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
Prof. Dr.–Ing. Heinz Herwig	Fakultät für Maschinenbau und Verfahrenstechnik
	Professur Technische Thermodynamik
	Forschungsgruppe Mehrphasenströmungen

2.1.2 Projektbearbeiter

Dipl.–Ing. Erik Wassen	Professur Technische Thermodynamik
Dr.–Ing. Qiang Yu	Forschungsgruppe Mehrphasenströmungen

2.2 Ausgangsfragestellung/Einleitung

Die letzten zwei Jahrzehnte haben eine enorme Entwicklung im Bereich der numerischen Strömungsmechanik hervorgebracht. Zum einen haben sich Rechengeschwindigkeit und Speicherplatzangebot der Rechner um mehrere Größenordnungen erhöht. Andererseits wurde diese Entwicklung begleitet durch eine beständige Fortentwicklung der eingesetzten numerischen Simulationsverfahren in Bezug auf deren Berechnungsgenauigkeit, Effizienz und der zunehmenden Berücksichtigung komplexer physikalischer Wechselwirkungen. Die Weiterentwicklung der numerischen Methoden von der Berechnung vereinfachter, grenzschichtartiger Strömungen bis hin zur direkten numerischen Simulation (DNS) turbulenter Strömungen war begleitet von einer Steigerung der Anforderungen an die verwendete Hardware.

Heute gehören Verfahren zur numerischen Simulation von phasengekoppelten, dispersen Mehrphasenströmungen unter Berücksichtigung von Stoff–, Impuls– und Wärmeübertragung zwischen den Phasen mit 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, und den absehbaren Grenzen für die weitere Steigerung deren Leistung, kam es in den letzten Jahren verstärkt zur Entwicklung von massiv parallelen Rechnerarchitekturen. Diese Multiprozessorsysteme mit verteiltem Speicher (MIMD–Architekturen) erreichen durch den parallelen Einsatz sehr vieler, relativ einfacher Prozessoren eine skalierbare Rechenleistung und Speicherkapazität, die die modernster Supercomputer zu übertreffen vermag. Die Nutzung dieser hohen Rechenleistung für Aufgabenstellungen der numerischen Strömungsmechanik ist jedoch nicht unproblematisch. Um die potentiell sehr hohe Rechenleistung einer solchen Multiprozessorarchitektur in der Praxis bei der Berechnung strömungsmechanischer Problemstellungen auch wirklich zu erreichen, ist eine konzeptionelle Überarbeitung bestehender Lösungsalgorithmen im Hinblick auf deren mögliche parallele Abarbeitung notwendig. Eine hohe parallele Effizienz kann jedoch nur unter Anwendung speziell auf die parallele Hardwarearchitektur ausgerichteter Parallelisierungskonzepte erreicht werden.

Für Finite–Volumen–Methoden (FVM) zur Berechnung von Einphasenströmungen auf blockweise strukturierten Gitternetzen hat sich in den letzten Jahren die Methode der Gebietszerlegung als eine einfach zu realisierende und effizient anwendbare Parallelisierungsmethode etablieren können. Hierbei wird zunächst für eine vorgegebene Strömungsgeometrie mit einer bestimmten Anzahl von Gitterblöcken ein optimales Gitternetz erzeugt. Durch weitere Zerlegung der Gitterblöcke wird deren Anzahl der Anzahl der Prozessoren angepaßt. Die Kopplung der zerlegten Gebiete wird genauso behandelt wie die Kopplung von geometrisch bedingten Blöcken. Andere Parallelisierungsverfahren, wie z.B. die Methode der Zeitparallelisierung für instationäre Strömungen und die Kombinationsmethode/Methode dünner Gitter wurden für die numerische Simulation von Einphasenströmungen auf Parallelrechnern entwickelt.

Demgegenüber waren für die Berechnung von dispersen Mehrphasenströmungen auf massiv parallelen Rechnerarchitekturen bis vor kurzer Zeit noch keine Verfahren entwickelt bzw. veröffentlicht. Soweit zur Modellierung der dispersen Phase das Euler–Verfahren (Zwei–Fluid–Modell) herangezogen wird, so sind die für Einphasenströmungen entwickelten Parallelisierungsverfahren unmittelbar auf die daraus resultierenden Lösungsverfahren übertragbar. Für das bei dispersen Mehrphasenströmungen häufig angewandte, und aus physikalischer Sicht vorteilhaftere Lagrange–Verfahren fehlten jedoch bisher geeignete Ansätze zu deren 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 [Tysinger97, Tsuji96, Yone93].

Für das Forschungsvorhaben in D2 waren folgende Aufgaben vorgesehen :

1. Die Entwicklung geeigneter Datenstrukturen zur Beschreibung komplexer, 3–dimensionaler Strömungsfelder unter Verwendung von blockstrukturierten Gitternetzen auf MIMD–Rechnerarchitekturen.

2. Die Entwicklung eines Lagrange–/PSI–Cell–Verfahrens zur Berechnung disperser Mehrphasenströmungen auf der Grundlage der zur Verfügung stehenden Finite–Volumen–Verfahren auf blockstrukturierten 2– und 3–dimensionalen Gitternetzen.

3. Die Entwicklung effizienter, paralleler Lösungsalgorithmen für die zeitlich gemittelten Navier–Stokes– und Skalartransportgleichungen sowie für die Bewegungsgleichungen der dispersen Phase unter Berücksichtigung der Phasenwechselwirkungen.

4. Die Integration von Verfahren zur automatischen Lastverteilung über die Prozessoren eines Parallelrechnersystems.

Diese Aufgaben hatten zum Ziel, ein effizientes und flexibles Berechnungsverfahren für Strömungen in komplexen Geometrien zu erarbeiten, mit dem die rechenintensiven Simulationen phasengekoppelter, disperser Mehrphasenströmungen auf einer breiten Palette von High–Performance–Computing (HPC)–Plattformen effektiv durchgeführt werden können. Der im Berichtszeitraum erreichte Bearbeitungsstand ist in der nachfolgenden Tabelle übersichtsartig zusammengefaßt.

Übersicht über den Bearbeitungsstand des Forschungsvorhabens D2 für den Bearbeitungszeitraum 1996–1998

	2–d			3–d		
	Navier–Stokes Löser	Lagrange–Verfahren	parallelisiertes Verfahren	Navier–Stokes Löser	Lagrange–Verfahren	parallelisiertes Verfahren
Preprozessing :						
Skript–orientiert	×	–	•	×	–	–
Schnittstelle zu CFX von AEA	–	–	–	•	–	•
Gittereigenschaften :						
blockstrukturiert	×	×	•	×	•	•
multilevel	×	×	•	□	□	□
Adaptivität	–	–	–	□	□	□
laminar	×	×	•	×	•	•
turbulent	×	×	•	•	•	•
stationär	×	×	•	×	•	•
instationär	×	•	•	□	□	□
Gleichungslöser (lin. GLS) :						
SIP–Solver nach Stone	×	–	•	×	–	•
CG + hierarch. DD–Vorkonditionierer	–	–	–	□	–	□
Parallelisierungsverfahren :						
Domain Decomposition	–	–	•	–	–	•
DSM Algorithmus	–	–	•	–	–	•
erweiterter DSM Algorithmus	–	–	□	–	–	□
Multi–Threading + Parallel I/O	–	–	–	–	–	□
Fluid → Partikel Einwegkopplung	×	×	•	•	•	•
Fluid ↔ Partikel Wechselwirkung :						
Turbulenz	×	×	•	•	•	•
Stoffübertragung	□	□	□	□	□	□
Impulsübertragung	×	×	•	•	•	•
Wärmeübertragung	□	□	□	□	□	□
Partikel–Wand Stoß	–	×	•	–	•	•
Partikel–Partikel Stoß	–	•	•	–	□	□
Postprozessing :						
Schnittstelle zu Tecplot	×	×	•	•	•	•
Schnittstelle zu Ensight	•	•	•	•	•	•

(–) nicht zutreffend
(×) Bearbeitungsstand vor Beginn des SFB 393
(•) Ergebnisse des SFB–Teilprojekts D2
(□) beabsichtigte Implementationen und Weiterentwicklungen

Die folgenden Kapitel zeigen die Vorgehensweise, beschreiben die verschiedenen entwickelten Parallelisierungsverfahren und die für eine Reihe von repräsentativen

Testfällen auf verschiedenen Parallelrechnerarchitekturen erzielten Ergebnisse im Detail. Des weiteren sei an dieser Stelle auch auf die in 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 Das Euler–Lagrange–Verfahren

Die Bewegung der kontinuierlichen Phase

a) Grundgleichungen

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

$$\frac{\partial}{\partial x}(\rho_F\, u_F\, \Phi) + \frac{\partial}{\partial y}(\rho_F\, v_F \Phi) \quad + \quad \frac{\partial}{\partial z}(\rho_F\, w_F \Phi) \quad =$$

$$\frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial z}\right) \quad + \quad S_\Phi + S_\Phi^P \tag{2.1}$$

Hierbei ist ρ_F die Dichte des Fluids, u_F, v_F und w_F sind die Fluidgeschwindigkeiten in x-, y- bzw. z–Richtung, Φ ist eine allgemeine Variable, Γ_Φ ein allgemeiner Transportkoeffizient und S_Φ ein allgemeiner Quellterm. Die Größe S_Φ^P ist ebenfalls ein Quellterm, der die Wechselwirkung (Stoff–, Impuls–, Wärmeübertragung) zwischen disperser und fluider Phase beschreibt.

Der Zusammenhang zwischen Φ, S_Φ, S_Φ^P und Γ_Φ sowie die verwendeten Konstanten sind in Tabelle 2.1 angegeben. In dieser Tabelle ist μ die laminare und μ_t die turbulente Viskosität, p ist der Druck und f_x, f_y und f_z sind die kartesischen Komponenten der pro Masseneinheit einwirkenden äußeren Kräfte.

b) Numerische Lösung

Das Strömungsgebiet wird durch ein block–strukturiertes, nicht–orthogonales, randangepaßtes numerisches Gitter beschrieben. Die Transportgleichungen werden mit Hilfe einer Finite–Volumen–Methode diskretisiert. Zur Lösung der diskretisierten und linearisierten Gleichungen wird die SIP-Methode nach [Stone68] verwendet. Die Druck–Geschwindigkeits–Kopplung wird realisiert durch die Anwendung eines SIMPLE-Algorithmus [Patankar80, Perić92].

Das Lösungsverfahren für die fluide Phase ist in den Programmpaketen FAN-2D (für zweidimensionale Strömungen) bzw. FAN–3D (für dreidimensionale Strömungen) implementiert, die von Perić und Lilek entwickelt [Perić92, Perić93] und in D2 parallelisiert sowie für die Berechnung disperser Mehrphasenströmungen modifiziert [Frank94, Frank96b, Frank97c] wurden.

Φ	S_Φ	S_Φ^P	Γ
1	0	0	0
u_F	$\frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial u_F}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial v_F}{\partial x}\right) + \frac{\partial}{\partial z}\left(\Gamma_\Phi \frac{\partial w_F}{\partial x}\right) - \frac{\partial p}{\partial x} + \rho_F f_x$	$S_{u_F}^P$	μ_{eff}
v_F	$\frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial u_F}{\partial y}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial v_F}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma_\Phi \frac{\partial w_F}{\partial y}\right) - \frac{\partial p}{\partial y} + \rho_F f_y$	$S_{v_F}^P$	μ_{eff}
w_F	$\frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial u_F}{\partial z}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial v_F}{\partial z}\right) + \frac{\partial}{\partial z}\left(\Gamma_\Phi \frac{\partial w_F}{\partial z}\right) - \frac{\partial p}{\partial z} + \rho_F f_z$	$S_{w_F}^P$	μ_{eff}
k	$P_k - \rho_F\,\varepsilon$	0	$\frac{\mu_t}{\sigma_k}$
ε	$\frac{\varepsilon}{k}(c_{\varepsilon_1} P_k - c_{\varepsilon_2}\,\rho_F\,\varepsilon)$	0	$\frac{\mu_t}{\sigma_\varepsilon}$

$$P_k = \mu_t \left\{ 2 \cdot \left[\left(\frac{\partial u_F}{\partial x}\right)^2 + \left(\frac{\partial v_F}{\partial y}\right)^2 + \left(\frac{\partial w_F}{\partial z}\right)^2 \right] + \left(\frac{\partial u_F}{\partial y} + \frac{\partial v_F}{\partial x}\right)^2 + \left(\frac{\partial u_F}{\partial z} + \frac{\partial w_F}{\partial x}\right)^2 + \left(\frac{\partial w_F}{\partial y} + \frac{\partial v_F}{\partial z}\right)^2 \right\}$$

$$\mu_{eff} = \mu + \mu_t \quad , \quad \mu_t = \rho_F c_\mu \frac{k^2}{\varepsilon}$$

$$c_\mu = 0.09 \quad , \quad c_{\varepsilon_1} = 1.44 \quad , \quad c_{\varepsilon_2} = 1.92 \quad , \quad \sigma_k = 1.0 \quad , \quad \sigma_\varepsilon = 1.3$$

Tabelle 2.1: Quellterme und Transportkoeffizienten für verschiedene Variablen Φ

Die Bewegung der dispersen Phase

Zur Modellierung der Bewegung der dispersen Phase wird die Lagrange-Methode 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 Scheranträmung des Teilchens (Saffman-Kraft), die Gravitationskraft und der hydrostatische Auftrieb. Die translatorischen Bewegungsgleichungen für ein Partikel lauten damit:

$$\frac{d}{dt}\vec{v}_P = \frac{3}{4}\frac{\rho_F}{\rho_P d_P}\Bigg(v_{rel} C_D(Re_P)\,[\vec{v}_F - \vec{v}_P]$$

$$+ \frac{v_{rel}}{\omega_{rel}} C_M(\sigma) \cdot \left[(\vec{v}_F - \vec{v}_P) \times (\vec{\omega} - \vec{\Omega}) \right]$$

$$+ \quad \frac{2\nu_F^{1/2}}{\pi|\vec{\Omega}|^{1/2}} C_A \left[(\vec{v}_F - \vec{v}_P) \times \vec{\Omega}\right]\Bigg) + \frac{\rho_P - \rho_F}{\rho_P}\,\vec{g}\,, \qquad (2.2)$$

mit

$$\vec{\Omega} = \mathrm{rot}\ \vec{v}_F \quad , \quad Re_P = \frac{d_P\, v_{rel}}{\nu_F} \quad , \quad Re_\omega = \frac{1}{4}\frac{{d_P}^2 \omega_{rel}}{\nu_F} \quad , \quad \sigma = \frac{1}{2}\frac{d_P \omega_{rel}}{v_{rel}}\,,$$

$$v_{rel} = |\vec{v}_F - \vec{v}_P| \quad , \quad \omega_{rel} = |\vec{\omega} - \vec{\Omega}|\,.$$

Die Gleichung für die Rotationsbewegung des Partikels lautet:

$$\frac{d}{dt}\,\vec{\omega} = -\frac{15}{16\pi}\frac{\rho_F}{\rho_P}\omega_{rel}\xi_m(Re_\omega)\left[\vec{\omega} - \vec{\Omega}\right]\,. \qquad (2.3)$$

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 [Frank92].

Zur Modellierung der Partikel–Wand–Kollisionen wurde hier ein Wandrauhigkeitsmodell nach [Sommerfeld92] verwendet. Hierbei stößt das Partikel mit einer um einen Winkel γ angestellten virtuellen Wand. Der Inklinationswinkel γ ist dabei eine Gauß–verteilte Zufallsgröße, deren Mittelwert 0° ist und deren Standardabweichung von den Rauhigkeitsparametern der Wand sowie vom Partikeldurchmesser abhängt. Partikel–Partikel–Kollisionen wurden in dieser Arbeit nicht berücksichtigt. Zur Modellierung des Einflusses der Fluidturbulenz auf die Partikelbewegung wurde hier das sogenannte Lagrangian-Stochastic-Deterministic (LSD) Tubulenzmodell verwendet [Schönung87]. 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 [Crowe77] berücksichtigt.

2.3.2 Teilaufgabe Das Parallelisierungsverfahren für den Navier-Stokes-Löser

Die Parallelisierung der Fluidberechnung basiert auf der weit verbreiteten Methode der Domain Decomposition. Das numerische Gitter wird hierzu in Partitionen aufgeteilt (siehe Abb. 2.1). Die Strömung in jedem dieser Teilgebiete wird von einem Prozessor des Parallelrechners berechnet. Bei dem hier verwendeten Rechenverfahren für die fluide Phase wird für jede Zelle des numerischen Gitters derselbe Rechenaufwand benötigt. Aus diesem Grund ist der Rechenaufwand, der in einer Gitterpartition entsteht, proportional zur Anzahl der Gitterzellen in dieser Partition. Um eine gleichmäßige Verteilung der Rechenlast unter den Prozessoren zu erreichen, sollten daher die Teilgitter gleich groß sein, d.h. die gleiche Anzahl Gitterzellen besitzen.

Die Strömungsgrößen im Inneren der Partitionen können wie im seriellen Fall berechnet werden. An den physikalischen Rändern ergeben sich die Werte der Variablen aus den physikalischen Randbedingungen. An den Partitionsrändern jedoch ist

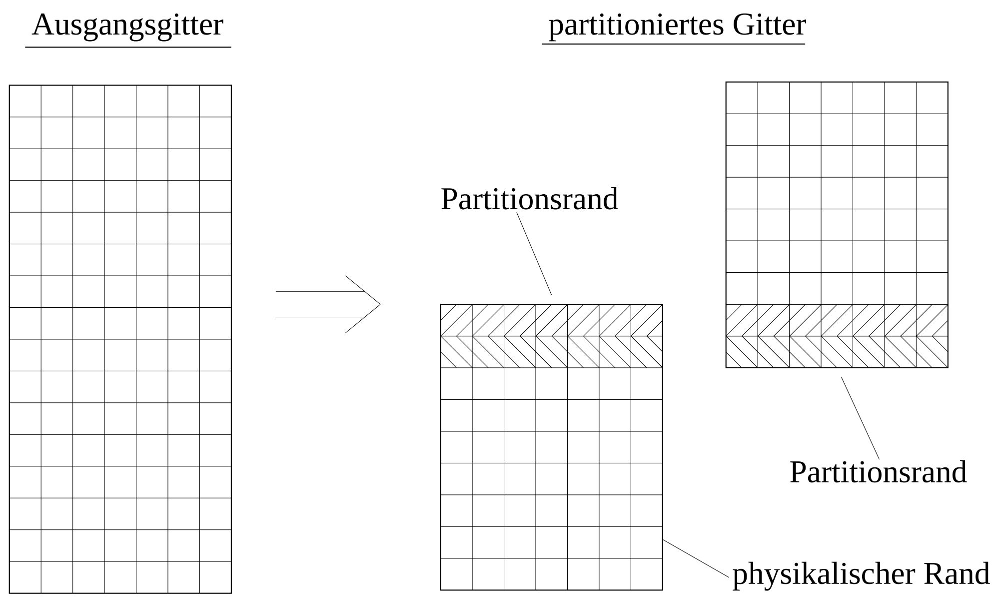

Abbildung 2.1: Prinzip der Gitterpartitionierung

eine unabhängige Berechnung nicht möglich. Hier entstehen Datenabhängigkeiten, die einen Datenaustausch zwischen den Partitionen und somit eine Kommunikation zwischen den Prozessoren erforderlich machen. Zu diesem Zweck wird die Partitionierung so durchgeführt, daß sich die Teilgitter überlappen, wie in Abb. 2.1 gezeigt. Dadurch können die Strömungsgrößen für die umittelbar am Partitionsrand gelegenen Zellen von der Nachbarpartition übertragen werden, wo sie zuvor berechnet wurden.

Durch die notwendige Kommunikation zwischen den Prozessoren entsteht ein zusätzlicher Zeitaufwand, der die Effizienz der Parallelisierung reduziert. Um diesen Zusatzaufwand zu minimieren, muß die Rechenlast für alle Prozessoren möglichst gleich groß sein, damit Wartezeiten bei der Kommunikation vermieden werden. Außerdem müssen die durch die Gitteraufteilung entstehenden Partitionsränder möglichst klein sein, um die Menge der auszutauschenden Daten zu minimieren.

2.3.3 Teilaufgabe Parallelisierung des Lagrange–Lösers mit Domain Decomposition

Um das Lagrange-Verfahren auch auf massiv parallelen Rechnern mit ihrer beschränkten Speichergröße verwenden zu können, wurde zunächst das Prinzip des Domain Decomposition Verfahrens auch auf das Lagrange–Berechnungsverfahren angewandt. Wie in Abb. 2.2 dargestellt, existieren zwei Arten von Prozessen: Ein Host-Prozeß und eine Anzahl von Node-Prozessen. Jeder Node-Prozeß speichert lokal und permanent die Gittergeometrie- und Fluiddaten eines Gitterblocks. Die Anzahl der Node-Prozesse ist deshalb gleich der Anzahl der Blöcke des numerischen Gitters. Die Trajektorienberechnung wird von den Node-Prozessen durchgeführt, wobei ein Node alle Trajektoriensegmente berechnet, die den ihm zugeteilten Gitterblock durchqueren.

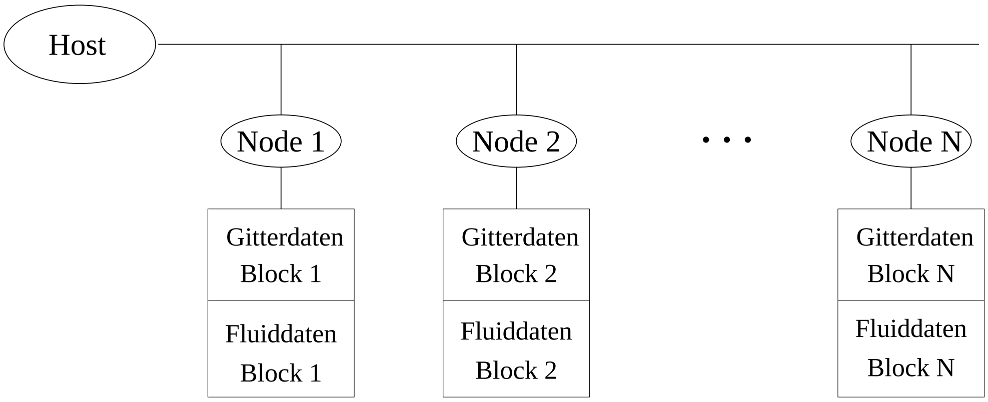

Abbildung 2.2: Prinzip des Domain Decomposition Verfahrens

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 eines 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.

Das DD-Verfahren ist einfach zu implementieren, da es dieselbe Verteilung von Gitter- und Fluiddaten unter den Prozessoren verwendet wie der parallelisierte Navier-Stokes-Löser. Ein Nachteil des DD-Verfahrens ist, daß eine ungünstige Verteilung der Rechenlast auftreten kann. Eine ungleichmäßige Lastverteilung kann unterschiedliche Ursachen haben:

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. 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. - 5. gelten speziell für die DD-Methode für das Lagrange-Verfahren. 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.

Die in Kapitel 2.3.3 beschriebenen Probleme der Rechenlastverteilung können behoben werden, indem die Zuordnung von Gitterpartitionen zu Node-Prozessen nicht statisch, sondern dynamisch erfolgt. Dies ist das Prinzip des Distributed Shared Memory Verfahrens (Abb. 2.3). Im Vergleich zu dem oben beschriebenen Verfahren wird hier eine zusätzliche Klasse von Prozessen eingeführt, die sogenannten Memory-Manager-Prozesse. Diese Prozesse übernehmen die Aufgabe, die Gitter- und Fluiddaten zu speichern. Jeder Memory-Manager-Prozeß speichert während der gesamten Rechnung die Daten eines Gitterblocks, führt aber selber keine Trajektorienberechnungen durch. Die eigentliche Berechnung wird wiederum in den Node-Prozessen durchgeführt. Im Unterschied zum DD-Verfahren kann ein Node-Prozeß Partikeltrajektorien in jeder beliebigen Partition berechnen. Die jeweils aktuell benötigten Gitter- und Fluiddaten werden vom Memory-Manager bereitgestellt, der diese gespeichert hat. Wie beim DD–Verfahren hat der Host-Prozeß die Aufgabe, die Partikelstartbedingungen zu verwalten und an die Nodes zu verteilen.

Im folgenden seien die Aufgaben der einzelnen Prozesse näher beschrieben:

<u>Host:</u>
Der Host verteilt die Partikelstartbedingungen an die Nodes. Dabei ist eine Startbedingung nicht nur der Partikelstatus zu Beginn der Rechung, sondern auch der Status beim Eintritt eines Partikels in einen anderen Gitterblock. Die Startbedingung wird an einen Node gesendet, und nach der Berechnung des Trajektoriensegmentes wird der Endstatus des Partikels von diesem zurückgesendet. Hat das Partikel eine Partitionsgrenze erreicht, wird der Endstatus als neue Startbedingung vom Host an den nächsten Node zur weiteren Berechnung gesendet. Dies wird solange wiederholt, bis alle Trajektorien Abbruchbedingungen erfüllt haben (z.B. Erreichen eines Ausströmrandes). Die optimale Zuteilung von Startbedingungen an die Node-Prozesse ist abhängig von der Anzahl der für eine bestimmte Partition aktuell vorliegenden Startbedingungen. Außerdem wird eine Startbedingung bevorzugt an einen Node gesendet, in dessen Speicher bereits die Daten für die Partition vorliegen, in der diese Startbedingung lokalisiert ist. Somit müssen die Daten nicht erst vom Memory-Manager übertragen werden und der Kommunikationsaufwand wird verringert.

<u>Node:</u>
Der Node empfängt jeweils eine Partikelstartbedingung vom Host, berechnet das

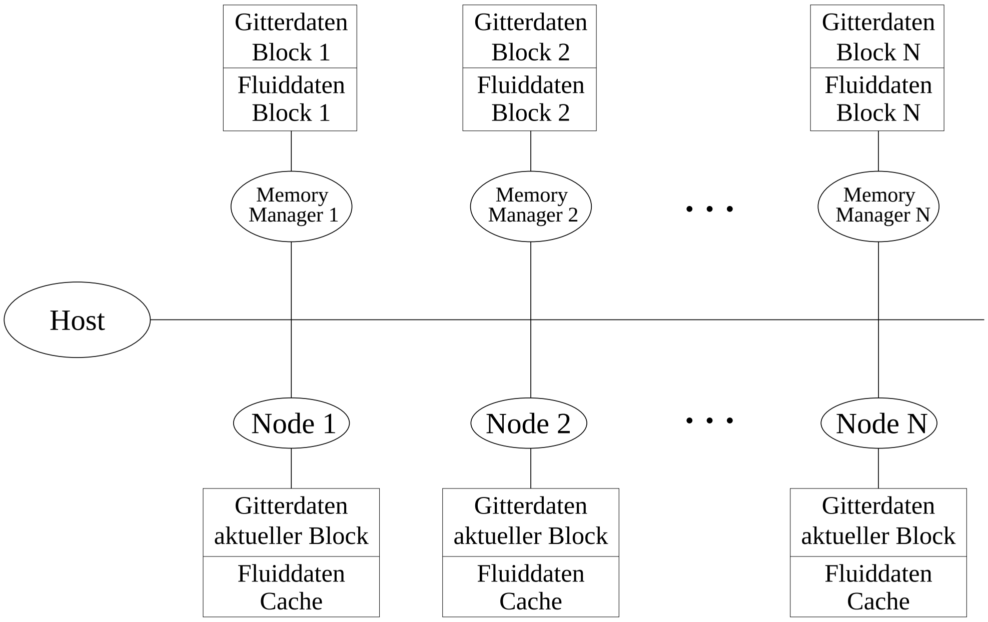

Abbildung 2.3: Prinzip des Distributed Shared Memory Verfahrens

zugehörige Trajektoriensegment und sendet den Endstatus des Partikels zurück an den Host. Nach dem Empfang der Startbedingung wird geprüft, ob der zugehörige Partikelort in dem Gitterblock liegt, dessen Gitter- und Fluiddaten aktuell im lokalen Speicher vorliegen. Ist dies der Fall, so kann sofort mit der Trajektorienberechnung begonnen werden. Im anderen Fall müssen diese Daten zunächst vom entsprechenden Memory-Manager zum Node übertragen werden.
Zur Verfolgung eines Partikels im Strömungsgebiet und zur Detektierung der Annäherung an Blockgrenzen müssen dem Node die kompletten Geometriedaten des aktuellen Blockes zur Verfügung stehen. Dies gilt jedoch nicht für die Fluiddaten. Es werden immer nur die Strömungsgrößen für die Gitterzelle benötigt, in der sich das Partikel gerade aufhält. Dadurch ergeben sich bezüglich der Übertragung der Fluiddaten vom Memory-Manager zum Node verschiedene mögliche Strategien. Die erste Möglichkeit ist, daß jeweils nur die Daten für die aktuelle Gitterzelle übertragen werden. Sobald das Partikel in eine andere Zelle eintritt, werden die Strömungsgrößen für diese Zelle beim Memory-Manager angefordert. Die zweite Möglichkeit ist, daß ebenso wie die Gittergeometrie auch die Fluiddaten für den kompletten Block zu Beginn der Berechnung des Trajektoriensegments übertragen werden. Zwischen diesen beiden Strategien sind auch Abstufungen möglich, wie z.B. die Übertragung der Fluiddaten für eine bestimmte Umgebung der Gitterzelle.

Memory-Manager:

Ein Memory-Manager speichert die Gittergeometrie- und Fluiddaten eines Gitterblocks permanent während der gesamten Trajektorienberechnung. Wenn ein Node die Daten für diesen Block anfordert, werden sie zum entsprechenden Node gesendet.

Durch die dynamische Zuweisung von Partitionen zu Prozessoren wird mit dem DSM-Verfahren eine sehr gute Verteilung der Rechenlast unter den Prozessoren erreicht. Der Kommunikationsaufwand ist gegenüber dem DD-Verfahren höher, da nicht nur Partikeldaten, sondern zusätzlich auch Gitter- und Fluiddaten ausgetauscht werden müssen. Die Wahl der optimalen Strategie für die Übertragung der Fluidwerte (zellenweise, blockweise oder eine Zwischenstufe) ist u.a. abhängig von der Größe des zur Verfügung stehenden Speichers und der Schnelligkeit des Kommunikationsnetzwerkes des verwendeten Rechners.

Ein Nachteil des DSM-Verfahrens ist, daß es nur auf solchen Rechnern effizient eingesetzt werden kann, die es erlauben, einen Prozessor des Rechners mit mehreren Prozessen zu belegen. Durch die Einführung der zusätzlichen Memory-Manager-Prozesse werden wesentlich mehr Prozesse benötigt als Gitterblöcke vorhanden sind. Im allgemeinen ist die Anzahl der Prozesse gleich $2N+1$, wobei N die Anzahl der Gitterblöcke ist. Wenn diese Prozesse alle auf getrennten Prozessoren laufen, reduziert dies stark die Effizienz der Parallelisierung. Ist es jedoch möglich, einen Prozessor mit jeweils einem Node-Prozeß und einem Memory-Manager-Prozeß zu belegen, so gibt es zwischen diesen beiden Prozessen nur wenig gegenseitige Beeinflussung, da sie einen komplementären Ressourcenbedarf haben. Der Node-Prozeß benötigt wenig Speicher und viel Rechenleistung, wogegen der Memory-Manager-Prozeß viel Speicher und wenig Rechenleistung beansprucht.

Literatur zu 2.3/(Fremdliteratur und eigene Vorarbeiten)

[Zitierte Fremdliteratur]

[Crowe77] 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.

[Patankar80] Patankar S.V.
Numerical Heat Transfer and Fluid Flow
McGraw–Hill, New York, 1980.

[Perić92] 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.

[Perić93] Perić M., Lilek Ž.
Users Manual for the FAN–2D Software for the Calculation of Incompressible Flows
Institut für Schiffbau der Universität Hamburg, 1993.

[Schönung87] 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.

[Sommerfeld92] 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.

[Stone68] Stone, H.L.
Iterative solution of implicit approximations of multidimensional partial differential equations
SIAM J. Numer. Anal., Vol. 5, 1968, pp.530-558.

[Tsuji96] 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.

[Tysinger97] 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.

[Yone93] S. 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.

[**Veröffentlichungen zu eigenen Vorarbeiten**]

[Frank92] Frank Th.
Numerische Simulation der feststoffbeladenen Gasströmung im horizontalen Kanal unter Berücksichtigung von Wandrauhigkeiten
Dissertation, Techn. Universität Bergakademie Freiberg, 1992.

[Frank93] 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.

[Frank94] 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.

[Schade95] K.–P. Schade, Th. Frank, Th. Hädrich, D. Petrak
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.

2.4 Ergebnisse

2.4.1 Teilaufgabe Vergleich der PVM– und MPI–Implementierungen auf Cray–T3D und Parsytec GC–128

Beschreibung des Testfalls

Als Testfall wurde die Strömung eines Gas–Feststoff–Gemisches durch eine Reihe zueinander versetzt angeordneter Rohre ausgewählt (siehe Abb. 2.4). Diese Geometrieanordnung tritt z.B. in Rohrbündelwärmeübertragern auf, wobei insbesondere Erosion und Anhaftungserscheinungen an den einzelnen Rohrreihen von Interesse sind. Die Strömungsgeometrie weist eine Einströmöffnung auf der linken Seite, Symmetrierandbedingungen am oberen und unteren Geometrierand und eine Ausströmöffnung an der rechten Seite auf. Für die Tests wurden Vernetzungen der Strömungsgeometrie mit 4 bis 64 Gitterblöcken verwendet. Da sich aus Gründen der Gittergenerierung die in Abb. 2.4 gezeigte Geometrie jedoch nicht mit nur 4 Gitterblöcken vernetzen läßt, wurde für die Testrechnungen mit 4 Gitterblöcken (respektive Prozessoren) ein modifiziertes Gitter mit nur drei Rohrreihen verwendet. Leider zeigen die Ergebnisse, daß die so ermittelten Ergebnisse nicht vollständig vergleichbar sind mit den Ergebnissen, die auf dem Gesamtgitter erzielt wurden.

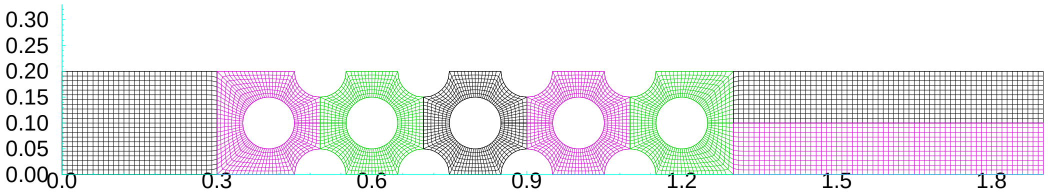

Abbildung 2.4: Numerisches Gitter für eine Strömung durch gegeneinander versetzte Rohrreihen; Vernetzung mit 8 Gitterblöcken.

Es wurde eine Gaseinströmgeschwindigkeit von $u_F = 10\ m/s$ gewählt. Die Gasphase (Luft unter Normalbedingungen) transportierte Feststoffpartikeln mit einer Dichte von $\rho_P = 2500\ kg/m^3$, für die eine Eintrittsgeschwindigkeit von $u_P = 9 \ldots 11\ m/s$ angenommen wurde. Die Partikelkonzentration war im Eintrittsquerschnitt gleichverteilt.

Es wurde jeweils ein Iterationszyklus bestehend aus der Berechnung des Fluidströmungsfeldes mit 3000 äußeren Iterationen und der Berechnung von 20000 Partikeltrajektorien ausgeführt. Auf Grund von CPU–Zeit–Beschränkungen auf der Cray T3D mußte die Zahl der berechneten Partikeltrajektorien für diese Testrechnungen auf 1000 reduziert werden. Um die Testrechnungen vergleichbar zu gestalten, wurden die Ausführungszeiten für den Parsytec GC–128 jeweils durch den entsprechenden Faktor dividiert.

Für den Testfall verwendete Soft– und Hardware

Die Berechnung des Testfalls wurde unter Verwendung der beiden in den Abschnitten 2.3.3 und 2.3.3 beschriebenen Parallelisierungsverfahren durchgeführt. Für das DSM–Verfahren wurden dabei zwei verschiedene Modifikationen verwendet:

1. Wenn bei der Berechnung einer Partikeltrajektorie ein bestimmtes CV (Control Volume, finites Volumenelement) durchquert wird, so werden zur Fortsetzung der Berechnung die Strömungsdaten des betreffenden und der angrenzenden CV's benötigt (5 CV's im 2–d Fall, 7 CV's im 3–d Fall). Die Strömungsdaten für diese Anzahl von CV's werden genau dann angeliefert, wenn eine Partikeltrajektorie das betreffende CV durchquert. Diese Art der Datenbereitstellung wird im folgenden mit DSM–Point–Methode bezeichnet. Sie ist charakterisiert durch einen häufigen Austausch kleiner Datenmengen.
2. Die gesamten zum aktuell bearbeiteten Gitterblock gehörenden Strömungsdaten werden vom Memory–Manager–Knoten zu dem rechnenden Knoten transferiert, sobald dem rechnenden Knoten eine Partikelanfangsbedingung für diesen Gitterblock zur Berechnung zugeteilt wird. Diese Art der Datenbereitstellung wird im folgenden als DSM–Block–Methode bezeichnet. Sie ist charakterisiert durch einen seltenen Austausch großer Datenmengen.

Als Testplattformen wurden der Parsytec GC–128 der TU Chemnitz und die Cray T3D des Edinburgh Parallel Computing Centre (EPCC) der Edinburgh University verwendet. Auf beiden Parallelrechnersystemen kamen zu PVM 3.2 und MPI 1.1 kompatible Softwareumgebungen zum Einsatz.

Ergebnisse der Testrechnungen

Insgesamt wurden 12 verschiedene Testserien durchgeführt, indem die Testrechnungen für die Strömungsgeoemtrie aus Abb. 2.4 auf numerischen Gitternetzen mit 4, 8, 16, 32 und 64 Gitterblöcken für jeweils zwei Hardware–Plattformen (Parsytec GC–128, Cray T3D), für Implementierungen mit zwei verschiedenen Parallelisierungsstandards (PVM, MPI) und für die drei verschiedenen Parallelisierungsverfahren für den Lagrange–Löser (DD, DSM–Point, DSM–Block) ausgeführt wurden. Für die Testrechnungen wurde jeweils die absolute Programmausführungszeit T_A, die Rechenzeit T_R, die Kommunikationszeit T_K und die für die Ein-/Ausgabe benötigte Zeit $T_{I/O}$ bestimmt. Aus den so ermittelten Werten wurde die Differenzzeit $T_D = T_A - T_R - T_K - T_{I/O}$ ermittelt. Diese Differenzzeit T_D besteht im wesentlichen aus den Wartezeiten für die einzelnen Prozessoren, die in synchronen Kommunikationsoperationen und in globalen Barrieren anfällt. Zu den in den Abb. 2.5 – 2.8 angegebenen Kommunikationszeiten T_K für das DD–Verfahren ist noch zu bemerken, daß es sich dabei um Mittelwerte über alle Prozessoren handelt. Da das DD–Verfahren für den hier betrachteten Testfall eine schlechte Lastverteilung aufweist, streuen die Kommunikationszeiten in einem weiten Bereich. Die Kommunikationszeiten für die Prozessoren mit einer geringen Rechenlast schließen die Zeit des Wartens auf die Prozessoren mit einer hohen Rechenlast mit ein.

Die Abb. 2.5 – 2.8 zeigen den Vergleich der Ergebnisse für die drei Parallelisierungsverfahren untereinander. Dabei zeigen die Ergebnisse für den Parsytec GC–128 ein mitunter schwer zu interpretierendes Bild. Insbesondere die mit der PVM–Implementierung auf dem Parsytec GC–128 erzielten Meßwerte (Abb. 2.5) lassen im Vergleich mit den auf der Cray T3D erzielten Ergebnissen (Abb. 2.7) auf Probleme der PVM–Implementierung auf dem Parsytec GC–128 schließen. Generell

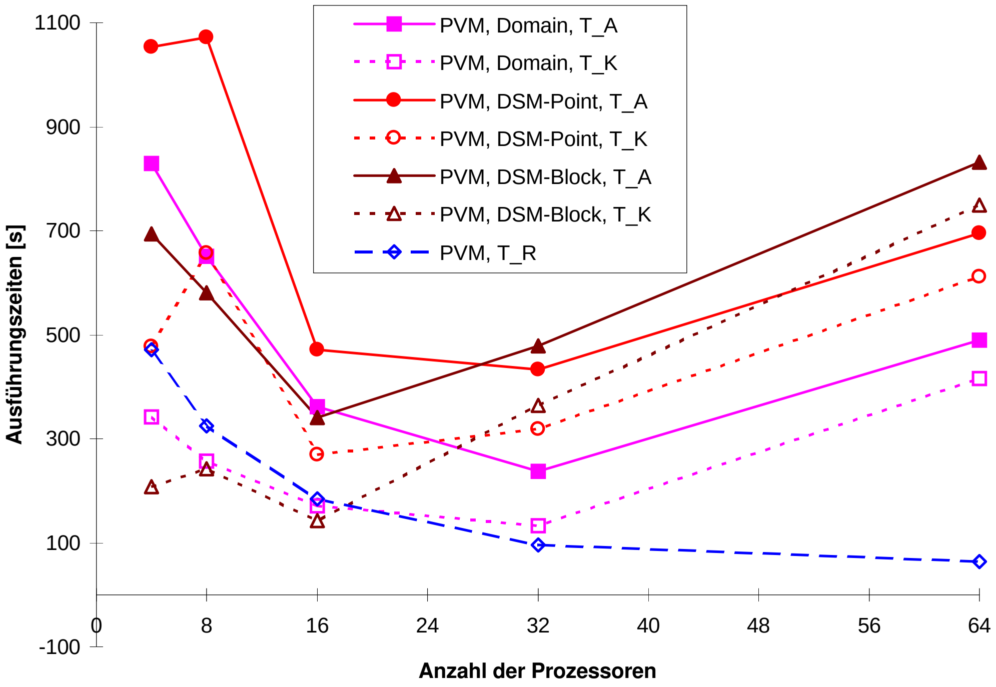

Abbildung 2.5: Ausführungszeiten der PVM–Implementierungen auf dem Parsytec GC–128.

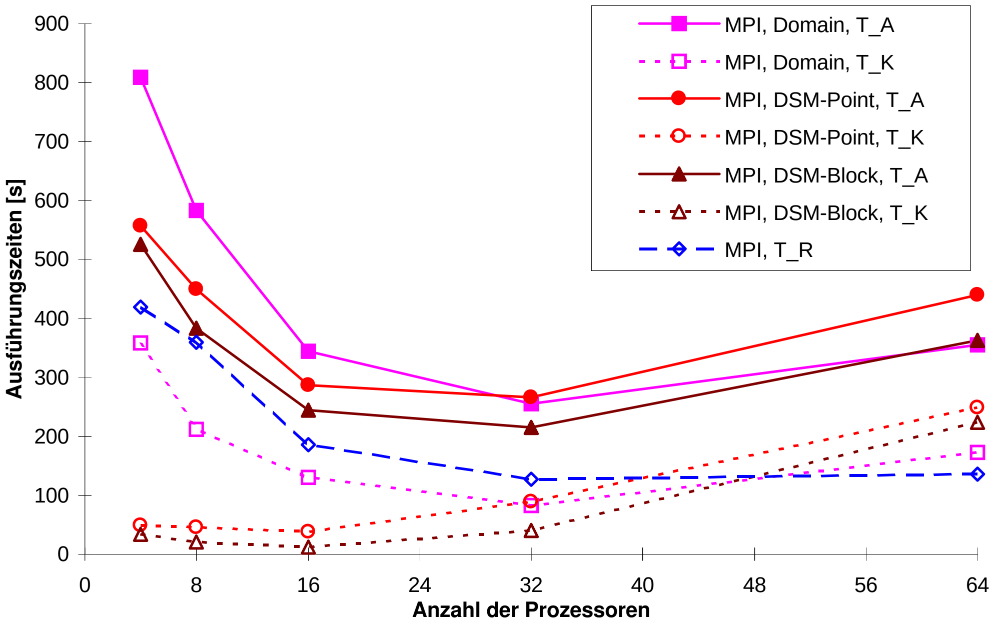

Abbildung 2.6: Ausführungszeiten der MPI–Implementierungen auf dem Parsytec GC–128.

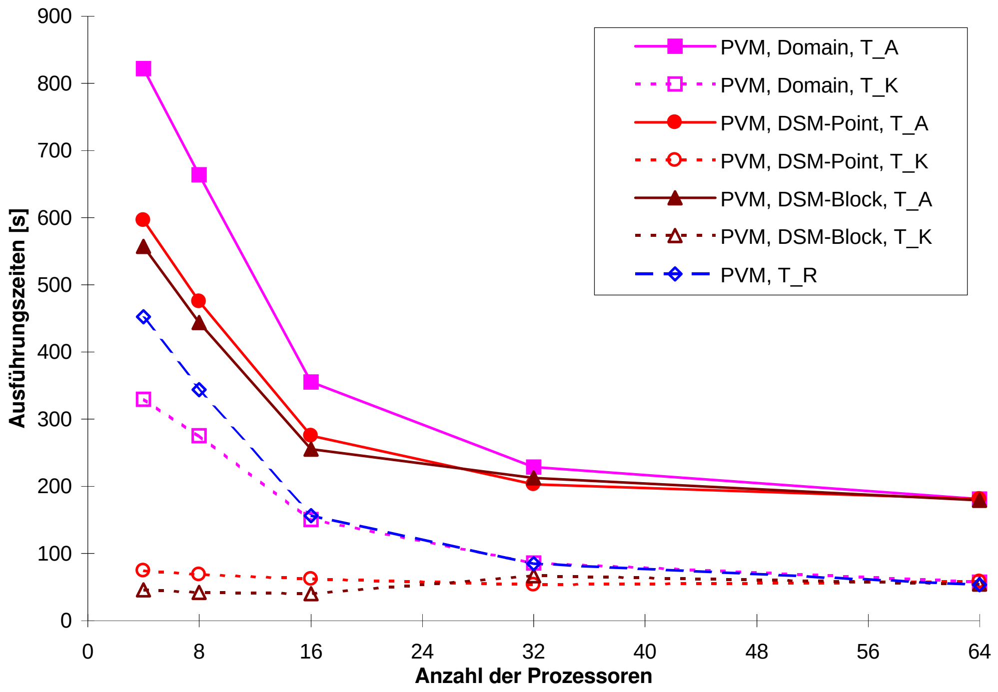

Abbildung 2.7: Ausführungszeiten der PVM–Implementierungen auf der Cray T3D.

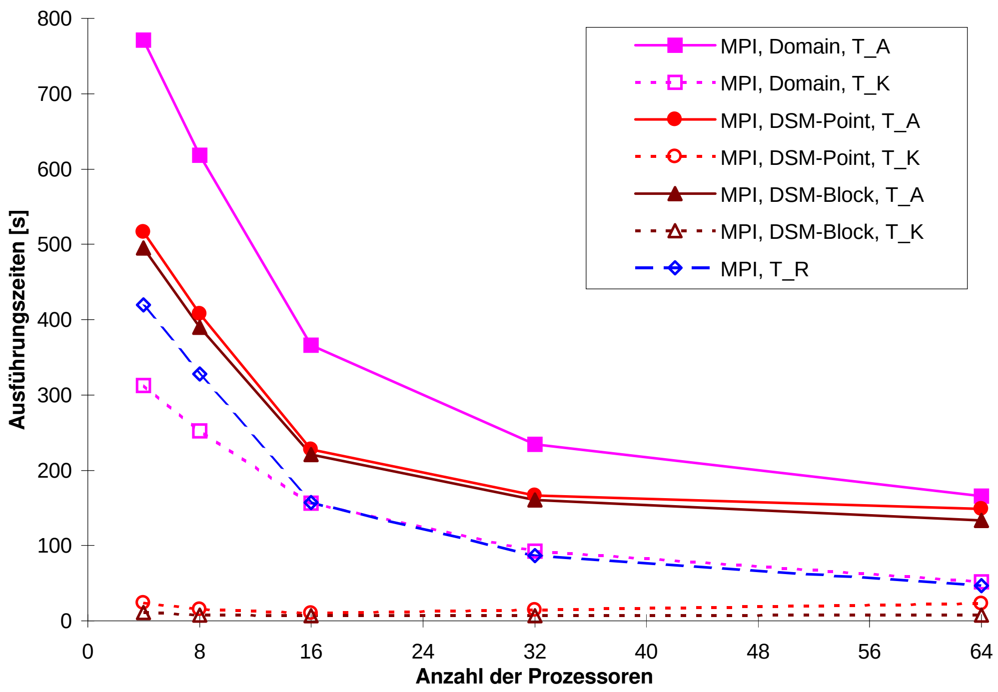

Abbildung 2.8: Ausführungszeiten der MPI–Implementierungen auf der Cray T3D.

läßt sich für die Testergebnisse auf dem Parsytec GC–128 feststellen, daß die geringere Kommunikationsleistung dieser MIMD–Architektur bei einer Skalierung über eine größere Anzahl von verwendeten Prozessoren zu einer früheren Stagnation in den Programmausführungszeiten führt. Unabhängig vom eingesetzten Parallelisierungsverfahren steigen die Programmausführungszeiten beim Übergang von 32 auf 64 Prozessoren (respektive Gitterblöcken des numerischen Gitternetzes) wieder an, was auf den Kommunikationsengpaß zwischen den 32–Knoten–Partitionen des Parsytec GC–128 zurückgeführt werden kann. Für die Testrechnungen auf 4 bis 32 Gitterblöcken zeigt Abb. 2.6 für den Parsytec GC–128 einen Vorteil der DSM–Verfahren gegenüber der Domain Decomposition für das Lagrange'sche Lösungsverfahren. Die großen Setup–Zeiten für die Knoten–Kommunikation auf dem Parsytec GC–128 führen zu einem klaren Abstand zwischen dem DSM–Point– und dem DSM–Block–Verfahren. Letzteres ist für die Architektur des Parsytec GC–128 das vorteilhafteste Verfahren, da es relativ seltene Knoten–Knoten–Kommunikation mit größeren Datenmengen mit dem Vorteil der dynamischen Lastverteilung des DSM–Verfahrens verbindet.

Die Testergebnisse auf der Cray T3D (Abb. 2.7 und 2.8) zeigen ein einheitlicheres und besser interpretierbares Bild. Über den gesamten Skalierungsbereich von 4 bis 64 Prozessoren kann für alle drei eingesetzten Parallelisierungsverfahren eine fortschreitende Reduktion der Programmausführungszeiten festgestellt werden. Auch können sich auf Grund der besseren Lastverteilung die DSM–Verfahren gegenüber dem Domain Decomposition Verfahren klar durchsetzen. Die Unterschiede zwischen dem DSM–Point– und dem DSM–Block–Verfahren sind jedoch auf Grund der sehr hohen Kommunikationsleistung (große Bandbreite, kurze Setup–Zeiten) der Cray T3D nur sehr gering. Ein detaillierterer Blick auf die gemessenen Einzelzeiten zeigt für die Testrechnungen auf 64 Gitterblöcken, daß der Anteil der Kommunikations– und I/O–Zeiten bei 2/3 der absoluten Programmausführungszeiten liegt, während die Rechenzeiten um einen Faktor von 0.63 gegenüber den Testrechnungen auf 32 Gitterblöcken weiter gesenkt werden konnten. Dies zeigt, daß die vorliegenden Testergebnisse für die Cray T3D maßgeblich durch die CPU–Zeit–Beschränkungen am EPCC beeinflußt wurden, die eine Reduzierung des ursprünglich vorgesehenen Testfalls auf die Berechnung von 1000 Partikeltrajektorien notwendig machten.

2.4.2 Teilaufgabe Verifikation des 3–dimensionalen Berechnungsverfahrens an einer Zyklonströmung

Neben der Entwicklung und Implementierung eines 3–dimensionalen Euler–Lagrange–Verfahrens für die Berechnung disperser Mehrphasenströmungen auf der Basis blockstrukturierter Gitternetze und der Entwicklung leistungsfähiger Parallelisierungsverfahren war auch die Verifikation der im Verfahren eingesetzten physikalisch–mathematischen Modelle ein wesentlicher Bestandteil des Forschungsvorhabens. Leider stand für einen derartigen Vergleich experimenteller und numerischer Ergebnisse kein umfassendes experimentelles Datenmaterial für eine geeignete 3–dimensionale Mehrphasenströmung zur Verfügung, das den Vergleich auf der Ebene der Geschwindigkeitsfelder (oder anderer die Strömung charakterisierender

Feldgrößen) mit hoher räumlicher Auflösung ermöglicht hätte. So fiel die Wahl auf die numerische Simulation der Partikelabscheidung in einer Serie von geometrisch ähnlichen Standardzyklonen. Die Hauptgründe für diese Auswahl waren :

1. Bei der Zyklonströmung handelt es sich um eine ausgeprägt 3–dimensionale Strömung, die nicht auf ein ebenes oder rotationssymmetrisches Problem reduziert werden kann.

2. Die Zyklonströmung ist in der Vergangenheit durch eine Vielzahl von Autoren untersucht worden, so daß detaillierte Grundkenntnisse über diesen Strömungstyp in der Fachliteratur vorliegen.

3. Der Wissenstand über partikelbeladene Gasströmungen in Zyklonen gestattet eine qualitative Bewertung der berechneten Gas–Partikel–Strömung.

4. Das experimentelle Datenmaterial aus der Dissertation von König [König90] gestattet darüber hinaus den quantitativen Vergleich mit den numerisch bestimmten Kenngrößen auf der Ebene integraler Parameter wie Gesamtdruckverlust und Partikeltrenngrad.

In der Folge wurden mit dem Euler–Lagrange–Verfahren (FAN–3D, PartFlow) numerische Berechnungen für die Partikelabscheidung von Quartzpartikeln ($d_P = 0 \ldots 50\ \mu m$, $\rho_P = 2500\ kg/m^3$) in einer Serie von im Maßstab 1:2:4:8 ähnlichen Standardzyklonen (Z10,. . .,Z80) durchgeführt [Frank97c, Frank98a]. Die Geometrie entsprach den von König [König90] untersuchten Standardzyklonen (Abb. 2.10). Für die numerische Berechnung wurden die Zyklone mit einem numerischen Gitter mit 42 Gitterblöcken und ca. 250.000 finiten Volumenelementen vernetzt (Abb. 2.11). Die Gaseintrittsgeschwindigkeiten wurden in Abhängigkeit von der Zyklongröße zwischen $u_F = 3.8\ m/s$ und $u_F = 24.9\ m/s$ variiert.

Das Geschwindigkeitsfeld der Gasphase zeigt den für Zyklone typischen asymmetrischen Hauptwirbel sowie die aus der Literatur bekannte Sekundärströmung im oberen Teil des Zykloninnenraumes entlang des Zyklondeckels und weiter abwärts entlang der Außenwand des Tauchrohres (Abb. 2.12). Der durch das Tauchrohr entweichende Gasstrom weist kurz hinter dem Eintritt in das Tauchrohr noch einen erheblichen Drall auf, der erst mit zunehmender Lauflänge abgebaut wird.

Für jede der durchgeführten Strömungsberechnungen wurde des weiteren der Gesamtdruckverlust [Frank98a] und der Partikeltrenngrad

$$T(d_P) = 1 - \frac{\dot{N}_{out}(d_P)}{\dot{N}_{in}(d_P)}$$

in Abhängigkeit vom Partikeldurchmesser d_P ermittelt. Der Vergleich mit den experimentellen Ergebnissen zeigt eine gute bis sehr gute Übereinstimmung. Wird der berechnete Gesamtdruckverlust geringfügig unterbestimmt, so zeigt sich für die berechneten Partikeltrenngrade für alle Zyklongrößen und Gaseintrittsgeschwindigkeiten eine sehr gute Übereinstimmung. Die Tendenz zu verbesserter Partikelabscheidung bei erhöhter Gaseinrittsgeschwindigkeit wird von den numerischen Berechnungen für alle untersuchten Zyklone richtig wiedergegeben (Abb. 2.9).

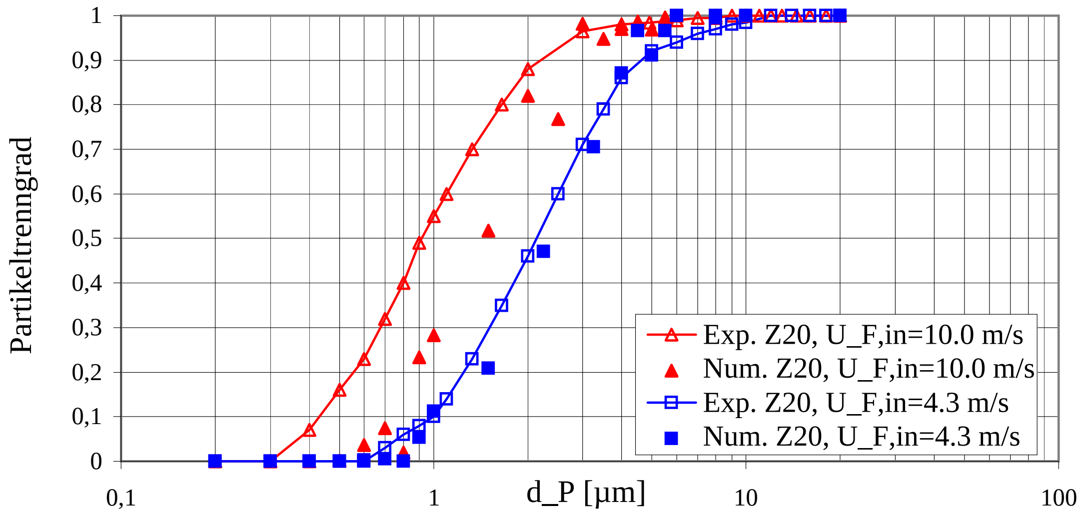

Abbildung 2.9: Vergleich der Partikeltrenngrade für Z20 und Gaseintrittsgeschwindigkeiten $u_F = 4.3\ m/s$ und $10\ m/s$.

Die Rechnungen zur numerischen Simulation der Partikelabscheidung in Zyklonen erfolgten wegen der permanent hohen Auslastung des Parsytec Power–GC 128 mit der seriellen Programmversion auf der SGI Cray Origin 2000 am URZ der TU Dresden. Eine parallele Berechnung war dort wegen der hohen Nutzeranzahl und der insgesamt 54 verfügbaren Prozessoren ebenfalls nicht möglich.

Weitere detaillierte Ergebnisse sind in den Veröffentlichungen [Frank97c, Frank98a] enthalten.

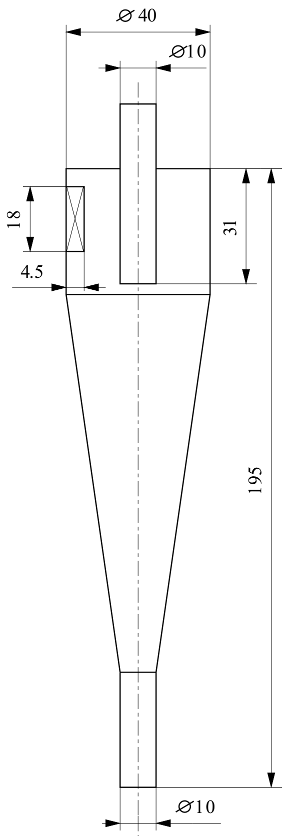

Abbildung 2.10: Schematische Darstellung des Standardzyklons Z10.

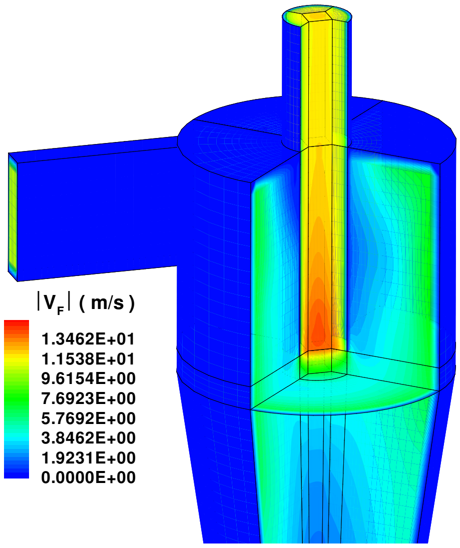

Abbildung 2.12: Geschwindigkeitsverteilung der mittleren Gasgeschwindigkeiten im oberen Teil des Z10–Zyklons, $u_F = 10\ m/s$.

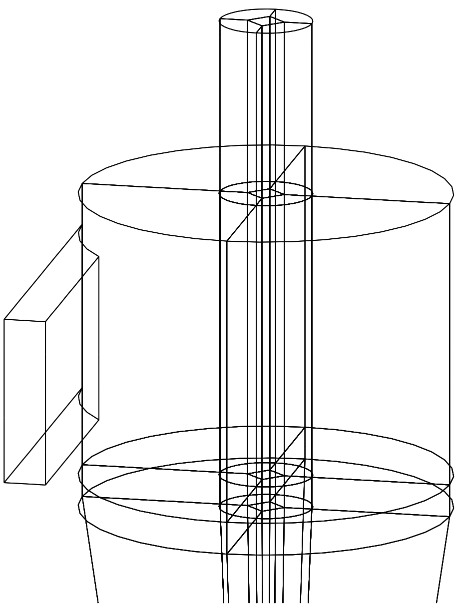

Abbildung 2.11: Gitterblockstruktur im oberen Teil des Zyklons.

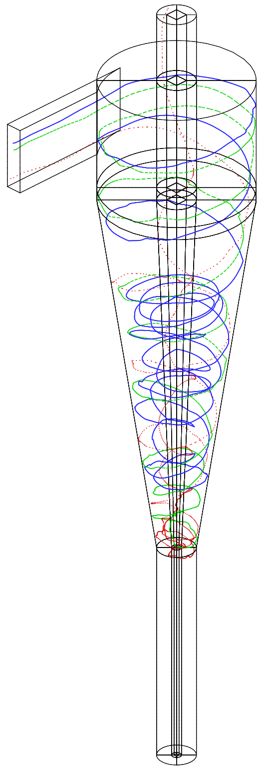

Abbildung 2.13: Partikeltrajektorien in Z10 für Gaseintrittsgeschwindigkeiten $u_F = 10\ m/s$, $d_P = 1, \ldots, 5\ \mu m$.

2.4.3 Teilaufgabe Parallele, numerische Simulation einer Gas–Feststoff–Strömung in einem 3–dimensionalen Kanal

Unter 2.4.1 sind die auf der Cray T3D am EPCC in Edinburgh und auf dem Parsytec Power–GC 128 an der TU Chemnitz durchgeführten Untersuchungen der Parallelisierungsverfahren aus 2.3.3 und 2.3.3 ausführlich diskutiert worden. Wegen des frühen Zeitpunktes der Durchführung erstrecken sich diese Untersuchungen jedoch ausschließlich auf das 2–dimensionale Euler–Lagrange–Verfahren. Mit dem Fortgang der Entwicklungsarbeiten ergab sich im Frühjahr 1998 die Möglichkeit, die entwickelten Parallelisierungsverfahren (Domain Decomposition, DSM) auch für die Berechnung einer einfachen 3–dimensionalen Gas–Feststoff-Strömung einzusetzen. Als Testkonfiguration wurde eine Gas–Feststoff–Strömung in einem gekrümmten Kanal mit quadratischem Querschnitt gewählt.

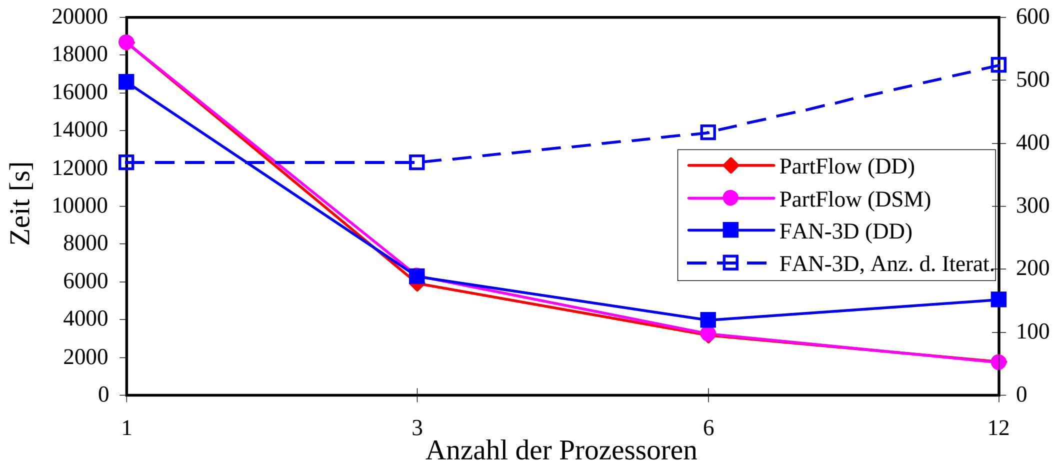

Abbildung 2.14: Vergleich der Ausführungszeiten von FAN–3D und PartFlow für einen 3–dimensionalen Testfall.

Die Ausgangskonfiguration des numerischen Gitternetzes weist 3 Gitterblöcke mit insgesamt 6700 finiten Volumenelementen auf. Die Gaseintrittsgeschwindigkeit lag bei 10 m/s, der Turbulenzgrad wurde im Eintritt mit 10% angenommen, die Feststoffphase entsprach in ihren physikalischen Eigenschaften Quartzsand mit einem Partikeldurchmesser $d_P = 210 \ldots 410\ \mu m$. Die Testfallrechnungen wurden mit der mit MPI parallelisierten Programmversion (FAN–3D, PartFlow) auf der SGI Cray Origin 2000 am URZ der TU Dresden ausgeführt. Das Diagramm in Abb. 2.14 zeigt die Ausführungszeiten des Navier–Stokes–Lösers FAN–3D und des Lagrange–Lösers PartFlow (Parallelisierung mit Domain Decomposition (DD) bzw. Distributed–Shared–Memory–Verfahren (DSM)) im Vergleich mit der auf nur einer CPU ausgeführten seriellen Programmversion.

Die für FAN–3D ermittelten Ausführungszeiten (rechte Skale) zeigen für diesen speziellen Testfall schon für die Abarbeitung auf 12 Prozessoren ein erneutes Ansteigen der Ausführungszeiten. Hauptursache hierfür ist vordergründig die für eine effiziente Parallelisierung des Navier–Stokes–Lösers zu geringe Anzahl finiter Volumenelemente pro Prozessor (Ergebnis der Beschränkungen für diesen Testfall). Aber auch der Anstieg der für eine konvergente Lösung notwendigen Iterationszyklen (ebenfalls rechte Skale) des Gleichungslösers für das System aus linearisierten Gleichungen (SIP–Solver) ist für diese Entwicklung mit verantwortlich. Demgegenüber zeigen die Ergebnisse für die Parallelisierungsverfahren für den Lagrange'schen Berechnungsanteil (linke Skale) sehr gute Skalierungseigenschaften. Auf Grund der homogenen Konzentrationsverteilung der dispersen Phase im untersuchten Strömungsbeispiel sind die zu beobachtenden Unterschiede zwischen DD– und DSM–Verfahren jedoch eher gering.

Literatur zu 2.4/(eigene Veröffentlichungen im Berichtszeitraum)

[**DFG–Forschungsberichte**]

[Frank97b] Th. Frank, E. Wassen
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.

[**Begutachtete Konferenzbeiträge, Proceedings**]

[Frank96a] Th. Frank
Comparison of three parallelization methods for calculation of disperse multiphase flows using the Lagrangian approach
Proc. 3rd Int. Conference "Parallel CFD '96", Implementations and Results Using Parallel Computers, Capri, Italy, May 20–23, 1996.

[Frank96b] Th. Frank, E. Wassen
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.

[Frank97c] Th. Frank, E. Wassen, Q. Yu
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.

[Frank97d] Th. Frank, E. Wassen
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.

[Frank98a] Th. Frank, E. Wassen, Q. Yu
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.

[Schade95] K.–P. Schade, Th. Frank, Th. Hädrich, D. Petrak
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 European Conference of ILASS–Europe on Atomization and Sprays, 21.–23. March 1995, Nürnberg, Germany.

[Wassen97] E. Wassen, Th. Frank, Q. Yu
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.

[**Konferenzbeiträge in Fachzeitschriften**]

[Frank97e] Th. Frank, E. Wassen, Q. Yu
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.

[Frank98c] Th. Frank, E. Wassen, Q. Yu
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.

[**Weitere zitierte Literatur**]

[König90] C. König
Untersuchungen zum Abscheideverhalten von geometrisch ähnlichen Zyklonen
Dissertation, Universität Kaiserslautern, Germany, 1990.

2.5 Offene Fragen/Ausblick

Im Berichtszeitraum sind im Teilprojekt D2 eine Reihe substantieller Lösungen für die parallele numerische Simulation von stationären, 2– und 3–dimensionalen, dispersen Mehrphasenströmungen auf MIMD– und anderen HPC–Rechnerplattformen erarbeitet worden. Die durchgeführten Untersuchungen haben gezeigt, daß Euler–Lagrange–Verfahren effektiv auf MIMD–Architekturen einzusetzen sind. Durch die Implementation der z.Zt. gebräuchlichsten Parallelisierungsstandards PVM 3.2 und MPI 1.1 ist eine außerordentlich breite Palette von HPC–Rechnerarchitekturen (MIMD, SMP–Shared Memory, WS–Cluster–Computing) abgedeckt, auf denen das numerische Verfahren ohne umfassende Änderungen eingesetzt werden kann.

Trotz der guten Fortschritte in der Bearbeitung des Forschungsvorhabens haben sich Fragestellungen ergeben, die der weitergehenden Untersuchung und Bearbeitung bedürfen.

1. Die entwickelten numerischen Verfahren sind in der Lage, 2– und 3–dimensionale, stationäre Mehrphasenströmungen mit Phasenwechselwirkungen zu berechnen. Insbesondere für 3–dimensionale Strömungen ist jedoch die Annahme der Stationarität der Strömung häufig nicht gegeben. Typische Beispiele derartiger Mehrphasenströmungen sind vertikale Gas–Tropfen– oder Gas–Feststoff–Strömungen, wie sie bei Sprühtrocknungsprozessen, bei der Rauchgasreinigung und beim pneumatischen Transport in vertikalen Rohrleitungen auftreten. Das Wechselspiel aus Schwerkraft und der auf die Partikeln wirkenden Strömungskräfte kann bei geringen Schwankungen in der Fluidströmung zu einem ausgeprägten instationären Strömungsbild führen. Dies unterstreicht die Bedeutung der Entwicklung effektiver Berechnungsverfahren für die Simulation instationärer, phasengekoppelter Mehrphasenströmungen.

2. Mit der zunehmenden Hinwendung zur Simulation instationärer und/oder 3–dimensionaler, disperser Mehrphasenströmungen gewinnt die Stabilität und das Konvergenzverhalten der verwendeten Gleichungslöser für großdimensionierte, lineare Gleichungssysteme zunehmend an Bedeutung. Da der bisher in den Navier–Stokes–Lösern von M. Perić verwendete Lösungsalgorithmus (SIP–Solver nach Stone) für eine größere Anzahl von Finite–Volumen–Elementen und zunehmende Gitterblockanzahlen (d.h. verwendete Prozessoren) zu einem Anstieg der für eine konvergente Lösung benötigten Iterationszyklen, langen Iterationszeiten und geringer Konvergenzgeschwindigkeit führt, wird die Implementation verbesserter Gleichungslöser als notwendig betrachtet.

3. Für das entwickelte parallele Euler–Lagrange–Verfahren wurde der Weg eines mit kommerziellen Programmen extern durchgeführten Pre– und Postprozessings gewählt. Mit zunehmender Problemgröße, Anzahl der finiten Volumenelemente im numerischen Gitternetz und Anzahl der verwendeten Prozessoren des Parallelrechnersystems nimmt die Bedeutung einer parallelen Daten–Ein-/Ausgabe zu. Die momentan verwendete serielle Daten–Ein/Ausgabe führt mit steigender Prozessorzahl zu einem zunehmend ungünstigeren Verhältnis zwischen der insgesamt benötigten Ausführungszeit und der reinen Rechenzeit.

Die Weiterentwicklung der parallelen Lösungsalgorithmen für Euler–Lagrange–Verfahren unter Berücksichtigung der hier aufgezeigten Fragestellungen wird einen Schwerpunkt der Arbeiten im Teilprojekt D2 im bevorstehenden Bearbeitungszeitraum bilden.