob Yström, Solver-Experte bei Comsol, über die hohe Kunst, Rechneralgorithmen so anzupassen, dass Hardware-Ressourcen geschont werden, und was der Systemanbieter anders macht als der Wettbewerb.

Jacob Yström leitet das Team „Solver-Entwicklung“ der Entwicklungsabteilung von Comsol. Er hat einen Lehrauftrag für Numerische Analysis am Royal Institute of Technology in Stockholm

Herr Dr. Yström, gelten die Ergebnisse über die Performance von direkten und iterativen Lösern heute noch, die Sie vor drei Jahren veröffentlicht¹ haben? Die Tendaussagen, die wir damals gemacht haben, haben nach wie vor ihre Gültigkeit. Allerdings sollte nicht unerwähnt bleiben, dass uns heute Compute Cluster zur Verfügung stehen, die Parallelrechnen ermöglichen. Und Comsol

Multiphysics kann sowohl von Multi-Core- als auch von Distributed-Computing-Architekturen Gebrauch machen.

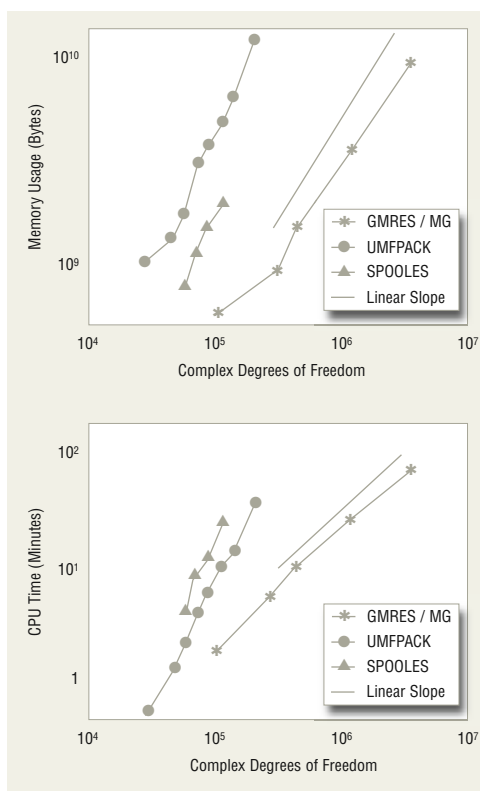
Haben Sie Ihre Untersuchungen in diese Richtung modifiziert?

Natürlich haben wir entsprechende Performance-Messungen durchgeführt. Wir verwenden sie allerdings nur für interne Zwecke.

Inwieweit müssen die Zah-

lenwerte in den Plots² verändert werden?

Die dargestellten Grafen geben die Trends richtig wieder. Sie können davon ausgehen, dass sich an den Angaben für den Speicherbedarf kaum, bei den CPU-Zeiten indes deutlich etwas geändert hat, beispielsweise weil heute Multi-Core-Architekturen genutzt werden können. Dies gilt für die



Quelle: Comsol News (10)

Vergleich von Memory-Ressourcen (oben) und CPU-Zeiten für eine 3D-Ausbreitung von elektromagnetischen Wellen bei der Reflexion an einem kubischen Metallreflektor. Zwar sind die Absolutwerte nicht mehr aktuell, wohl aber die ersichtlichen Trends. Anstatt von UMFPACK würde heute PARDISO als Benchmark verwendet.

PACK-Alternative, aber mehr noch für den neuen direkten Löser PARDISO. PARDISO stammt von Intel und nutzt die Multi-Core-Technologie besser als UMFPACK. Daher wäre für eine neue Version des Benchmarks PARDISO die erste Wahl. Übrigens ist PARDISO nicht nur schneller als UMFPACK, sondern braucht auch noch weniger Memory. Beim Speicherbedarf dürfte PARDISO gleichauf mit der SPOOLES-Alternative liegen. Und PARDISO dürfte um den Faktor 2 schneller sein als UMFPACK.

Lassen Sie mich kurz zusammenfassen: Selbst heute gibt es einen deutlichen Leistungsunterschied zwischen PARDISO und UMFPACK auf der einen und GMRES auf der anderen Seite.

Absolut richtig. Natürlich profitieren auch unsere iterativen Löser von der Multi-Core-Technologie. Der Trend bei der Skalierung zwischen linearen direkten und iterativen Lösern verändert sich beim Parallelrechnen nicht. In der Tat lässt sich argumentieren, dass es leichter ist, Performance-Gewinne bei iterati-

ven Lösern durch Parallelisierung zu erreichen als durch direkte Löser. Ich persönlich bin der Meinung, dass der Unterschied in Zukunft eher noch größer werden wird.

Um die Qualität von Löser-Technologien einzuschätzen: Welche Rolle spielen dabei Algorithmen, von denen ja die meisten öffentlich zugänglich sind?

Dazu muss ich ein wenig ausholen: Die partiellen Differenzialgleichungen (PDEs), die die physikalische Aufgabenstellung beschreiben, sind entweder zeitabhängig, stationär oder treten in Form von Eigenwert-Problemen auf. Durch die Anwendung der Finite-Elemente-Methode auf das PDE-Problem kommt es zu (stationär) algebraischen Gleichungen oder algebraischen Eigenwertgleichungen. Um diese zu lösen, haben wir eine Methodenstruktur entwickelt, die man im Allgemeinen Löser, im Englischen „Solver“, nennt. Es gibt sehr leistungsfähige Löser für die genannten PDE-Typen und einfachere Löser-Typen für lineare und nichtlineare algebraische Gleichungen. Die High-Level Solver machen intensiv Gebrauch von den einfacheren Löser-Typen. Darüber hinaus gibt es Löser für Optimierungsaufgaben und für Parameter-Probleme.

Gibt es darüber hinaus weitere Löser-Technologien?

O ja. Comsol Multiphysics verwendet sogenannte Utility Solver für die situationsbedingte Gitternetzanzpassung³ und für Sensitivitätsanalysen. Von diesen wurden der algebraische Eigenwertlöser ARPACK, einer der differenzialalgebraischen Löser (IDA), der Optimierungslöser SNOPT und der lineare algebraische direkte Löser von außen übernommen und in Comsol Multiphysics implementiert. Für allgemeine Matrixoperationen verwenden wir zudem die Standard-Bibliothek BLAS, alles Übrige stammt von uns selbst. Und nicht nur das – wir haben die vollständige Kontrolle über die linearen und nichtlinearen algebraischen Sub-Probleme, die bei der Verwendung von IDA und ARPACK auftreten. Des Weiteren nutzen wir eine eigene „PDE Engine“ für die PDE-basierte Optimierung mit SNOPT.

Wie stellt sich dies bezogen auf die verschiedenen Releases von Comsol Multiphysics dar?

¹) Yström, J., Comsol News 1/2006, Seite 32ff.

²) Siehe die Grafiken auf dieser Seite.

³) Im Englischen „adaptive mesh refinements“.

Seit der Version 3.1, als der Beitrag geschrieben wurde, kamen der Generalized Alpha Solver für die Berechnung von zeitabhängigen Problemen, der Sensitivity Solver und der Segregated Solver zur Lösung von algebraischen Problemen hinzu. Dabei würde ich die Unterstützung von zeitabhängigen und stationären Problemen durch den Segregated Solver als wichtigste Ergänzung bezeichnen.

Warum?

Zuvor konnte nur der Ansatz für vollständige Kopplung der PDEs und die Newton-Methode verwendet werden.

Nun, die Newton-Methode ist allgemein bekannt und etabliert...

Richtig. Sie ist ein Standard und einfach anzuwenden, falls die Jacobi-Matrix berechnet werden kann. Jedoch sind bei der Implementierung des Algorithmus eine Reihe von Details zu beachten. Die Newton-Methode ist dann optimal geeignet, wenn die Ausgangswerte sich nicht zu sehr von der gesuchten Lösung unterscheiden. Jedoch ist es oftmals gar nicht so einfach, in diese Situation zu kommen. Oftmals muss daher eine andere Lösungsstrategie angewendet werden.

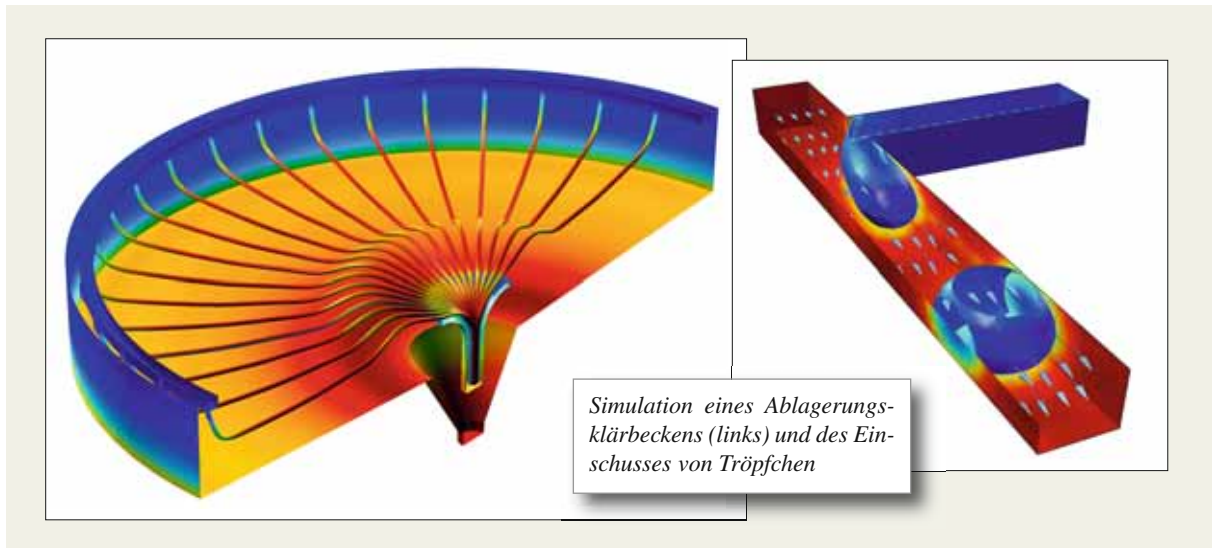
Sie brauchen uns ja keine Betriebsheimnisse zu verraten, aber können Sie uns einen Hinweis geben, was Sie bei der Newton-Methode in Comsol Multiphysics anders gemacht haben?

Wenn man etwas im Haus entwickelt, hat das den Vorteil, dass man jedes noch so kleine Detail im Algorithmus vollständig unter Kontrolle hat. Man hat keine Black Box vor sich, wo man nicht genau weiß, was passiert.

Wir wenden ja nicht die konventionelle Newton-Methode an, sondern eine gedämpfte Variante davon. Bei den linearen iterativen Lösern gehen wir ebenfalls eigene Wege. Wir haben sie zwar nicht erfunden – wir verwenden publizierte Ergebnisse –, aber wir haben spezielle Versionen implementiert, die uns geeignet erschienen. Übrigens geht die Originalversion unserer gedämpften Newton-Methode auf Arbeiten von Peter Deuffhard und Michael Hanke zurück, der am Royal Institute of Technology in Stockholm lehrt.

Wie vermeiden Sie Konvergenzprobleme bei der Newton-Methode?

Hierzu verwenden wir den sogenannten Segregated Solver. Er nutzt Iterationstechniken zwischen verschiedenen algebraischen Subproblemen. Sie ergeben sich durch bestimmte Gruppierung der Variablen in der Jacobi-Matrix. Während



*Simulation eines Ablagerungs-
klärbeckens (links) und des Ein-
schusses von Tröpfchen*

der Lösung eines Subproblems werden die anderen Variablen nicht verändert. Mit anderen Worten: Die volle Kopplung, die durch die gesamte Jacobi-Matrix repräsentiert ist, wird nicht berechnet, sondern lediglich die Kopplung innerhalb eines Unterraums.

Nennen Sie uns doch ein Beispiel.

Die Hauptmotivation für die Entwicklung des Segregated Solvers war die Lösung von den κ - ϵ -Turbulenzmodellen, wie sie in Strömungsberechnungen vorkommen. Ein anderes Beispiel sind Kontaktprobleme in der Strukturmechanik. Für Kontaktprobleme verwenden wir eine vereinfachte Version des Segregated Solvers, „Augmented Lagrange Solver“ genannt. Dieser wird allerdings erst in einer der späteren Versionen von Comsol Multiphysics implementiert.

Was bringt der Segregated Solver aus Sicht der ingenieurtechnischen Anwendung?

Der Unterschied ist riesig! Zuvor hatten wir keine für den Anwender praktikable Lösertechnologie für die genannten Anwendungsfelder. Besonderen Reiz hat unsere Lösung dadurch, dass der Anwender die Variable frei gruppieren kann und beliebig zwischen der vollständig gekoppelten Lösung und einem Segregated-Ansatz wählen kann.

Unterscheidet sich dieser Ansatz von den Lösungsstrategien Ihrer Wettbewerber?

Definitiv. Der Ansatz für die vollständige Kopplung ist am schwierigsten zu implementieren, da keine Kopplung vernachlässigt werden darf. Ich gehe davon aus, dass hier manche unserer Wettbewerber Schwierigkeiten haben, da sie zur Berechnung der unterschiedlichen

physikalischen Phänomene verschiedene Technologien verwenden. In diesem Fall besteht die einzige Möglichkeit darin, einen Segregated-Ansatz zu wählen. Wir hingegen haben die Segregation nur als eine Option implementiert, um bei nichtlinearen Problemen dem Anwender die volle Kontrolle beim Lösungsprozess zu geben. Wir haben sie nicht gewählt aufgrund von Beschränkungen in unserer Software.

Ich gehe davon, dass es nicht sehr viele Experten gibt, die mit derartig ausgefeilten numerischen Methoden umzugehen wissen. Auf der anderen Seite liegt es auf der Hand, dass gerade im Engineering der Trend in Richtung von „Realistic Simulation“ geht. Dies verlangt aber nach immer mehr Verständnis von physikalischen und numerischen Details. Wird der Ingenieur da nicht überfordert?

Sie haben recht – vom Feedback unserer Anwender wissen wir, dass der Umgang mit den Löseereinstellungen oftmals nicht einfach ist. Auf der anderen Seite gibt es sehr viele begabte Studenten und Wissenschaftler, die sich sehr intensiv mit Fragen der Numerischen Mathematik und des wissenschaftlichen Rechnens beschäftigen. Wir selbst müssen stets auf der Hut sein, nicht Dinge in unsere Software zu implementieren, die nur von Highend-Anwendern nachgefragt wurden. Wir dürfen nur Methoden einführen, von denen wir wissen, dass sie der effektiven Lösung von relevanten Aufgabenstellungen aus dem Engineering dienen.

Was also raten Sie?

Wer von den Ingenieuren in Berechnung und Simulation einsteigen will, sollte ein

oder zwei Grundstudiumssemester in Numerischer Analysis absolvieren und mit einer kommerziellen oder akademischen Berechnungssoftware anfangen zu arbeiten. Anhand der beigefügten Dokumentation sollte der Anwender ein Gefühl dafür bekommen, wie die implementierten Methoden arbeiten. Wichtig ist, anhand von einfachen Beispielen Erfahrungen mit den Lösern zu sammeln. Ich glaube nicht, dass der Berechnungsingenieur von morgen ein ausgewiesener Experte in Numerischer Mathematik sein muss. Zumindest ist dies die Auffassung, die wir hier bei Comsol vertreten.

Wie spiegelt sich dies in der Roadmap von Comsol Multiphysics wieder?

Unsere Produktentwicklung geht in die Richtung, mehr Voreinstellungen zu bieten. Wir haben erkannt, dass unsere sehr fortschrittlichen Lösertechnologien nur dann genutzt werden, wenn es eine intuitive Benutzerführung mit entsprechend geeigneten Default-Einstellungen gibt. Comsol ist für seine intuitive Benutzerführung bei der Auswahl der physikalischen Phänomene bekannt. Aber unsere Anwender tun sich schwer damit, bei den unterschiedlichen Lösern die umfangreichen Einstellmöglichkeiten zu nutzen, die wir zuvor diskutiert haben. Doch hieran arbeiten wir.

Was packen Sie als Nächstes an? Zum Beispiel bei der Version 4.0?

Zumindest kann ich so viel sagen: Die Benutzerführung wird sich bei der neuen Version deutlich von der in Version 3.5a unterscheiden.

Vielen Dank für das Gespräch!

Interview: BERNHARD D. VALNION