Teilprojekt D2

2.1 Teilprojekt D2

Effiziente parallele Algorithmen für die numerische Simulation 3–dimensionaler, stark phasengekoppelter, disperser Mehrphasenströmungen

2.1.1 Antragsteller

Dr.–Ing. Thomas Frank	Technische Universität Chemnitz Fakultät für Maschinenbau und Verfahrenstechnik Professur Technische Thermodynamik Forschungsgruppe Mehrphasenströmungen

2.1.2 Projektbearbeiter

Dr. sc. nat. Klaus Bernert	Professur Technische Thermodynamik
DI Klaus Pachler	Forschungsgruppe Mehrphasenströmungen

2.2 Ausgangsfragestellung/Einleitung

Die letzten zwei Jahrzehnte haben eine enorme Entwicklung im Bereich der numerischen Strömungsmechanik hervorgebracht. Zum einen sind Rechengeschwindigkeit und Speicherplatzangebot der Rechner um mehrere Größenordnungen gewachsen. Zum anderen gab es eine beständige Fortentwicklung der eingesetzten numerischen Simulationsverfahren bezüglich ihrer Effizienz und Genauigkeit sowie der zunehmenden Berücksichtigung komplexer physikalischer Wechselwirkungen.

Verfahren zur numerischen Simulation von phasengekoppelten, dispersen Mehrphasenströmungen unter Berücksichtigung von Stoff–, Impuls– und Wärmeübertragung zwischen den Phasen gehören zu den Anwendungen, die die höchsten Anforderungen an die Rechenleistung der Computersysteme stellen. Wegen der hohen Kosten, die mit dem Einsatz von Supercomputern verbunden sind, kam es in den letzten Jahren verstärkt zur Entwicklung von massiv parallelen Rechnerarchitekturen. Multiprozessorsysteme mit verteiltem Speicher (MIMD–Architekturen) erreichen durch den parallelen Einsatz vieler relativ einfacher Prozessoren eine Rechenleistung und Speicherkapazität, die die modernster Supercomputer zu übertreffen vermag. Um die potentiell sehr hohe Rechenleistung einer solchen Multiprozessorarchitektur bei der Berechnung strömungsmechanischer Problemstellungen auch wirklich zu erreichen, ist eine konzeptionelle Überarbeitung serieller Lösungsalgorithmen für ihren Einsatz auf einer bestimmten parallelen Hardwarearchitektur notwendig.

Als numerische Methode zur Berechnung von Einphasenströmungen kommt im vorliegenden Forschungsvorhaben ein Finite–Volumen–Verfahren zum Einsatz, welches über blockweise strukturierten Gitternetzen arbeitet. Für Verfahren dieses Typs hat sich die Methode der Gebietszerlegung als eine einfach zu realisierende und effizient anwendbare Parallelisierungsmethode etablieren können. Hierbei wird den Prozessoren der parallelen Maschine eine bestimmte Zahl von Gitterblöcken fest zugeordnet. Die Kopplung der Blöcke erfolgt über Datenaustausch innerhalb der Prozessoren und zwischen den Prozessoren.

Für die Berechnung von dispersen Mehrphasenströmungen auf massiv parallelen Rechnerarchitekturen waren bis vor kurzer Zeit noch keine Verfahren entwickelt bzw. veröffentlicht. Soweit zur Modellierung der dispersen Phase das Euler–Verfahren (Zwei–Fluid–Modell) herangezogen wird, sind die für Einphasenströmungen entwickelten Parallelisierungsverfahren unmittelbar übertragbar. Für das bei dispersen Mehrphasenströmungen häufig angewandte, und aus physikalischer Sicht vorteilhaftere Lagrange–Verfahren fehlten jedoch geeignete Ansätze zu dessen Parallelisierung. Auch heute sind neben den eigenen Veröffentlichungen (siehe Kapitel 2.4.3) nur wenige Arbeiten aus der internationalen Fachliteratur bekannt, die geeignete Parallelisierungsansätze für das Lagrange–Verfahren darstellen [2.13, 2.14, 2.15].

Für das Forschungsvorhaben D2 standen im Berichtszeitraum die folgenden Arbeiten im Mittelpunkt:

1. Erhöhung der Flexibilität und der parallelen Effizienz des parallelen Strömungslösers durch Übergang vom Ein-Block-Verfahren (jeder Prozessor bearbeitet genau einen Gitterblock) zu einem Multi-Block-Konzept.

2. Verbesserung der Effizienz des Strömungslösers mit Hilfe der Mehrgittertechnik.

3. Anpassung des Verfahrens zur Partikelberechnung an die aus dem Übergang zum Multi-Block-Konzept und der Implementierung des Mehrgitterverfahrens resultierende veränderte Speicherstruktur.

4. Erweiterung der Funktion des Programmsystems MISTRAL/PartFlow–3D auf instationäre Mehrphasenströmungen mit 2-Wege-Koppelung.

5. Einführung von skalaren Transportgleichungen als Tracer für Speciestransportvorgänge und einer Energiegleichung für instationäre Strömungen

Insgesamt hatten die Arbeiten das Ziel, ein effizientes und flexibles Verfahren zur Berechnung disperser Mehrphasenströmungen in komplexen Geometrien zu erarbeiten, mit dem die rechenintensive Simulation phasengekoppelter, disperser Mehrphasenströmungen auf einer breiten Palette von High–Performance–Computing (HPC)–Plattformen effektiv durchgeführt werden kann.

Die folgenden Kapitel zeigen die Vorgehensweise, und beschreiben den gegenwärtigen Stand der Realisierung der angegebenen Schwerpunkte. Des weiteren sei auf die nach 2.4.3 aufgeführten Veröffentlichungen verwiesen, in denen die wichtigsten Ergebnisse aus dem Teilprojekt D2 niedergelegt sind.

2.3 Forschungsaufgaben/Methoden

2.3.1 Teilaufgabe Steigerung der Effizienz des Strömungslösers

Grundgleichungen

Die kontinuierliche Phase der betrachteten Strömungen besteht aus einem Newton'schen Fluid. Die Fluidströmung ist inkompressibel, turbulent, statistisch stationär und isotherm. Die Fluidturbulenz wird mit Hilfe eines k-ε-Modells modelliert. Unter diesen Annahmen können die zeitlich gemittelten Gleichungen zur Beschreibung der Fluidströmung in Form der folgenden verallgemeinerten Transportgleichung dargestellt werden:

$$\frac{\partial}{\partial x_j}\left(\rho_F\, u_j^F\, \Phi\right) - \frac{\partial}{\partial x_j}\left(\Gamma\, \frac{\partial \Phi}{\partial x_j}\right) = S_F + S_P, \quad j = 1, 2, 3. \tag{3.1}$$

Hierbei ist ρ_F die Dichte des Fluids, u_j^F sind die kartesischen Komponenten der Fluidgeschwindigkeit, Φ ist eine allgemeine Variable und Γ der zugehörige Transportkoeffizient. S_F beschreibt einen allgemeinen Quellterm, S_P modelliert die Wechselwirkung zwischen disperser und fluider Phase. Eine genauere Beschreibung der in Gleichung (3.1) auftretenden Größen wird durch Tabelle 2.1 gegeben, dabei bezeichnen ρ_F und μ die Dichte und die kinematische Viskosität des Fluids, p ist der Druck und f_i sind die Komponenten der pro Masseneinheit einwirkenden äußeren Kräfte.

Gleichung für	Φ	Γ	S_F	S_P
Masse	1	0	0	0
Impuls	u_i^F	μ_{eff}	$\frac{\partial}{\partial x_j}\Big(\Gamma\frac{\partial u_j^F}{\partial x_i}\Big) - \frac{\partial p}{\partial x_i} + \rho_F f_i$	$S_{u_i}^P$
Turbulente kinetische Energie	k	$\frac{\mu_t}{\sigma_k}$	$P_k - \rho_F\,\varepsilon$	0
Dissipation von k	ε	$\frac{\mu_t}{\sigma_\varepsilon}$	$\frac{\varepsilon}{k}\left(c_{\varepsilon_1} P_k - c_{\varepsilon_2}\,\rho_F\,\varepsilon\right)$	0
$P_k = \mu_t \frac{\partial u_i^F}{\partial x_j}\Big(\frac{\partial u_i^F}{\partial x_j} + \frac{\partial u_j^F}{\partial x_i}\Big);\ \mu_{eff} = \mu + \mu_t,\ \mu_t = \rho_F c_\mu \frac{k^2}{\varepsilon}$				
$c_\mu = 0.09,\quad c_{\varepsilon_1} = 1.44,\quad c_{\varepsilon_2} = 1.92,\quad \sigma_k = 1.0,\quad \sigma_\varepsilon = 1.3$				

Tabelle 2.1: Strömungsvariablen, Transportkoeffizienten und Quellterme

Numerische Lösung

Ausgangspunkt für die im Berichtszeitraum vorgenommenen Arbeiten war ein im Programmpaket MISTRAL/3D implementierter Strömungslöser mit folgenden Eigenschaften (vgl. [2.28]):

- Vernetzung des dreidimensionalen Lösungsgebietes durch ein blockstrukturiertes hexagonales Gitter

- Finite-Volumen-Dikretisierung der zu lösenden Gleichungen
- Iterative Lösung des Gleichungssystems durch das SIMPLE-Verfahren [2.6, 2.7]
- Verwendung des SIP-Verfahrens [2.12] als Gleichungslöser
- Parallelisierung des numerischen Verfahrens für Computersysteme mit MIMD-Struktur auf der Grundlage des Gebietszerlegungsverfahrens.

Zur Steigerung der Effizienz des beschriebenen Verfahrens wurde das bestehende Konzept an zwei Stellen erweitert. Ein erster Schritt diente der gleichmäßigeren Verteilung der Rechenarbeit auf die Prozessoren einer parallelen Rechenanlage und bestand in der Änderung des Grundsatzes „je Prozessor ein Block“ auf das Multiblock-Prinzip, bei dem jeder Prozessor eine variable Anzahl von Blöcken nacheinander bearbeitet. Den zweiten Schwerpunkt bildete der Einsatz der Mehrgittertechnik zur Beschleunigung der Konvergenz der Verfahrens.

Multiblock-Konzept

Der entscheidende Schritt beim Übergang zum Multiblock-Konzept war die Umstellung der zuvor benutzten Speicherung von 3D-Variablen in separaten Feldern auf Strukturen, welche Pointer für alle zu einem Block gehörigen Variablen enthalten. Jeder Prozessor arbeitet über einem Feld von solchen Strukturen, wobei die Feldelemente den auf dem Prozessor liegenden Blöcken entsprechen. Durch die Abbildung einer zusätzlichen Arbeitsstruktur vom gleichen Typ auf die Strukturen für die einzelnen Blöcke wird die Arbeit der Prozessoren mit ihren Blöcken organisiert. Das folgende Programmstück zeigt eine typische Schleife für die Arbeit eines Prozessors:

```
for (n=1; n<=number_of_local_blocks; n++)
{   set_pointers_for_block(n);
    work_on_block(n);
}
```

Die Zuordnung der Blöcke zu den Prozessoren erfolgt statisch auf der Grundlage eines heuristischen Prinzips, welches aus der Optimierung von Maschinenbelegungen als LPT-Regel (largest processing time first) bekannt ist: Es wird vorausgesetzt, daß die Anzahl der Blöcke die Zahl der Prozessoren übersteigt. Die Blöcke werden nach der Zahl ihrer Kontrollvolumen geordnet. Nun wird, beginnend mit dem größten, ein Block nach dem anderen jeweils einem der Prozessoren zugeordnet, für den die Zahl der Kontrollvolumen aller schon zugeordneten Blöcke minimal ist.

Mehrgitterverfahren

Für die Anwendung der Mehrgittertechnik bestanden im Programmsystem MISTRAL/3D zwei Möglichkeiten, zum einen die Beschleunigung der äußeren SIMPLE-Iterationen, d.h. die Anwendung eines äußeren Mehrgitterverfahrens auf das gesamte Gleichungssystem, und zum anderen die Beschleunigung der Konvergenz der inneren Iterationen für die Druckkorrekturgleichung. Beide Möglichkeiten wurden, da sie sich nicht gegenseitig ausschließen, so implementiert, daß sie miteinander kombiniert werden können.

Grundlage für die Anwendung des Mehrgitterverfahrens war eine nochmalige Erweiterung der Speicherstruktur, um den Zugriff auf die verschiedenen Gitterebenen durch einfache Adressverschiebungen zu realisieren. Das äußere Mehrgitterverfahren benutzt das ursprüngliche SIMPLE-Verfahren als Glättungsverfahren und als Grobgitterlöser, Restriktion und Prolongation werden jeweils trilinear ausgeführt. Im inneren Mehrgitterverfahren für die Druckkorrektur werden die Gleichungen für die gröberen Gitter durch Addition der entsprechenden Feingittergleichungen ermittelt, da für die gröberen Gitter keine Differentialgleichung vorliegt. Das ursprünglich genutzte SIP-Verfahren dient als Glätter, als Grobgitterlöser wird ein mit einer unvollständigen Cholesky-Zerlegung vorkonditioniertes CG-Verfahren eingesetzt. Auch hier finden trilineare Interpolationen für Restriktion und Prolongation Verwendung.

2.3.2 Teilaufgabe Parallele Partikelberechnung

Zur Modellierung der Bewegung der dispersen Phase wird ein von der Forschungsgruppe in D2 entwickeltes 3–dimensionales Euler–Lagrange–Verfahren verwendet. Hierbei wird die Bewegung einer großen Anzahl diskreter Partikel berechnet, wobei jedes simulierte Partikel wiederum eine große Anzahl realer Partikel repräsentiert, die dieselben physikalischen Eigenschaften besitzen. Entlang einer so berechneten Trajektorie bewegen sich pro Zeiteinheit eine Anzahl Partikel N_P, d.h. jede Trajektorie repräsentiert einen bestimmten Anteil des gesamten Partikelmassenstroms $\dot{m}_{P,ges}$.

Unter der Annahme, daß die Dichte der dispersen Phase wesentlich größer als die Dichte der fluiden Phase ist ($\rho_P/\rho_F \gg 1$), sind die wichtigsten auf ein Partikel wirkenden Kräfte die Widerstandskraft, die Auftriebskraft durch die Rotation des Partikels (Magnus-Kraft), die Auftriebskraft durch Scherantrömung des Teilchens (Saffman-Kraft), die Gravitationskraft und der hydrostatische Auftrieb. Die Bewegungsgleichungen für die translatorische und rotatorische Bewegung einer Partikel lauten damit:

$$\frac{d}{dt}\,\vec{x}_P = \vec{v}_P \tag{3.2}$$

$$\begin{aligned}\frac{d}{dt}\,\vec{v}_P = \; & \frac{3}{4}\frac{\rho_F}{(\rho_P+\frac{1}{2}\rho_F)d_P}\Bigg(v_{rel}C_D(Re_P)\left[\vec{v}_F-\vec{v}_P\right] \\ & + \frac{v_{rel}}{\omega_{rel}}C_M(\sigma)\cdot\left[(\vec{v}_F-\vec{v}_P)\times(\vec{\omega}-\vec{\Omega})\right] \\ & + \frac{2\nu_F^{1/2}}{\pi|\vec{\Omega}|^{1/2}}C_A\left[(\vec{v}_F-\vec{v}_P)\times\vec{\Omega}\right]\Bigg) + \frac{\rho_P-\rho_F}{\rho_P+\frac{1}{2}\rho_F}\,\vec{g}\,,\end{aligned} \tag{3.3}$$

$$\frac{d}{dt}\,\vec{\omega} = -\frac{15}{16\pi}\frac{\rho_F}{\rho_P}\omega_{rel}\xi_m(Re_\omega)\left[\vec{\omega}-\vec{\Omega}\right]\,. \tag{3.4}$$

$$\text{mit}: \quad Re_P = \frac{d_P\,v_{rel}}{\nu_F}\,, \quad Re_\omega = \frac{1}{4}\frac{{d_P}^2\omega_{rel}}{\nu_F}\,, \quad \sigma = \frac{1}{2}\frac{d_P\omega_{rel}}{v_{rel}}\,,$$

$$\vec{\Omega} = \frac{1}{2}\mathrm{rot}\;\vec{v}_F\,, \quad v_{rel} = |\vec{v}_F-\vec{v}_P|\,, \quad \omega_{rel} = |\vec{\omega}-\vec{\Omega}|\,.$$

In diesen Gleichungen bezeichnet der Index F Fluidgrößen, der Index P Größen, die sich auf das Partikel beziehen, $\vec{\omega}$ ist die Rotationsgeschwindigkeit des Partikels, $\vec{\Omega}$ die Rotation des Fluides am Partikelort und ν_F die kinematische Viskosität des Fluides. Die Größen der Koeffizienten C_D, C_M, C_A und ξ_m können der Literatur entnommen werden [2.18, 2.10, 2.2].

Zur Modellierung der Partikel–Wand–Kollisionen wurde hier ein Wandrauhigkeitsmodell nach [2.9] verwendet. Die Modellierung von Partikel–Partikel–Kollisionen erfoglte mit Hilfe von Stoßwahrscheinlichkeitsmodellen nach Oesterle [2.3] und Sommerfeld [2.11]. Zur Modellierung des Einflusses der Fluidturbulenz auf die Partikelbewegung wurde das sogenannte Lagrangian-Stochastic-Deterministic (LSD) Tubulenzmodell verwendet [2.8]. Der Einfluß der dispersen Phase auf das Fluid wurde durch die Einführung von Quelltermen in die Fluidtransportgleichungen nach der Particle-Source-In-Cell (PSI-Cell) Methode [2.1] berücksichtigt. Eine detailliertere Beschreibung der eingesetzten Teilmodelle und des Lösungsverfahrens findet sich z.B. in [2.18, 3.5].

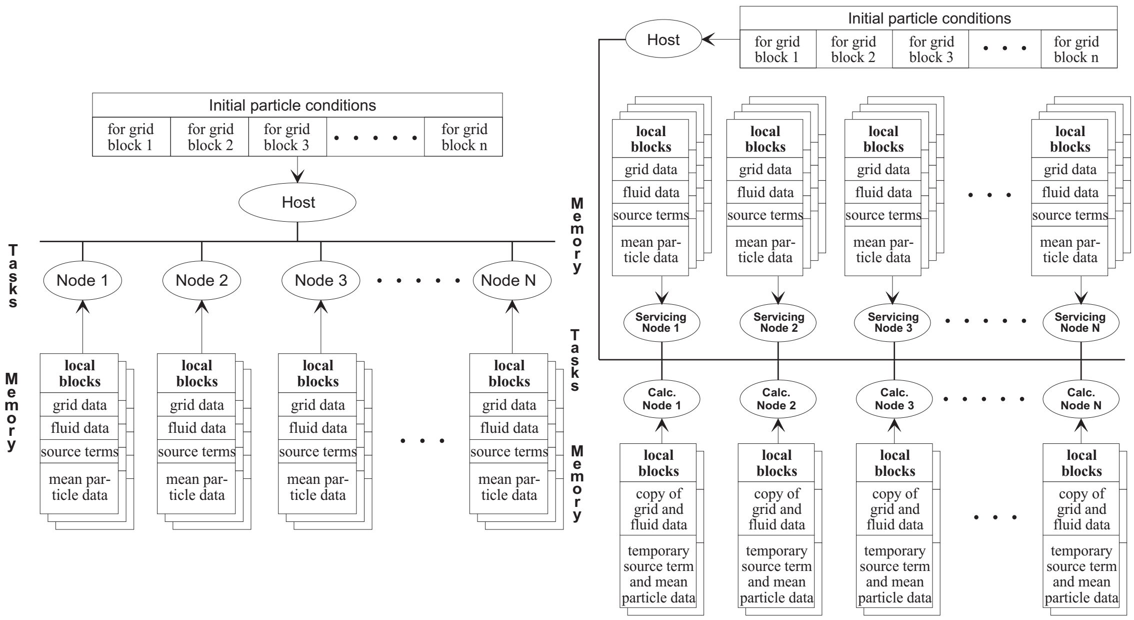

Abbildung 2.1: SDD– und DDD–Verfahren für die Parallelisierung des Lagrange'schen Berechnugnsverfahrens.

Parallelisierungsverfahren mit Statischer Domain Decomposition

Neben der Möglichkeit, die Berechnung der Fluid–Partikel–Strömung in einem seriellen Modus auszuführen, wurde zunächst für das 3–dimensionale Lösungsverfahren die sogenannte Statische Domain Decomposition (SDD) implementiert. Um das Lagrange-Verfahren auch auf massiv parallelen Rechnern mit ihrer pro Prozessorknoten beschränkten Speichergröße verwenden zu können, wurde zunächst das für den Navier–Stokes–Löser angewandte Prinzip der Gebietszerlegung auch auf das Lagrange–Berechnungsverfahren angewandt. Wie in Abb. 2.1 dargestellt, existieren zwei Arten von Prozessen: Ein Host-Prozeß und eine Anzahl von Node-Prozessen. Die Speicherstruktur für die für jeden der Node–Prozesse lokal und permanent gespeicherten Daten der Gittergeometrie und Fluidströmung entspricht dabei exakt derselben Struktur wie im Navier–Stokes–Löser. Eine Umverteilung der Datenstrukturen findet beim Übergang von der Berechnung der Fluidströmung zur Berechnung der Partikelbewegung nicht statt. Die Trajektorienberechnung wird von den Node-Prozessen durchgeführt, wobei ein Node alle Trajektoriensegmente berechnet, die die ihm zugeteilten Gitterblöcke durchqueren.

Der Host verteilt zu Beginn der Rechnung die Partikelstartbedingungen entsprechend den Startpositionen an die Node-Prozesse. Die Startbedingungen werden nur an solche Nodes gesendet, die die Daten des korrespondierenden Blockes gespeichert haben, in dem Partikelstartpositionen lokalisiert sind. Ein Node berechnet ein Trajektoriensegment vom Eintrittspunkt in den Block (Einströmrand oder Partitionsrand) bis zum Austrittspunkt aus dem Block (Ausströmrand oder Partitionsrand). Liegt der Austrittspunkt an einem Partitionsrand, so wird der Partikelstatus in diesem Punkt zunächst an den Host-Prozeß gesendet, der den Partikelstatus dann an den betreffenden Node-Prozeß weiterleitet. Dort wird dann die Trajektorienberechnung fortgesetzt. Diese Verfahrensweise wird wiederholt, bis alle Trajektorien eine Abbruchbedingung erreicht haben. Gleichzeitig zur Berechnung der Trajektorie erfolgt die Berechnung der Quelltermwerte für die Phasenwechselwirkungsterme.

Das SDD-Verfahren ist einfach zu implementieren, da es dieselbe Block–Prozessor–Zuordnung verwendet wie der parallelisierte Navier-Stokes-Löser. Der wesentliche Nachteil des SDD–Verfahrens besteht darin, daß keinerlei Mechanismen für eine Lastbalancierung existieren. Daraus resultierende ungleichmäßige Lastverteilungen zwischen den Prozessoren (mit den entsprechenden negativen Folgen für die parallele Effizienz des Verfahrens) kann resultieren aus :

1. Unterschiedliche Rechenleistung der rechnenden Prozessoren, z.B. bei heterogenen Workstation-Clustern.
2. Unterschiedlich große Blöcke des numerischen Gitters, d.h. verschiedene Anzahlen von Gitterzellen pro Prozessor.
3. Inhomogene Partikelverteilung im Strömungsgebiet, z.B. bei Freistrahlströmungen hinter Düsen, bei rezirkulierenden oder stark separierten Strömungen.
4. Partikel-Wand-Kollisionen, da in der Nähe der Wand der Integrationszeitschritt verkleinert wird. Außerdem erzeugt die Detektierung and Berechnung des Wandstoßes einen zusätzlichen Rechenaufwand.
5. Lokal höhere Konzentration der Partikelphase und daraus resultierende höhere Wahrscheinlichkeit von Partikel–Partikel–Kollisionen; dies führt zu lokal höherem Berechnungsaufwand auf dem betreffenden Prozessor.
6. Strömungsgebiete mit großen Gradienten der Fluidgeschwindigkeit und daraus resultierendem kleinen turbulenten Zeitmaß. Dies führt aus Gründen der Rechengenauigkeit ebenfalls zu einer Verkleinerung des Integrationszeitschritts in diesen Gebieten.

Die ersten beiden Punkte sind allen Domain Decomposition Verfahren gemeinsam und gelten auch für den parallelisierten Navier-Stokes-Löser. Die Punkte 3. - 6. gelten speziell für das SDD–Verfahren für das Lagrange'sche Berechnungsverfahren. Die dadurch verursachten Inhomogenitäten in der Verteilung des Rechenaufwands im Strömungsgebiet sind im allgemeinen zu Beginn der Rechnung nicht bekannt. Als Folge der statischen Zuordnung der Gitterblöcke zu den Prozessoren kann es zu einer inhomogenen Verteilung der Rechenlast unter den Prozessoren kommen, was die Effizienz der Parallelisierung reduziert.

Parallelisierungsverfahren mit Dynamischer Domain Decomposition

Das DDD–Parallelisierungsverfahren wurde entwickelt, um die angeführten Nachteile des SDD–Verfahrens hinsichtlich Lastbalancierung zu beseitigen. Abb. 2.1 zeigt das Prozessorschema dieses Verfahrens, das drei Klassen von unterschiedlich agierenden Prozessen aufweist : den Host-Prozess, die Service- und die eigentlichen Rechenprozesse. Dabei sind die sogenannten Service-Prozesse identisch mit den MPI-Prozessen, die innerhalb des Navier–Stokes–Lösers die Berechnung der Fluidströmung ausführen und operieren demzufolge auch auf denselben Datenstrukturen mit identischer Block–Prozessor–Zuordnung. Die Kommunikationsstruktur der Service–Prozesse ist als permanente Message–Loop ausgebildet, so daß diese Prozesse keine eigentliche Berechnungsarbeit leisten, sondern lediglich die Anforderungen der Rechenprozesse ständig entgegennehmen und beantworten. Die Rechenprozesse werden erst zur Laufzeit des Lagrange–Lösers in den Lösungsprozeß einbezogen und starten zunächst mit einer leeren Datenstruktur hinsichtlich lokal vorhandener Geometrie– und Fluidströmungsdaten. Nach der Zuteilung von Berechnungsaufgaben bzgl. der Partikelbewegung durch den Host-Prozess aquirieren diese Rechenprozesse dynamisch die hierzu benötigten Daten von demjenigen Service–Prozess, der in seiner lokalen Speicherstruktur über die entsprechenden Daten verfügt.

Die mittels standard-konformer MPI–Prozeduren empfangenen Daten werden dynamisch in die Block–Speicherstruktur des Rechenprozesses eingeordnet und stehen von nun an für einen bestimmten Zeitraum für Berechnungen der Partikelbewegung lokal zur Verfügung. Ändern sich die Berechnungsanforderungen an den Rechenprozess, so werden die Daten zunächst in einer Datenpipeline endlicher Länge aufbewahrt (Caching). Beim Überlauf dieser Datenpipeline werden die dynamischen Speicherstrukturen für die am längsten nicht mehr benötigten Geometrie– und Strömungsdaten aus der lokalen Speicherstruktur des Rechenprozesses entfernt.

Durch die Aufhebung des unmittelbaren Zusammenhangs zwischen dem Speicherort der Geometrie– und Fluidströmungsdaten und dem Ort der Partikeltrajektorienberechnung auf dem entsprechenden Gitterblock werden zunächst zwei wesentliche Dinge gegenüber dem SDD–Verfahren erreicht : 1. ein Trajektoriensegment auf einem bestimmten Gitterblock der Strömungsgeometrie kann durch einen beliebigen Rechenprozess des MIMD–Systems berechnet werden, und 2. liegen mehrere Anfangsbedingungen für ein und denselben Gitterblock vor, so sind diese von verschiedenen Rechenprozessen zeitgleich bearbeitbar. Die Folge ist eine dynamische Lastbalancierung des Parallelrechnersystems für den Lagrange'schen Berechnungsalgorithmus mit folgenden Eigenschaften :

- Unabhängigkeit der parallelen Effizienz des Lagrange'schen Berechnungsverfahrens von der konkreten Gebietszerlegung,
- Unabhängigkeit der Lastverteilung und damit der Effizienz des Algorithmus von der konkreten Fluid–Partikel–Strömung, wie z.B. von lokaler Phasentrennung/ –entmischung, nicht homogener Konzentrationsverteilung im Strömungsgebiet, lokal auftretender starker Partikel–Wand– oder Partikel–Partikel–Wechselwirkung,
- funktionierende Lastbalancierung auch auf heterogenen MIMD–Systemen wie z.B. Workstation–Clustern mit unterschiedlicher Rechenleistung der eingebundenen Prozessorknoten.

Es ist noch zu erwähnen, daß die Service-Prozesse und der Host-Prozess nicht auf eigenständigen physischen Prozessoren ausgeführt werden brauchen. Der Bedarf an Rechenleistung ist bei diesen Prozessen recht gering, so daß sich eine Superposition dieser Prozesse mit den N Rechenprozessen auf denselben N physischen Prozessoren ohne Einbuße an der Effizienz des DDD–Verfahrens bewährt hat, wenngleich Untersuchungen gezeigt haben, daß nicht alle MPI–Implementierungen eine solche Prozessüberlagerung auf einem physischen Prozessorknoten gleich gut unterstützen. Hier mußte LAM–MPI gegenüber MPICH in allen verfügbaren Versionen der Vorzug gegeben werden, da die Dämonprozesse von MPICH unabhängig von der tatsächlichen Rechenlast des MPI–Benutzer–Prozesses annähernd 100% der CPU–Leistung verbrauchen (siehe hierzu auch [3.7]).

2.3.3 Teilaufgabe Instationäre Mehrphasenströmungen

Die Instationärität einer Strömung wird von unterschiedlichen Kriterien beeinflußt. Die einfachsten Ursachen sind zeitlich veränderbare Randbedingungen (Einlaßmassenstrom) oder Geometrien (bewegter Kolben). Des weiteren gibt es Strömungen(vortex shading), die eine in der Zeit periodische Druckverteilung hinter der Ablösekante generieren. Bei Mehrphasenströmungen induziert die disperse Phase auf Grund der Wechselwirkung in der kontinuierlichen Phase eine transiente Änderung des Strömungsfelds.

Grundgleichungen

Es gelten die unter 2.3. definierten Annahmen, wobei für den Fall der instationären Strömungen Erweiterungen vorgenommen wurden. Die allgemeine Transportgleichung lautet:

$$\frac{\partial}{\partial t}\Big(\rho_F\,\Phi\Big) + \frac{\partial}{\partial x_j}\Big(\rho_F\,u_j^F\,\Phi\Big) - \frac{\partial}{\partial x_j}\left(\Gamma\,\frac{\partial \Phi}{\partial x_j}\right) = S_F + S_P,\ \ j = 1,2,3. \qquad (3.5)$$

Gleichung (3.5) unterscheidet sich von Gleichung (3.1) durch den zusätzlichen ersten Term. S_F beschreibt einen allgemeinen Quellterm, S_P modelliert die Wechselwirkung zwischen disperser und fluider Phase, Eine genauere Beschreibung der in Gleichung (3.5) auftretenden Größen wird durch Tabellen 2.1 und 2.2 gegeben.

Gleichung	Φ	Γ	S_F	S_P
Masse	1	0	0	$\sum \frac{\partial m_k^P}{\partial t} N_k^P$
Impuls	u_i^F	μ_{eff}	$\frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial u_j^F}{\partial x_i}\right) - \frac{\partial p}{\partial x_i} + \rho_F f_i$	$\sum \frac{\partial (m_k^P u_k^P)}{\partial t} N_k^P$
Skalar	c_i	$\frac{\mu_t}{\sigma_c}$	0	$c_i \sum \frac{\partial m_k^P}{\partial t} N_k^P$
Enthalpie	h	$\frac{\mu_t}{\sigma_h}$	$\mu P_k/\mu_t + \rho_F \varepsilon$	0
$h = c_p T + \|u_i\|^2/2 + k$, $\sigma_c = 1.0$, $\sigma_h = 0.9$, $0 \le c_i \le 1$				

Tabelle 2.2: Strömungsvariablen, Transportkoeffizienten und Quellterme

Die Energiegleichung wird mittels der totalen Enthalpie h dargestellt und erlaubt eine einfache Koppelung mit der dispersen Phase. Die Quellterme S_F beinhalten den Beitrag für das viskose Heizen bei inkompressibler Strömung und den Einfluß der Dissipation nach dem k-ε-Modell.

Die Wechselwirkungsterme werden für jedes Rechenpartikel pro Kontrollvolumen berechnet und mit der physikalischen Partikelanzahl N_k^P multipliziert. Eine Skalargleichung wird zum Transport der Dampfphase verwendet. Eine weitere Skalargleichung ohne Quellterme wird als Tracer für die kontinuierliche Phase verwendet. Für Masse und Skalar wird der Massenübergang als Austauschterm berechnet.

Die Kopplung der Impulsgleichung läuft über den Austausch der Oberflächenkräfte (Widerstand, Druckgradient, virtuelle Masse), während die Körperkräfte (Gravitation, elektrische Ladung) unberücksichtigt bleiben. Die Diskretisierung erfolgt in einer semiimpliziten Weise [2.16], wobei sich die Geschwindigkeit der kontinuierlichen Phase in den Austauschtermen während der SIMPLE-Iterationen ändert, was eine Stabilisierung der Impulskoppelung bewirkt [2.32].

Instationärer Löser

Basis der Arbeiten war der stationäre Löser (siehe 2.3.1., 2.3.2., sowie [2.28]). Dieser wurde so erweitert, daß über einen Auswahlschalter der instationäre Löser optional aktiviert werden kann. Folgende Arbeiten wurden sowohl für die serielle als auch für die parallele Version von MISTRAL/PartFlow–3D durchgeführt:

- Implementation einer I/O-Prozedur, die das Modifizieren der wichtigsten Steuerparameter des Lösers in Abhängigkeit von der Realzeit der Berechnung ermöglicht
- Implementation von instationären Anfangsbedingungen für die disperse Phase
- Implementation einer Prozedur für die Verarbeitung der transienten Ausgabedaten der kontinuierlichen und der dispersen Phase im Zusammenhang mit dem kommerziellen Postprocessor Tecplot

- Erweiterung aller Erhaltungsgleichungen der kontinuierlichen Phase um den Instationärterm.
- Modifikationen im Partikellöser zur Abstimmung der Partikelintegrationszeit mit dem globalen, aktuellen Zeitschritt und zur Berücksichtigung von Partikel–Partikel–Kollisionen unter Verwendung der Modelle von Oesterle [2.3] und Sommerfeld [2.11].

Zeitdiskretisierung der kontinuierlichen Phase

Der Zeitterm in Gl. (3.5) wird als lokaler Term behandelt, wobei eine Mittelung über das Kontrolvolumen durchgeführt wird.

$$\int_V \frac{\partial(\rho_F\,\Phi)}{\partial t} dV = \overline{\frac{\partial(\rho_F\,\Phi)}{\partial t}} V \tag{3.6}$$

$$\text{mit} \quad \overline{\partial(\rho_F \Phi)/\partial t} \approx \partial(\rho_F \Phi_P)/\partial t$$

Diese Approximation für Φ auf der Basis der Finite–Volumen erlaubt eine Entwicklung nach der Taylor-Reihe für die Zeitkoordinate. Die Lösung zum Zeitpunkt $t + \Delta t$ kann nur von einem früheren Zustand abhängen. Die Anzahl der verwendeten Stützstellen in der Zeit zur Approximation und die Wahl von Vorwärts- oder Rückwärtsdifferenzen bestimmen die Eigenschaften des Verfahrens [2.6]. Folgende 3 Verfahren werden unterschieden:

$$\int_t^{t+\Delta t} \Phi(t)dt \cong [\alpha\Phi^{t+\Delta t} + (1-\alpha)\Phi^t]\Delta t \qquad [0 \le \alpha \le 1] \tag{3.7}$$

- $\alpha = 0$ ...... explizites Verfahren (vorwärtiges Euler-Verfahren)
- $\alpha = 0.5$ ... zeitgemitteltes Verfahren (Crank-Nicolson-Verfahren)
- $\alpha = 1$ ...... implizites Verfahren (rückwärtiges Euler-Verfahren)

Die aktuelle Mistral-Implementation enthält ein implizites Verfahren($\alpha = 1$), daß sich durch seine Stabilität (keine Zeitschrittbegrenzung!) auszeichnet. Der Speicherplatzbedarf ist zwar höher gegenüber expliziten Verfahren, doch die größere Robustheit ist entscheidend.

Das Volumenintegral über die partielle Zeitableitung der Funktion $\Phi(t, X_i)$ wird durch die mit $1/\Delta t$ multiplizierte Differenz der zu den zwei Zeiten aus einer Volumenmittelung gebildeten Werten von Φ approximiert.

$$\frac{1}{\Delta t} \int_t^{t+\Delta t} \int_{V_P} \frac{\partial}{\partial t}(\rho_F\,\Phi) dV_P dt \approx \frac{(V\rho_F\,\Phi)_P^{t+\Delta t} - (V\rho_F\,\Phi)_P^t}{\Delta t} \tag{3.8}$$

Die Gleichung (3.5) in linearisierter Form für primitive Variablen lautet für das Kontrolvolumen um den Zentralpunkt P:

$$A_P\Phi_P = \sum_{nb} A_{nb}\Phi_{nb} + S_\Phi \tag{3.9}$$

A_{nb} sind die Nachbarkoeffizienten und resultieren aus Konvektion und Diffusion. Der Zeitterm wird wie der Quellterm behandelt, wobei folgende Linearisierung über den Iterationsschritt n verwendet wird:

$$S(\Phi) = S_0^n + A_{S,P}^n(\Phi_P^{n+1} - \Phi_P^n) \quad \text{mit } A_{S,P}^n \approx \frac{\partial S^n}{\partial \Phi} \quad \text{und } A_{S,P}^n < 0 \tag{3.10}$$

Der Polkoeffizient A_P setzt sich aus Gl. (3.7) und Gl. (3.10) wie folgt zusammen:

$$A_P = \sum_{nb} A_{nb} + \frac{(\rho_F V)_P^{t+\Delta t}}{\Delta t} - A_{S,P}^n V_P \tag{3.11}$$

Der lineare Quellterm lautet:

$$S_\Phi = (S_0 - A_{S,P}^n \Phi) V_P + \frac{(\rho_F \Phi V)_P^t}{\Delta t} \tag{3.12}$$

Literaturverzeichnis

[**Zitierte Fremdliteratur**]

[2.1] Crowe, C.T., Sharma, M.P., Stock, D.E.: *The Particle–Source–In Cell (PSI–Cell) Model for Gas–Droplet Flows*, Trans. of ASME, J. Fluids Eng., Vol. 99, 1977, pp. 325–332.

[2.2] Crowe, C.T., Sommerfeld, M., Tsuj, Y.: *Multiphase Flows with Droplets and Particles*, CRC Press, Boca Raton, 1997.

[2.3] Fohanno, S., Oesterle, B.: *Analysis of the Effect of Collisions on the Gravitational Motion of Large Particles in a Vertical Duct*, Int. J. Multiphase Flows, Vol. 26, No. 2, pp. 267-292 (2000).

[2.4] Durst, F., Fischer, M., Jovanovic, J., Kikura, H., Lange, C.: *LDA measurments in the wake of a circular cylinder*, Notes in Numerical Fluid Dynamics,Ed. by Hirschel, Vieweg-Verlag .

[2.5] Turek, M., Schäfer, S.: *Benchmark Computations of Laminar Flow around a Cylinder*, Preprint 96-03(SFB359), Heidelberg (1996).

[2.6] Patankar, S.V.: *Numerical Heat Transfer and Fluid Flow*, McGraw–Hill, New York., 1980

[2.7] Perić, M.: *Ein zum Parallelrechnen geeignetes Finite–Volumen–Mehrgitterverfahren zur Berechnung komplexer Strömungen auf blockstrukturierten Gittern mit lokaler Verfeinerung*, Abschlußbericht zum DFG–Vorhaben Pe 350/3–1 im DFG–Habilitandenstipendiumprogramm, Stanford University, USA. 1992

[2.8] Schönung, B.: *Comparison of Different Dispersion Models for Particles in Lagrangian and Eulerian Prediction Codes*, Proceedings of the International Conference on Fluid Mechanics, Peking, July 1.-4., 1987, Peking University Press, China.

[2.9] Sommerfeld, M.: *Modelling of particle–wall collisions in confined gas–particle flows*, Int. Journal of Multiphase Flows, Vol. 18, No. 6, 1992, pp. 905–926.

[2.10] Sommerfeld, M.: *Modellierung und numerische Berechnung von partikelbeladenen turbulenten Strömungen mit Hilfe des Euler–Lagrange–Verfahrens,* Habilitationsschrift, Universität Erlangen/Nürnberg, Shaker Verlag, Aachen (1996).

[2.11] Sommerfeld, M.: *Inter-Particle Collisions in turbulent flows: a stochastic Lagrangian Model,* Conf. Turbulence and Shear Flow Phenomena, 1st Int. Symposium, Santa Barbara, CA, USA, Sept. 1999.

[2.12] Stone, H.L.: *Iterative solution of implicit approximations of multidimensional partial differential equations*, SIAM J. Numer. Anal., 5, 530-558, 1968.

[2.13] Tsuji, Y.: *Discrete Element Modelling of Clusters in Gas–Solid Flows,* Proc. 2nd Int. Symposium on Numerical Methods for Multiphase Flows, ASME Fluids Engineering Division Summer Meeting, San Diego, CA, USA, July 7–11, 1996, FED–Vol. 236, Vol. 1, p. 3.

[2.14] Tysinger, T.L., Missaghi, M.: *A Combined Shared–Memory and Distributed–Memory Model for Computation of Coupled Lagrangian Dispersed Phase and Eulerian Gas Phase Combustion,* Proc. Int. Conf. on Recent Developments and Advances Using Parallel Computers, "Parallel CFD '97", Manchester, England, May 19–21, 1997.

[2.15] Yonemura, S., Tanaka, T., Tsuji, Y.: *Cluster Formation in Gas–Solid Flow Predicted by the DSMC Method,* Proc. Int. Symposium on Gas–Solid Flows, ASME Fluids Engineering Conference, Washington D.C., USA, 1993, FED–Vol. 166, pp. 303–309.

[2.16] Beshay, K. R.: *Computer Modelling of Diesel Spray Dispersion and Evaporation,* PhD. Thesis, Imperial College London, (1985).

[**Veröffentlichungen zu eigenen Vorarbeiten (vor 1999)**]

[2.17] Bernert, K.: *τ-Extrapolation – Theoretical Foundation, Numerical Experiment and Application to Navier-Stokes Equations,* SIAM J. Sci. Comput. Vol. 18, No. 2, 1997.

[2.18] Frank, Th.: *Numerische Simulation der feststoffbeladenen Gasströmung im horizontalen Kanal unter Berücksichtigung von Wandrauhigkeiten,* Dissertation, Techn. Universität Bergakademie Freiberg, 1992.

[2.19] Frank, Th., Schade, K.P., Petrak, D.: *Numerical Simulation and Experimental Investigation of a Gas–Solid Two–Phase Flow in a Horizontal Channel,* Int. J. Multiphase Flow, Vol. 19, No. 1, 1993, pp. 187–198.

[2.20] Frank, Th.: *Comparison of three parallelization methods for calculation of disperse multiphase flows using the Lagrangian approach,* Proc. 3rd Int. Conf. "Parallel CFD '96", Implementations and Results Using Parallel Computers, Capri, Italy, May 20–23, 1996.

[2.21] Frank, Th. Wassen, E.: *Parallel Solution Algorithms for Lagrangian Simulation of Disperse Multiphase Flows,* Proc. 2nd Int. Symposium on Numerical Methods for Multiphase Flows, ASME Fluids Engineering Division Summer Meeting, San Diego, CA, USA, July 7–11, 1996, FED–Vol. 236, Vol. 1, pp. 11–20.

[2.22] Frank, Th. Wassen, E.: *Parallele Algorithmen für die numerische Simulation von Mehrphasenströmungen in komplexen 3–dimensionalen Geometrien,* Abschlußbericht zum DFG–Forschungsvorhaben im Normalverfahren, Projektnr.: Fr 1069/3-1, TU Chemnitz–Zwickau, FG Mehrphasenströmungen, Chemnitz, 1. Juni 1997.

[2.23] Frank, Th. Wassen, E., Yu, Q.: *A 3–dimensional Lagrangian Solver for Disperse Multiphase Flows on Arbitrary, Geometrically Complex Flow Domains using Block–structured Numerical Grids,* 7th Int. Symposium on Gas-Particle Flows, ASME Fluids Engineering Division Summer Meeting, Vancouver, BC, Canada, June 22–26, 1997, CD–ROM Proceedings, FEDSM97–3590.

[2.24] Frank, Th. Wassen, E.: *Parallel Efficiency of PVM– and MPI–Implementations of two Algorithms for the Lagrangian Prediction of Disperse Multiphase Flows,* JSME Centennial Grand Congress 1997, ISAC '97 Conference on Advanced Computing on Multiphase Flow, Tokyo, Japan, July 18–19, 1997.

[2.25] Frank, Th. Wassen, E., Yu, Q.: *Ein blockstrukturiertes Verfahren zur Berechnung disperser Gas–Feststoff–Strömungen in komplexen 3–dimensionalen Geometrien,* VDI–Gesellschaft Verfahrenstechnik und Chemieingenieurwesen, GVC–Jahrestagung 1997, Prozeß– und Umwelttechnik, Dresden, 24.–26. September 1997. In Chemie–Ingenieur–Technik, Vol. 69, Nr. 9/97, pp. 1270–1271.

[2.26] Frank, Th. Wassen, E., Yu, Q.: *Lagrangian Prediction of Disperse Gas–Particle Flow in Cyclon Separators,* ICMF '98 – 3rd International Conference on Multiphase Flow 1998, Lyon, France, June 8.–12., 1998, CD–ROM Proceedings, Paper No. 217, pp. 1–8.

[2.27] Frank, Th. Wassen, E., Yu, Q.: *Numerische Untersuchung der Strähnenbildung und Erosion in 3–dimensionalen Gas–Feststoff–Strömungen,* 49. Berg– und Hüttenmännischer Tag, TU Bergakademie Freiberg, Germany, 18./19. Juni 1998, in: Freiberger Forschungshefte.

[2.28] Frank, Th. Wassen, E., Yu, Q.: *Numerische Simulation auf massiv parallelen Rechnern,* Arbeits- und Ergebnisbericht 1996 - 1997 - 1998, SFB 393, TU Chemnitz (1998)

[2.29] Frank, Th., Schulze, I.: *Numerical simulation of gas–droplet flow around a nozzle in a cylindrical chamber using Lagrangian model based on a multigrid Navier–Stokes solver,* Proc. Int. Symposium on Numerical Methods for Multiphase Flows, ASME Fluids Engineering Division Summer Meeting, Lake Tahoe, NV, USA, June 19–23, 1994, FED–Vol. 185, pp. 93–107.

[2.30] Schade, K.–P., Th. Frank, Th., Hädrich, Th., Petrak, D.: *An Experimental and Numerical Study of the Two Phase Flows in Sprays with Hollow Cone Nozzles and Full Cone Nozzles,* PARTEC'95 — International Congress for Particle Technology, 11th Europe an Conference of ILASS–Europe on Atomization and Sprays, 21.–23. March 1995, Nürnberg, Germany.

[2.31] Wassen, E., Frank, Th., Yu, Q.: *A Comparison of Parallel Algorithms for the Numerical Simulation of Multiphase Flows,* The 1. Euro–Conference on Parallel and Distributed Computing for Computational Mechanics. Lochinver, Scotland, April 26 – May 1st, 1997.

[2.32] Brandstätter, W., Landertshamer, F., Pachler, K.: *Recent Findings Regarding High Pressure Diesel Spray Simulations,* ILASS–Europe on Atomization and Sprays, 6.–8. July 1998, Manchester, England.

2.4 Ergebnisse

2.4.1 Teilaufgabe Steigerung der Effizienz des Strömungslösers

Multiblock-Konzept

Mit dem Multiblock-Konzept entspricht das Programmsystem wesentlich besser den Anforderungen der Praxis als zuvor. Im Gegensatz zu einigen akademischen Testbeispielen ist bei realen Anwendungsaufgaben die Zahl der Blöcke im Ergebnis der Gittergenerierung relativ groß (häufig zwischen 100 und 200) und übersteigt die Anzahl der effektiv einsetzbaren Prozessoren (32 bis 64). Durch das implementierte heuristische Verfahren zur Lastbalancierung wurde in allen Fällen eine gute bis sehr gute Verteilung der Blöcke auf die Prozessoren erreicht, bei Anwendungsaufgaben ergeben sich für Strömungsberechnungen auf 32 Prozessoren parallele Effizienzen um 0.6. Die Einarbeitung weiterer Optimierungskriterien, z.B. durch den Versuch, benachbarte Blöcke nach Möglichkeit auf den gleichen Prozessor zu legen, ist denkbar, stößt durch die feste Vorgabe der Blöcke jedoch auf sehr enge Grenzen und wurde deshalb nicht untersucht.

Mehrgitterverfahren

Zuerst wurde das Mehrgitterverfahren zur Beschleunigung der Konvergenz der inneren Iteration für die Druckkorrektur implementiert. Im Ergebnis zeigte sich, daß dadurch nicht nur die innere Iteration wesentlich effizienter wurde, sondern daß gleichzeitig die für das ursprüngliche SIMPLE-Verfahren typische langsame und nichtmonotone Abnahme der Residuen für das Gesamtsystem in ein wesentlich schnelleres monotones Fallen der Residuen überging. Für die Berechnung einer statistisch stationären turbulenten Strömung in einem graden Kanal mit quadratischem Querschnitt konnte so bei 64*64*256=1 048 576 finiten Volumen die Rechenzeit auf 8 Prozessoren auf reichlich 6% verringert werden (vgl. Abb. 2.2).

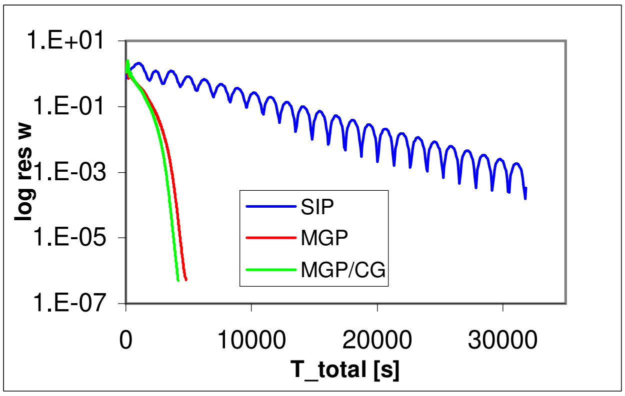

Abbildung 2.2: Konvergenz des SIMPLE-Verfahrens mit unterschiedlichen Lösungsverfahren für die Druckkoreekturgleichung; SIP – SIP-Verfahren, MGP – Mehrgitterverfahren, MGP/CG – MGP-Verfahren mit vorkonditionierter CG-Iteration auf dem gröbsten Gitter

Mit der Anwendung des Mehrgitterverfahrens auf die äußere Iteration und damit auf das Gesamtsystem der zu lösenden Gleichungen wurde eine weitere Beschleunigung der Konvergenz ereicht. Bei der zuvor angeführten Testaufgabe ergab sich eine nochmalige Verringerung der Rechenzeit auf ein Viertel des Wertes für das verbesserte SIMPLE-Verfahren (s. Abb. 2.3). Neben dem Mehrgitterverfrahren wurde auch das Full-Multigrid-Verfahren erfolgreich getestet, darüber hinaus besteht die Möglichkeit, Testrechnungen auf den im Mehrgitterverfahren verwendeten gröberen Gittern auszuführen.

Durch eine Reihe von Testrechnungen wurden die Eigenschaften des Mehrgitterverfahrens untersucht. So konnte gezeigt werden, daß die Zahl der Iterationen für eine vorgegebene Genauigkeit unabhängig von der Feinheit des Gitters ist und mit der Anzahl der Blöcke nur geringfügig steigt. Ein Vergleich der parallelen Effizienz zwischen einer CRAY-T3E (TU Dresden) und dem Chemnitzer Linux-Cluster CLIC (TU Chemnitz) fällt wegen des schnelleren Kommunikations-Netzwerks eindeutig zugunsten der CRAY aus. Auf dem CLIC mit FastEthernet werden gute parallele Effizienzen erreicht (0.5 bis 0.8), solange die Anzahl der finiten Volumen je Prozes-

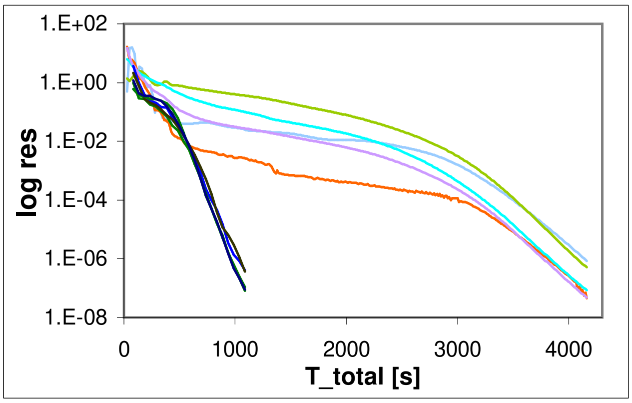

Abbildung 2.3: Konvergenz des Eingitterverfahrens (graue Kurven) und des Mehrgitterverfahrens (scharz), beide mit innerem Mehrgitterverfahren für den Druck, Residuen für u, w, Masse, k, ϵ

sor bei 50000 und darüber liegt. Eine ausführlichere Darstellung der ausgeführten Tests mit weiteren Grafiken und Tabellen ist in ([3.8]) veröffentlicht.

Neben der ober angeführten einfachen Kanalströmung wurden Strömungen in einem gekrümmten Kanal, in mehrfach abgewinkelten Kanälen, in Rohrverzweigungen und in Zyklonen bei Verwendung von bis zu 3 174 400 finiten Volumen auf maximal 128 Prozessoren berechnet. In allen Fällen erbrachte die Mehrgittertechnik eine signifikante Verringerung der Rechenzeiten gegenüber dem ursprünglichen Verfahren.

2.4.2 Teilaufgabe Parallele Partikelberechnung

Die Implementierung der in Abschnitt 2.3.2 und 2.3.2 beschriebenen Parallelisierungsverfahren für das Euler–Lagrange–Verfahren basieren auf explizitem Message Passing und einer auf Standard–MPI basierenden eigenen Kommunikationsschnittstelle. Im Ergebnis entstand ein Programmsystem, das mit sehr geringem Anpassungsaufwand auf die unterschiedlichsten Parallelrechenrsysteme portiert werden kann, wie dies im zurückliegenden Zeitraum mit Cray–T3D/T3E, SGI Origin 2000, Parsytec–GC und verschiedenen Cluster–Systemen mit den unterschiedlichsten Kommunikationsnetzwerken (SCI, Myrinet, GigaNet, FastEthernet) vorgenommen wurde.

Für die Bewertung der parallelen Effizienz der SDD– und DDD–Verfahren wurde zum einen die Cray–T3E der TU Dresden (64 DEC Alpha 21164 Prozessoren, 128 Mb Knotenspeicher, 3d–Torus–Verbindungsnetzwerk) verwendet. Demgegenüber standen zunächst ein lokal verfügbares Athlon–Prozessor–Cluster mit maximal 12 Prozessoren (512 Mb Knotenspeicher, FastEthernet–Netzwerk). Die Untersuchungen für größere Knotenanzahlen wurden letztendlich auf dem Chemnitzer Linux–Cluster (CLIC) mit bis zu 128 Prozessoren durchgeführt.

Für die vergleichenden Untersuchungen wurden zwei Testfälle definiert, bei denen es sich um eine Gas–Partikel–Strömung in einem dreifach gekrümmten Kanal mit quadratischem Querschnitt handelt. Im Testfall 1 sind die Kanalkrümmer mit jeweils 4 Krümmerschaufeln ausgestattet, die eine Entmischung beider Phasen weitgehend verhindern, so daß eine hinsichtlich der Partikelkonzentrationsverteilung weitgehend homogene Strömung die Folge ist. Im 2. Testfall sind die Krümmerschaufeln nicht vorhanden. Die Folge sind Sekundärströmungen nach jedem der drei Kanalkrümmer, die eine starke Entmischung der Partikelphase bewirken. Im Vergleich dieser beiden Testfälle ist die dynamische Lastbalancierung des DDD–Verfahrens gut zu beurteilen. Das ursprünglich aus 21 respektive 9 Gitterblöcken bestehende Gitternetz mit $80 \times 80 \times 496 = 3\,174\,400$ Gitterzellen wurde für die Untersuchungen einheitlich in 64 Gitterblöcke unterteilt, die je nach verwendeter Parallelrechner–Partition auf die Prozessoren des MIMD–Systems verteilt wurden. Für jede der Berechnungen wurden 5000 Partikeltrajektorien mit einer homogenen Partikelkonzentrationsverteilung im Einströmgebiet berechnet.

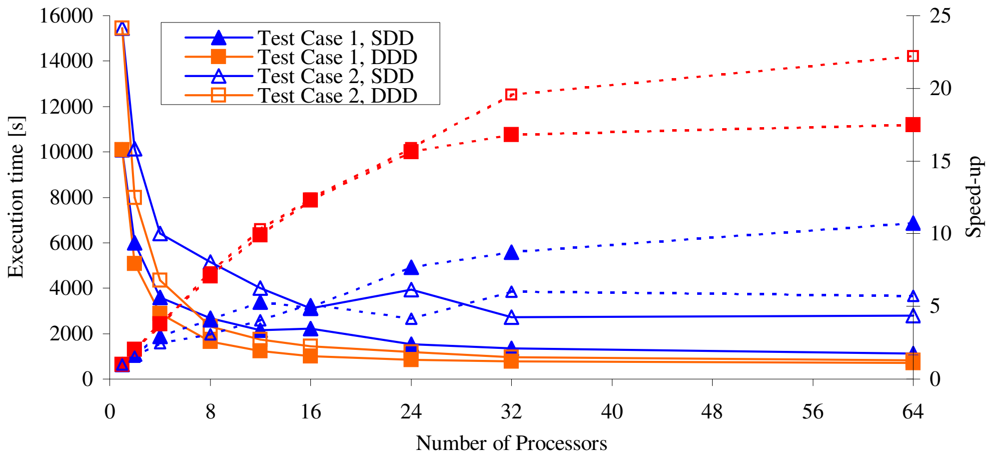

Abbildung 2.4: Ausführungszeit und Speed–up in Abhängigkeit von der Anzahl der Prozessoren; Vergleich der Effizienz der beiden Parallelisierungsverfahren für beide Testfälle.

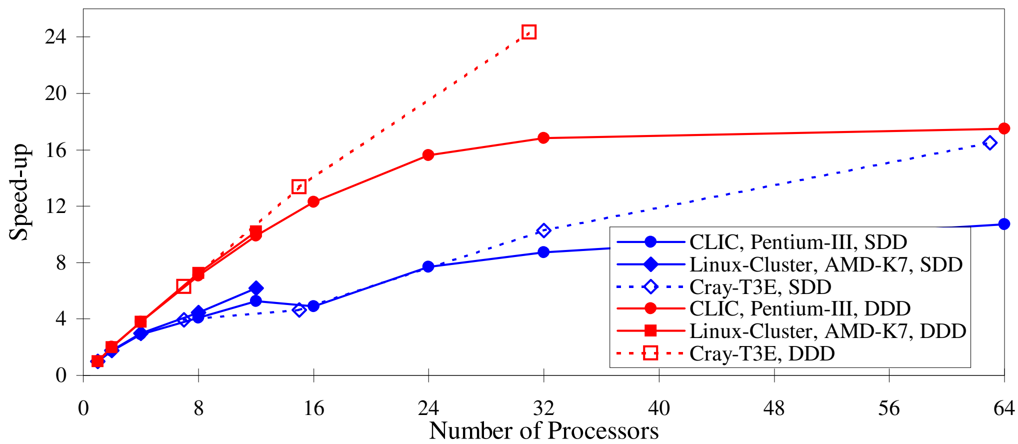

Abbildung 2.5: Vergleich der Parallelisierungsverfahren auf dem Chemnitzer Linux–Cluster (CLIC)und der Cray–T3E der TUD; Speed–up für den Testfall 1.

Abb. 2.4 zeigt die absoluten Ausführungszeiten und die daraus resultierenden Speed–up Ergebnisse für Berechnungen beider Testfälle mit den SDD– und DDD–Verfahren auf einer variierenden Anzahl von Prozessoren. Abb. 2.4 zeigt die deutliche Reduzierung der absoluten Berechnungszeiten, die zunächst mit beiden Parallelisierungsverfahren erreicht werden kann. Aus den Daten für den resultierenden Speed–up der Berechnugnen wird jedoch der klare Vorteil des DDD–Verfahrens gegenüber dem SDD–Verfahren deutlich. Des weiteren geht aus den beiden Kurven für das DDD–Verfahren hervor, daß die Berechnungen mit diesem Verfahren für beide Testfälle annähernd dieselbe absolute Ausführungszeit benötigten, d.h. die Berechnung ist tatsächlich annähernd unabhängig von der Art der Fluid–Partikel–Strömung. Hier weisen die Ergebnisse für das SDD–Verfahren einen deutlichen Unterschied auf.

Abb. 2.5 zeigt die Ergebnisse des direkten Vergleichs zwischen den 3 verschiedenen verwendeten Parallelrechnersystemen für die Berechnungen des 1. Testfalls. Während aus verständlichen Gründen die Ergebnisse für den CLIC und das Athlon–basierende Workstation–Cluster annähernd übereinstimen, ist der Einfluß des Hochgeschwindigkeits–Kommunikationsnetzwerkes der Cray–T3E deutlich zu sehen. So ist ein annähernd linearer Anstieg des Speed–up's für die Testfall–rechnungen auf der Cray bis zu den maximal verwendeten 32 Prozessoren zu beobachten. Auf den Workstation–Clustern mit FastEthernet–Kommunikationsnetzwerk wurden absolut geringere Speed–up–Werte erzielt. Darüber hinaus ist für den Testfall der weitgehend homogenen Fluid–Partikel–Strömung der Übergang in die Stagnation (Sättigung) für Prozessorzahlen größer 32 zu beobachten (Hierbei muß angemerkt werden, daß für die vorliegenden Daten wegen Problemen in der Hardware–Bereitstellung zunächst noch das sogenannte Service–Netzwerk für die Knoten–Kommunikation verwendet werden mußte. Auch der Einsatz von Kanalbündelung der beiden am CLIC vorhandenen FastEthernet–Netzwerke z.B. mittels der SCORE–Libraries kann zu einer weiteren positiven Veränderung dieser Ergebnisse noch beitragen.)

Weitere und detailliertere Untersuchungen, die die Eignung der entwickelten SDD– und DDD–Verfahren für die Berechnung disperser Mehrphasenströmungen auf MIMD– und speziell auf Workstation–Cluster–Systemen dokumentieren, finden sich in den Veröffentlichungen der Forschungsgruppe Mehrphasenströmungen [3.6, 3.8, 3.5, 3.14, 3.15, 3.3].

2.4.3 Teilaufgabe Instationäre Mehrphasenströmungen

Der instationäre Löser wurde implementiert und an einem Rechnungs–Messungsvergleich, der auf einem Laborexperiment [2.3] beruht, erprobt. Des Weiteren ist die Berechnung eines DFG–Benchmarks in Vorbereitung, der ein laminares Vortexshading hinter einem Zylinder zeigt[2.5]. Als Vergleichsgrößen werden die gemittelte Strouhal–Zahl, Lift- und Dragkoeffizient verwendet. Außerdem wurden Profile der gemittelten Fluidgeschwindigkeiten mit LDA vermessen [2.4].

Das Experiment von Fohanno besteht aus einem vertikalen, sich nach unten verengenden, Glaskanal, der mit 3mm Glaskugeln von oben nach unten durchströmt wird. Die Kanaleinschnürung verursacht starke Wand–Partikelinteraktion, die wiederum intensive Partikel–Partikelkollision bewirkt. Die Streichlinien der Partikel werden in einem Lichtschnitt photographisch ermittelt und mit einer nachfolgenden Bildverarbeitung ausgewertet.

Die Ergebnisse der Validierung anhand des Laborexperiments wurden in [3.6] und [3.12] diskutiert. Abbildung 2.6 zeigt die mittleren Partikelgeschwindigkeiten bzw. deren RMS Werte in einer Schnittebene des vertikalen, konvergenten Kanals. Die mittleren Geschwindigkeiten stimmen recht gut überein, während es bei den RMS–Werten Abweichungen in den absoluten Werten gibt. Die Übereinstimmung der Trends der RMS–Werte sind jedoch ausreichend. Die Abweichungen können aus Unterschieden bei der Berechnung der Partikel–Wandkollisionen kommen, die wiederum in Abweichungen von der idealen Kugelform der Glaspartikeln im Experiment zu suchen wären. Um eine ausreichend große Anzahl von Rechenpartikeln für die Mittelungsprozesse zur Verfügung zu haben wurden bis zu 100 000 Partikel simultan berechnet. Skalierungstests haben gezeigt, daß erst bei etwa 700 000 Partikel der Speicher der zehn Athlon Prozessoren mit 512 MB RAM voll ausgeschöpft wird. Die Simulation wurde nach dem DDD–Verfahren (siehe 2.4.2) auf zehn AMD 600 MHz und Fast–Ethernet–Netzwerk (100Mbit/s) unter Berücksichtigung von Phasenkoppelung, Wand- und Partikel–Partikelkollision durchgeführt.

Es zeigt sich, daß die Recheneffizienz im instationären Modus recht unterschiedlich zum stationären Modus ist. Die Kriterien, die die parallele Effiziens im Wesentlichen beeinflussen sind die Netzwerkbandbreite und der Rechenaufwand pro Partikel. Im aktuellen Fall — kalte Strömung kein Phasen- oder Wärmeübergang, sehr geringe Relativgeschwindigkeit zwischen Gas und Partikeln (Stokes–Zahl$\gg 1$) — ist der Rechenaufwand ein Minimum und damit für die Effizienz der ungünstigste Fall. Es wurde für das DDD–Verfahren eine parallele Effizienz von 10% erreicht, was sich bei großen Relativgeschwindigkeiten und kleiner Stokes–Zahl auf 60-70% erhöhte. Die Lastaufteilung ist ab einer Mindestlast von 2500 Partikeln bei minimaler Rechenlast pro Partikel als optimal zu bezeichnen.

Literaturverzeichnis

[**Dissertationen**]

[3.1] E. Wassen: *Entwicklung paralleler Algorithmen zur numerischen Simulation von Gas-Partikel-Strömungen unter Berücksichtigung von Partikel-Partikel-Kollisionen*, Diss., TU Chemnitz, 1998.

[**Begutachtete Publikationen**]

[3.2] Frank, Th., Bernert, K., Schneider, J.: *Numerische Untersuchungen der Gas–Partikel–Strömung in symmetrischen Doppelzyklon–Abscheidern*, VDI Berichte, Nr. 1511, 1999.

[3.3] Wassen E., Frank Th.: *Simulation of cluster formation in gas-solid flow induced by particle-particle collisions, Chemnitz University of Technology*, Chemnitz, Germany, March 1999, pp. 1-20. Published in International Journal of Multiphase Flow (March 2001) Volume 27, Issue 3, pages 437-458.

[3.4] Frank, Th., Schneider, J., Yu, Q. Wassen, E.: *Experimental and Numerical Investigation of Particle Separation in a Symmetrical Double Cyclone Separator*, 8th Int. Symposium on Gas–Particle Flows, ASME Fluids Engineering Division Summer Meeting, San Francisco, CA, U.S.A., July 18–22, 1999, CD–ROM Proceedings, Paper No. FEDSM99–7865, pp. 1–10.

[3.4] Frank Th., Schneider J., Bernert K.: *Numerische Untersuchung der Gas–Partikel–Strömung in symmetrischen Doppelzyklonabscheidern.* VDI–Gesellschaft Energietechnik und VDI–Gesellschaft Verfahrenstechnik und Chemieingenieurwesen, VDI–

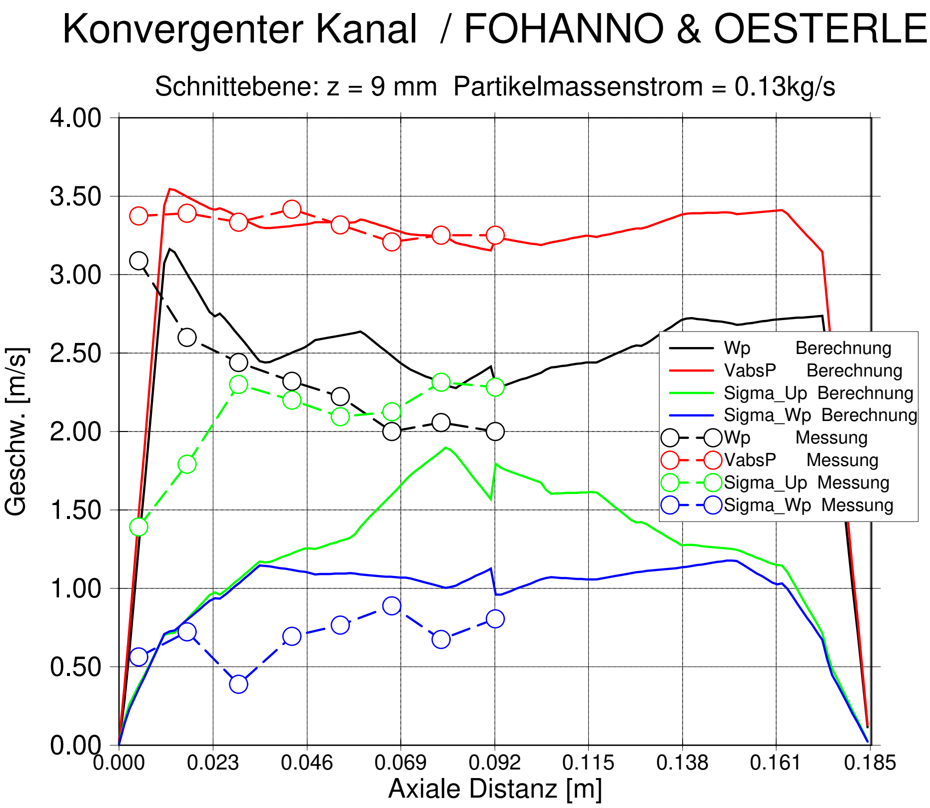

Abbildung 2.6: Vergleich Rechnung Messung mit dem instationären Löser.

Forum "Zyklonabscheider in der Energie– und Verfahrenstechnik", 27./28. Oktober 1999, Leverkusen, Germany.

[3.5] Frank Th.: *Application of Eulerian–Lagrangian Prediction of Gas–Particle Flows to Cyclone Separators*, VKI, Von Karman Institute for Fluid Dynamics, Lecture Series Programme 1999–2000, "Theoretical and Experimental Modeling of Particulate Flow", Bruessels, Belgium, 03.–07. April 2000.

[3.6] Pachler K., Bernert K., Frank Th., Schneider H.: *Simulation von Mehrphasenströmungen auf Parallelrechnern und deren Anwendung in der Verfahrenstechnik*, CHC-2000, 2. Chemnitz / Hamburger Colloqium, Auslegung und Modellierung energetischer Anlagen und Systeme, Technische Universität Hamburg-Harburg, Deutschland, 16./17. November 2000.

[3.7] Frank, Th., Bernert, K., Schneider, H., Pachler, K.: *Efficient Parallelization of Eulerian–Lagrangian Approach for Disperse Multiphase Flow Calculation on MIMD Computer Architectures,* IEEE International Conference on Cluster Computing — CLUSTER 2000, November 28.–December 2., 2000, Chemnitz, Germany.

[3.8] Bernert, K., Frank, Th., Schneider, H., Pachler, K.: *Multi–Grid Acceleration of a SIMPLE-Based CFD-Code and Aspects of Parallelization*, IEEE International Conference on Cluster Computing – CLUSTER 2000, November 28.–December 2., 2000, Chemnitz, Germany.

[**Sonstige Publikationen**]

[3.9] Frank Th.: *Simulation von Strömungen in der Verfahrenstechnik auf Clustercomputern*, Kolloqium anläßlich der Einweihung des Chemnitzer Linux-Clusters (CLIC) TU Chemnitz, Germany, 11. Oktober 2000.

[3.10] Frank Th., Wassen E., Bernert K., Yu Q.: *Numerische Berechnung der Partikelbewegung und -abscheidung in Zyklonen*, Internationales Innovationsseminar 2000 zu

Magdeburg INNSA - Innovation Relay Centre, tti Magdeburg Berechnung Verfahrenstechnischer Prozesse / Computational Process Engineering Magdeburg, 26.-27. Oktober 2000.

[3.11] Meyer A., Frank Th.: *Parallel Computing in Chemnitz und das Beispiel einer Anwendung bei Mehrphasenströmungen*, WIR - Sachsen, 1. Kolloquium Wissenschaftliches Rechnen in Sachsen ZHR der TU Dresden, 4. Dezember 2000.

[3.12] Pachler K., Bernert K., Frank Th.: *Simulation einer instationären Gas–Partikel–Strömung unter Berücksichtigung von Kollision, Wandstoß und Phasenswechselwirkung*, Sitzung der GVC–Fachausschüsse "Mehrphasenströmungen" und "Mischvorgänge", 1.–2. März 2001, Wernigerode, Germany.

[3.13] Schneider H., Frank Th., Schade K.-P., Bernert K., Hädrich Th., Erdmann H.-J.: *Numerical and Experimental Study of the Gas-Particle Flow in Pipework and Flow Splitting Devices of Coal-Fired Power Plant*, PARTEC 2001 - International Congress for Particle Technology Session 12, Multiphase Flow/CFD, Nürnberg, Germany, 27-29 March 2001.

[**Eingereichte Publickationen**]

[3.14] Bernert, K., Frank, Th., Schneider, H., Pachler, K.: *Numerical Simulation of Disperse Multiphase Flows with an Application in Power Engineering*, 3rd International Symposium on Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications, July 22-26, 2001, Atlanta, Georgia, USA; to appear in ASME Conference Proceedings.

[3.15] Frank Th., Bernert K., Pachler K., Schneider H.: *Aspects of Efficient Parallelization of Disperse Gas-Particle Flow Predictions using Eulerian-Lagrangian Approach*, ICMF 2001 - 4th International Conference on Multiphase Flow, Paper No. 311, pp. 1-13, New Orleans, Louisiana, U.S.A., May 27 - June 1, 2001.

[3.16] Schade K.-P., Schneider H., Frank Th., Erdmann H.-J., Hädrich Th., Bernert K.: *Experimental and Numerical Investigation of Particle Erosion Caused by Pulverized Fuel in Channels and Pipework of Coal-Fired Power Plant*, ICMF 2001 - 4th International Conference on Multiphase Flow, Paper No. 325, pp. 1-12, New Orleans, Louisiana, U.S.A., May 27 - June 1, 2001.

2.5 Offene Fragen/Ausblick

Im Berichtszeitraum sind im Teilprojekt D2 eine Reihe substantieller Fortschritte für die parallele numerische Simulation disperser Mehrphasenströmungen erzielt worden. Im Ergebnis können mit dem erarbeiteten Euler–Lagrange–Verfahren sowohl stationäre als auch instationäre Aufgabenstellungen aus der Praxis mit guter paralleler Effizienz auf Rechnerplattformen mit MIMD-Architektur bearbeitet werden. Mit den erreichten Fortschritten in der Bearbeitung des Forschungsvorhabens ergeben sich die nachfolgenden Fragestellungen für weitergehende Untersuchungen.

1. Ein erster Schwerpunkt der geplanten Arbeit ist die Zusammenführung der bisher überwiegend getrennt behandelten Algorithmen zur Berechnung der kontinuierlichen und der dispersen Phase. Ziel ist eine enge iterative Kopplung der beiden Verfahren einschließlich der erforderlichen Konvergenzkontrolle, um eine effiziente Berechnung komplexer stationärer Strömungsvorgänge disperser Mehrphasenströmungen mit starken Phasenwechselwirkungen (2–Wege–Kopplung auf Grund von Masse-, Impuls- und Wärmeübertragung zwischen den beteiligten Phasen) untersuchen zu können.

2. Eine notwendige Voraussetzung hierfür sind Verbesserungen des für die Berechnung der kontinuierlichen Phase implementierten Mehrgitterverfahrens bezüglich seiner Robustheit und der Kontrolle der Steuerparameter, um die Konvergenz des Verfahrens bei Berücksichtigung der stark schwankenden Phasenwechselwirkungsterme zu gewährleisten.

3. Einen weiteren Schwerpunkt bildet die Fortführung der begonnenen Arbeiten zur Berechnung instationärer Mehrphasenströmungen auf Parallelrechnern. Hier sind weitere Entwicklungen bezüglich der Phasenwechselwirkungen (Turbulenz, Masse–, Impuls– und Wärmeübertragung) und der Übergang zu einem Diskretisierungsschema mit 2. Genauigkeitsordnung in der Zeit notwendig.

4. Begleitende aber dennoch außerordentlich notwendige Arbeiten betreffen die weitere Optimierung der eingesetzten Parallelisierungstechniken und die Beseitigung serieller „Flaschenhälse“ in den Pre- und Postprozessorschritten.

   Ein absehbarer Effektivitätsgewin liegt u.a. in der Beschleunigung der für sehr große Partikelkollektive und für instationäre Simulationen sehr zeitintensiven initialen Partikellokalisierung auf großen blocktrukturierten Gitternetzen. Weitere Verbesserungen sind in der Handhabung der für die Lösung von Aufgabenstellungen aus der Praxis notwendigen sehr großen Datenmengen erforderlich. Das betrifft insbesondere die Simulation von hochtransienten Vorgängen (Datenaustausch zwischen den Teilen des Lösungsverfahrens, Auswertung) sowie generell die Daten–Ein- und -Ausgabe. Hier ist eine Parallelisierung erforderlich, weil die für die Ein- und Ausgabe benötigte Zeit für große Problemstellungen einen nicht mehr zu vernachlässigenden Teil der Nutzungsdauer eines MIMD–Rechnersystems einnimmt.

5. Ein letzter Schwerpunkt liegt in der Behandlung einer Reihe von komplexen und realitätsnahen Problemstellungen mit dem Ziel, die Eignung und Robustheit der eingesetzten Lösungsverfahren nachzuweisen und Rückschlüsse auf notwendige Weiterentwicklungen der Parallelrechnerarchitekturen, Kommunikationsstrukturen und Solvertechniken zu ermöglichen. In diesem Zusammenhang sollen in Zusammenarbeit mit Prof. Rehm (B9) auf Cluster–Architekturen mit Hochgeschwindigkeits–Kommunikationsnetzwerken unterschiedlicher Art und Struktur typische Anwendungsfälle bezüglich ihres Kommunikationsverhaltens vergleichend untersucht werden.