

Arbeitsbericht

Parallele Algorithmen für die numerische Simulation von Mehrphasenströmungen in komplexen 3-dimensionalen Geometrien

Abschlußbericht

DFG-Forschungsvorhaben im Normalverfahren

Projektnummer :	Fr 1069/3-1
Projektdauer :	15. Mai 1995 — 14. Mai 1997
Antragsteller :	Dr. Thomas Frank
Ort der Durchführung :	TU Chemnitz-Zwickau, FG Mehrphasenströmungen
Berichtsdatum :	1. Juni 1997

Inhaltsverzeichnis

1	Einleitung	3
2	Aufgabenstellung und Ziele des Forschungsprojektes	4
3	Beschreibung des Berechnungsverfahrens	5
3.1	Die Bewegung der kontinuierlichen Phase	5
3.1.1	Grundgleichungen	5
3.1.2	Numerische Lösung	5
3.2	Die Bewegung der dispersen Phase	6
3.2.1	Das Lagrange-Verfahren	6
3.2.2	Partikel-Wand-Kollision	8
3.3	Phasenkopplung und Berücksichtigung der Fluidturbulenz	9
3.3.1	Phasenkopplung	9
3.3.2	Berücksichtigung der Fluidturbulenz	10
4	Parallelisierung des Berechnungsverfahrens	11
4.1	Das Parallelisierungsverfahren für den Navier–Stokes–Löser	11
4.2	Die verschiedenen Parallelisierungsalgorithmen für das Lagrange–Verfahren . . .	12
4.2.1	Das Quasiserielle (QS) Verfahren	12
4.2.2	Das Domain Decomposition (DD) Verfahren	13
4.2.3	Das Distributed Shared Memory (DSM) Verfahren	14
5	Untersuchungen der numerischen und parallelen Effizienz	17
5.1	Vergleichende Untersuchungen der eingesetzten Parallelisierungsverfahren auf dem Parsytec GC–128	17
5.1.1	Die vertikale Kanalströmung	17
5.1.2	Die Umströmung einer Vollkegeldüse	17
5.1.3	Das MIMD–Parallelrechnersystem	18
5.1.4	Vergleich der parallelen Effizienz der drei Parallelisierungsverfahren	18
5.2	Vergleich der PVM– und MPI–Implementierungen auf Cray T3D und Parsytec GC–128	19
5.2.1	Gas–Feststoff–Strömung durch eine Rohrbündelreihe	23
5.2.2	Für den Testfall verwendete Hard– und Software	24
5.2.3	Ergebnisse der Testrechnungen auf Parsytec GC–128 und Cray T3D . . .	25
6	Arbeiten zur Berechnung 3–dimensionaler, disperser Mehrphasenströmungen	32
6.1	Grundlagen des 3–dimensionalen Euler–/Lagrange–Verfahrens	32
6.2	Geometrische Grundalgorithmen für das Lagrange–Lösungsverfahren	34
6.2.1	Lokalisierung der Partikelanfangsbedingungen auf dem numerischen Gitternetz	35
6.2.2	Der Partikel–Verfolgungsalgorithmus	37
6.2.3	Berechnung der Phasenwechselwirkungs-Quellterme und der mittleren Strömungsparameter der dispersen Phase	37
6.2.4	Behandlung des Partikel–Wand–Stoßvorganges	38
6.3	Zur Übertragbarkeit der Parallelisierungsverfahren	38
7	Teilnahme an Fachtagungen und Konferenzen	39

8	Andere Aktivitäten im Rahmen des Forschungsaufenthaltes an der Edinburgh University im Rahmen des TRACS-Programms	40
9	Abschließende Bemerkungen	41
10	Eigene Veröffentlichungen im Zeitraum 1995–1997	42
	Literatur	43

Abbildungsverzeichnis

1	Prinzip der Gitterpartitionierung	11
2	Prinzip des Quasiseriellen Verfahrens	12
3	Prinzip des Domain Decomposition Verfahrens	13
4	Prinzip des Distributed Shared Memory Verfahrens	15
5	Axialsymmetrische Umströmung einer Vollkegeldüse im Gegenstrom	17
6	Ausführungszeiten T_A für die vertikale Kanalströmung.	20
7	Speed-up S_N für die vertikale Kanalströmung.	20
8	Effizienz E_N für die vertikale Kanalströmung.	20
9	Ausführungszeiten T_A für die Umströmung der Vollkegeldüse.	21
10	Speed-up S_N für die Umströmung der Vollkegeldüse.	21
11	Effizienz E_N für die Umströmung der Vollkegeldüse.	21
12	Numerisches Gitter für die Strömung durch die gegeneinander verschobenen Rohrreihen; Vernetzung mit 8 Gitterblöcken.	23
13	Ausführungszeiten der PVM-Implementationen auf dem Parsytec GC-128.	26
14	Ausführungszeiten der MPI-Implementationen auf dem Parsytec GC-128.	26
15	Ausführungszeiten der PVM-Implementationen auf der Cray T3D.	27
16	Ausführungszeiten der MPI-Implementationen auf der Cray T3D.	27
17	Abhängigkeit der Testergebnisse vom Berechnungsumfang	28
18	Vergleich der Implementationen des DD-Verfahrens auf Parsytec-GC und Cray T3D.	29
19	Vergleich der Implementationen des DSM-Point-Verfahrens auf Parsytec-GC und Cray T3D.	29
20	Vergleich der Implementationen des DSM-Block-Verfahrens auf Parsytec-GC und Cray T3D.	29
21	Lastverteilung für das DD-Verfahren auf 8 Gitterblöcken.	31
22	Lastverteilung für das DSM-Block-Verfahren auf 8 Gitterblöcken.	31
23	Form eines allgemeinen Hexaeder-Kontrollvolumens des 3-dimensionalen, numerischen Gitternetzes.	32
24	Zerlegung eines dodecaeder-förmigen Kontrollvolumens in sechs tetraeder-förmige Teilkontrollvolumina.	34
25	Lokalisierung einer Partikelposition auf dem numerischen Gitter mit dem 'Single-shot' oder Strahlverfolgungs-Verfahren.	35
26	Lokalisierung einer Partikelposition P_I auf dem numerischen Gitter durch fortschreitende Bisektion der Gitterteilstruktur.	36
27	Der Partikel-Verfolgungsalgorithmus.	37

1 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 Masse-, Impuls- und Energieaustausch 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 10) nur zwei Arbeiten aus der internationalen Fachliteratur bekannt, die geeignete Parallelisierungsansätze für das Lagrange-Verfahren darstellen [20, 21].

Im nächsten Kapitel werden zunächst die Aufgabenstellung und die Ziele des Forschungsvorhabens vorgestellt. Danach wird das Berechnungsverfahren präsentiert, das in diesem Vorhaben benutzt wird. In der Folge werde drei verschiedene Parallelisierungskonzepte für das Euler-/

Lagrange–Verfahren zur Berechnung disperser Mehrphasenströmungen vorgestellt und diskutiert. Die Verfahren werden an drei repräsentativen Testfällen untersucht und die Ergebnisse für zwei unterschiedliche MIMD–Architekturen gegenübergestellt.

2 Aufgabenstellung und Ziele des Forschungsprojektes

Für das vorliegende Forschungsprojekt waren folgende Aufgaben vorgesehen :

- Parallelisierung des von Perić und Lilek [13] entwickelten Navier–Stokes–Lösers FAN–2D/3D nach der Gebietszerlegungsmethode,
- Anpassung des Pre– und Postprozessings an die auf MIMD–Architekturen benötigten Datenstrukturen,
- Entwicklung eines Lagrange–Lösungsverfahrens für die Berechnung der Bewegung der dispersen Phase auf der Grundlage der zur Verfügung stehenden FVM–Verfahren auf blockstrukturierten, 2– und 3–dimensionalen numerischen Gitternetzen,
- Entwicklung von geeigneten Verfahren zur Parallelisierung von Euler–/Lagrange–Läsern auf blockstrukturierten Gitternetzen für massiv parallele MIMD–Rechnerarchitekturen.

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 den zukünftigen Parallelrechnern effektiv durchgeführt werden können.

Da sich die Fertigstellung des 3–dimensionalen Navier–Stokes–Lösers FAN–3D durch Prof. M. Perić entgegen den vor Projektbeantragung getroffenen Aussagen stark verzögerte, wurde ein Teil von Verfahrensentwicklungen und Untersuchungen auf 2–dimensionale Problemstellungen beschränkt. Die aus diesen Entwicklungen hervorgegangenen Parallelisierungsverfahren und Ergebnisse sind jedoch ohne Einschränkung auf den 3–dimensionalen Fall direkt übertragbar.

Die genannten Ziele wurden zum größten Teil erreicht. Die nächsten Abschnitte zeigen die Vorgehensweise, beschreiben die verschiedenen entwickelten Parallelisierungsverfahren und die für eine Reihe von repräsentativen Testfällen erzielten Ergebnisse im Detail.

3 Beschreibung des Berechnungsverfahrens

Die numerische Simulation disperser Mehrphasenströmungen umfaßt sowohl die Simulation der fluiden Phase als auch der dispersen Phase. Die Grundgleichungen für die Bewegung der fluiden Phase werden in Kapitel 3.1 angegeben. Außerdem werden dort die Grundzüge der numerischen Lösung dieser Gleichungen dargestellt. In Kapitel 3.2 wird das Lagrange-Verfahren beschrieben, das zur Berechnung der Bewegung der dispersen Phase verwendet wird. Die Berücksichtigung des Einflusses der dispersen Phase auf die fluide Phase, d.h. die Phasenkopplung, wird in Kapitel 3.3 erläutert. Des weiteren wird in diesem Kapitel beschrieben, wie die Fluidturbulenz bei der Berechnung der Bewegung der dispersen Phase berücksichtigt wird.

3.1 Die Bewegung der kontinuierlichen Phase

3.1.1 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:

$$\begin{aligned} \frac{\partial}{\partial x}(\rho_F u_F \Phi) + \frac{\partial}{\partial y}(\rho_F v_F \Phi) + \frac{\partial}{\partial z}(\rho_F w_F \Phi) = \\ \frac{\partial}{\partial x} \left(\Gamma_\Phi \frac{\partial \Phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\Gamma_\Phi \frac{\partial \Phi}{\partial y} \right) + \frac{\partial}{\partial z} \left(\Gamma_\Phi \frac{\partial \Phi}{\partial z} \right) + S_\Phi + S_\Phi^P \end{aligned} \quad (1)$$

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 (Masse-, Impuls-, Energieaustausch) zwischen disperser und fluider Phase beschreibt.

Die Variable Φ kann die Werte 1, u_F , v_F , w_F , k und ε annehmen. Für $\Phi = 1$ ergibt sich aus Gleichung (1) die Kontinuitätsgleichung, für $\Phi = u_F$, $\Phi = v_F$ bzw. $\Phi = w_F$ ergeben sich die Impulsgleichungen in x -, y - bzw. z -Richtung, für $\Phi = k$ und $\Phi = \varepsilon$ die Gleichungen für die Turbulenzenergie und die Dissipationsrate.

Der Zusammenhang zwischen Φ , S_Φ , S_Φ^P und Γ_Φ sowie die verwendeten Konstanten sind in Tabelle 1 detailliert 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, zu denen z.B. die Gravitationskraft zählt.

Die Berechnung der Quellterme $S_{u_i}^P$ wird in Kapitel 3.3 beschrieben. Der Einfluß der dispersen Phase auf die Fluidturbulenz wurde hier vernachlässigt, weshalb $S_k^P = S_\varepsilon^P = 0$ ist.

3.1.2 Numerische Lösung

Die Grundlage für die numerische Lösung der Gleichungen für die Fluidströmung bildet eine Finite-Volumen-Diskretisierung dieser Gleichungen. Das Strömungsgebiet wird durch ein strukturiertes, nicht-orthogonales numerisches Gitter beschrieben, das im zweidimensionalen Fall aus Vierecken und im dreidimensionalen Fall aus Hexaedern besteht. Um auch komplexe Geometrien, wie sie im allgemeinen in technischen Anlagen zu finden sind, beschreiben zu können, ist das Gitter außerdem block-strukturiert. Zur Lösung der diskretisierten und linearisierten Gleichungen wird Stone's SIP-Methode verwendet [17]. Die Druck-Geschwindigkeits-Kopplung wird realisiert durch die Anwendung eines SIMPLE-Algorithmus [9, 11].

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	$\frac{\mu_t}{\sigma_\varepsilon}$
$P_k = \mu_t \left\{ 2 \cdot \left[\left(\frac{\partial u_F}{\partial x} \right)^2 + \left(\frac{\partial v_F}{\partial y} \right)^2 + \left(\frac{\partial w_F}{\partial z} \right)^2 \right] \right.$ $\left. + \left(\frac{\partial u_F}{\partial y} + \frac{\partial v_F}{\partial x} \right)^2 + \left(\frac{\partial u_F}{\partial z} + \frac{\partial w_F}{\partial x} \right)^2 + \left(\frac{\partial w_F}{\partial y} + \frac{\partial v_F}{\partial z} \right)^2 \right\}$			
$\mu_{eff} = \mu + \mu_t \quad , \quad \mu_t = \rho_F c_\mu \frac{k^2}{\varepsilon}$			
$c_\mu = 0.09 \quad , \quad c_{\varepsilon_1} = 1.44 \quad , \quad c_{\varepsilon_2} = 1.92 \quad , \quad \sigma_k = 1.0 \quad , \quad \sigma_\varepsilon = 1.3$			

Tabelle 1: Quellterme und Transportkoeffizienten für verschiedene Variablen Φ

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 wurden [11, 13].

3.2 Die Bewegung der dispersen Phase

3.2.1 Das Lagrange-Verfahren

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 \dot{N}_P , d.h. jede Trajektorie repräsentiert einen bestimmten Anteil des gesamten Partikelmassenstroms $\dot{m}_{P,ges}$.

Die hier betrachteten Stömungen sollen Feststoffbeladungen aufweisen, bei denen die Partikelphase als verdünnt angesehen werden kann, aber einen merkbaren Einfluß auf die Fluidphase ausüben kann. Aus diesem Grund werden Partikel-Partikel-Interaktionen hier vernachlässigt. Der Einfluß der Partikel auf das Fluid wird in Kapitel 3.3 beschrieben.

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 translatorischen Bewegungsgleichungen für ein Partikel lauten damit:

$$\frac{d}{dt} \begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix} = \begin{bmatrix} u_P \\ v_P \\ w_P \end{bmatrix} \quad (2)$$

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} u_P \\ v_P \\ w_P \end{bmatrix} = & \frac{3}{4} \frac{\rho_F}{\rho_P d_P} \left(v_{rel} C_D (Re_P) \begin{bmatrix} u_F - u_P \\ v_F - v_P \\ w_F - w_P \end{bmatrix} \right. \\ & + \frac{v_{rel}}{\omega_{rel}} C_M(\sigma) \cdot \begin{bmatrix} (v_F - v_P)(\omega_z - \Omega_z) - (w_F - w_P)(\omega_y - \Omega_y) \\ (w_F - w_P)(\omega_x - \Omega_x) - (u_F - u_P)(\omega_z - \Omega_z) \\ (u_F - u_P)(\omega_y - \Omega_y) - (v_F - v_P)(\omega_x - \Omega_x) \end{bmatrix} \\ & + \frac{2\nu_F^{1/2}}{\pi |\vec{\Omega}|^{1/2}} C_A \begin{bmatrix} (v_F - v_P)\Omega_z - (w_F - w_P)\Omega_y \\ (w_F - w_P)\Omega_x - (u_F - u_P)\Omega_z \\ (u_F - u_P)\Omega_y - (v_F - v_P)\Omega_x \end{bmatrix} \\ & \left. + \frac{\rho_P - \rho_F}{\rho_P} \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \right) \quad (3) \end{aligned}$$

mit

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

$$v_{rel} = \sqrt{(u_F - u_P)^2 + (v_F - v_P)^2 + (w_F - w_P)^2}$$

$$\omega_{rel} = \sqrt{(\omega_x - \Omega_x)^2 + (\omega_y - \Omega_y)^2 + (\omega_z - \Omega_z)^2}$$

Die Gleichung für die Rotationsbewegung des Partikels lautet:

$$\frac{d}{dt} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = -\frac{15}{16\pi} \frac{\rho_F}{\rho_P} \omega_{rel} \xi_m (Re_\omega) \begin{bmatrix} \omega_x - \Omega_x \\ \omega_y - \Omega_y \\ \omega_z - \Omega_z \end{bmatrix} \quad (4)$$

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 Größe der Koeffizienten C_D , C_M , C_A und ξ_m können der Literatur entnommen werden [3].

Als Ergebnis der Berechnung der Fluidphase sind die Fluidgrößen für den Zentralpunkt jeder Zelle des numerischen Gitters bekannt. Innerhalb einer Gitterzelle sind diese Größen konstant. Da die Partikel im allgemeinen kleiner sind als die Gitterzellen, ist aus Gründen der Rechengenauigkeit eine exaktere Angabe der Fluidgrößen notwendig. Zur Berechnung der Fluidwerte am Ort des Partikels wird folgende Gradientenmethode verwendet:

$$\Phi|_P = \Phi|_C + \text{grad } \Phi|_C \cdot (\vec{r}_P - \vec{r}_C) \quad (5)$$

Hierbei bezeichnet Φ eine allgemeine skalare Variable und die Indices C bzw. P kennzeichnen Werte im Zellenmittelpunkt bzw. am Partikelort.

3.2.2 Partikel-Wand-Kollision

Die meisten für industrielle Anwendungen wichtigen dispersen Mehrphasenströmungen sind Strömungen innerhalb von Anlagen, wie z.B. Strömungen in Zyklonen oder der pneumatische Transport von granularen Materialien in Rohrleitungssystemen. Dabei wird die Bewegung größerer Partikel, die hauptsächlich durch deren Massenträgheit dominiert wird, stark beeinflusst durch die die Strömung begrenzenden Wände. Betrachtet man die Kollisionen der Partikel mit den Wänden, so wurde in der Vergangenheit gezeigt, daß die Wandrauigkeit und die Abweichung der Partikelform von einer idealen Kugel eine große Rolle spielt [3, 6, 18].

In dieser Arbeit wurde zur Simulation der Partikel-Wand-Kollisionen ein Wandrauigkeitsmodell nach Sommerfeld [16] benutzt, das leicht modifiziert wurde. In diesem Modell stößt das Teilchen mit einer um den Winkel γ angestellten virtuellen Wand. Der Inklinationswinkel γ ist eine Zufallsvariable aus einer Gauss-Verteilung mit dem Mittelwert 0° und der Standardabweichung $\Delta\gamma$. $\Delta\gamma$ ist abhängig vom Partikeldurchmesser d_P und den Rauigkeitsparametern der Wand und kann abgeschätzt werden durch:

$$\begin{aligned} \Delta\gamma &= \arctan \frac{2\Delta H_r}{L_r} & \text{für } d_P &\geq \frac{L_r}{\sin(\arctan \frac{2H_r}{L_r})} \\ \Delta\gamma &= \arctan \frac{2H_r}{L_r} & \text{für } d_P &< \frac{L_r}{\sin(\arctan \frac{2H_r}{L_r})} \end{aligned} \quad (6)$$

Hierbei ist L_r der mittlere Rauigkeitsabstand, H_r die mittlere Rauigkeitstiefe und ΔH_r die Standardabweichung der Rauigkeitstiefe. Es wird angenommen, daß die Wandrauigkeit keine bevorzugte Richtung besitzt. Deshalb wird die angestellte Wand zusätzlich um einen Azimutwinkel σ_a gedreht, wobei die Drehachse dem Normalenvektor der ursprünglichen, nicht angestellten Wand entspricht. Der Azimutwinkel ist eine gleichverteilte Zufallszahl im Intervall $[-\pi, \pi]$.

Die Partikelgeschwindigkeiten und -winkelgeschwindigkeiten werden in ein Koordinatensystem transformiert, das an der Kollisionsebene orientiert ist. Für die folgenden Gleichungen soll die y -Achse des transformierten Koordinatensystems identisch mit dem Normalenvektor der Kollisionsebene sein. Aus den Impuls- bzw. Drehimpulsgleichungen ergeben sich die Geschwindigkeiten und Winkelgeschwindigkeiten nach dem Stoß. Dabei wird unterschieden, ob das Partikel während des gesamten Stoßvorgangs auf der Wand gleitet (Gleitstoß) oder nicht (Haftstoß) [18]:

1. Gleitstoß für : $-\frac{2}{7f(e+1)} \leq \frac{v_p^{(1)}}{|v_r|} \leq 0$:

$$\begin{aligned} u_p^{(2)} &= u_p^{(1)} + \epsilon_x f (e + 1) v_p^{(1)} \quad , \quad v_p^{(2)} = -e v_p^{(1)} \quad , \\ w_p^{(2)} &= w_p^{(1)} + \epsilon_z f (e + 1) v_p^{(1)} \quad , \\ \omega_x^{(2)} &= \omega_x^{(1)} - \frac{5}{d_p} \epsilon_z f (e + 1) v_p^{(1)} \quad , \quad \omega_y^{(2)} = \omega_y^{(1)} \quad , \\ \omega_z^{(2)} &= \omega_z^{(1)} + \frac{5}{d_p} \epsilon_x f (e + 1) v_p^{(1)} \end{aligned} \quad (7)$$

2. Haftstoß für : $\frac{v_p^{(1)}}{|v_r|} < -\frac{2}{7f(\epsilon+1)} :$

$$\begin{aligned} u_p^{(2)} &= \frac{5}{7} (u_p^{(1)} - \frac{d_p}{5} \omega_z^{(1)}) \quad , \quad v_p^{(2)} = -\epsilon v_p^{(1)} \quad , \\ w_p^{(2)} &= \frac{5}{7} (w_p^{(1)} + \frac{d_p}{5} \omega_x^{(1)}) \quad , \\ \omega_x^{(2)} &= \frac{2}{d_p} w_p^{(1)} \quad , \quad \omega_y^{(2)} = \omega_y^{(1)} \quad , \quad \omega_z^{(2)} = -\frac{2}{d_p} u_p^{(1)} \end{aligned} \quad (8)$$

mit

$$|v_r| = \sqrt{(u_p^{(1)} + \frac{d_p}{2} \omega_z^{(1)})^2 + (w_p^{(1)} - \frac{d_p}{2} \omega_x^{(1)})^2}$$

und

$$\epsilon_x = \frac{u_p^{(1)} + \frac{d_p}{2} \omega_z^{(1)}}{|v_r|} \quad , \quad \epsilon_z = \frac{w_p^{(1)} - \frac{d_p}{2} \omega_x^{(1)}}{|v_r|}$$

In diesen Gleichungen ist ϵ der Elastizitätskoeffizient und f der Reibungsbeiwert. Die Superscripte (1) und (2) kennzeichnen Werte vor bzw. nach dem Stoß.

3.3 Phasenkopplung und Berücksichtigung der Fluidturbulenz

3.3.1 Phasenkopplung

Zur Modellierung des Einflusses der dispersen Phase auf das Fluid wird die sogenannte Particle-Source-In-Cell (PSI-Cell) Methode verwendet [1, 2]. Hierbei werden zusätzliche Quellterme $S_{u_i}^P$ in die Impulsgleichungen für das Fluid eingeführt (siehe Gl. (1) und Tab. 1). Die Quellterme werden während der Trajektorienberechnung für jede Gitterzelle bestimmt. Der Quellterm für die x -Impulsgleichung ergibt in folgender Weise:

$$S_{u_F}^P = -\frac{1}{V_{ijk}} \sum m_P \dot{N}_P \left[u_{P,out} - u_{P,in} - g_x \left(1 - \frac{\rho_F}{\rho_P}\right) (t_{out} - t_{in}) \right] \quad (9)$$

Hierin ist V_{ijk} das Volumen der aktuellen Gitterzelle, m_P die Masse des Partikels und g_x die Komponente der Gravitationskraft in x -Richtung. Die Indices *in* und *out* bezeichnen Größen beim Eintritt des Partikels in die Gitterzelle bzw. beim Austritt aus der Zelle. Die entsprechenden Quellterme für die y - und z -Impulsgleichungen ergeben sich analog.

Zur Berechnung der phasengekoppelten Strömung wird das folgende iterative Lösungsverfahren verwendet:

1. Berechnung einer konvergenten Lösung für die Fluidströmung ohne Berücksichtigung der dispersen Phase.
2. Berechnung einer großen Anzahl von Partikeltrajektorien im Strömungsgebiet und gleichzeitige Berechnung der Quellterme für jede Zelle des numerischen Gitters.
3. Erneute Berechnung der Fluidströmung unter Einbeziehung der Quellterme der dispersen Phase.
4. Wiederholung der Schritte 2. und 3. bis zur Konvergenz.

3.3.2 Berücksichtigung der Fluidturbulenz

Im Gegensatz zur Bewegung großer Partikel wird die Bewegung kleiner Partikel stark von der Fluidströmung beeinflusst. Insbesondere für die Dispersion kleiner Partikel in der Strömung spielt die Turbulenz des Fluides eine große Rolle. Zur Modellierung des Einflusses der Turbulenz auf die Partikelbewegung wird hier das sogenannte Lagrangian-Stochastic-Deterministic (LSD) Turbulenzmodell nach Schönung [15] und Milojević [7] verwendet. Hierzu werden in Gleichung (3) die zeitlich gemittelten Fluidgeschwindigkeiten u_{iF} durch die Momentangeschwindigkeiten U_{iF} ersetzt, die sich aus der Summe der mittleren Geschwindigkeiten und der Schwankungsgeschwindigkeiten u'_{iF} ergeben:

$$U_F = u_F + u'_F \quad , \quad V_F = v_F + v'_F \quad , \quad W_F = w_F + w'_F$$

Unter der Annahme isotroper Turbulenz werden die Schwankungsgeschwindigkeiten als Zufallszahl aus einer Gauss-Verteilung mit dem Mittelwert 0 und der Standardabweichung σ generiert. σ ergibt sich aus der Turbulenzenergie k zu:

$$\sigma = \sqrt{\frac{2}{3}k} \quad (10)$$

Mit den daraus resultierenden Momentangeschwindigkeiten werden die Bewegungsgleichungen für das Partikel gelöst. Dabei ist die Größe des Zeitschritts Δt , um den sich das Partikel weiterbewegt, begrenzt durch die Lebensdauer des Turbulenzwirbels T_L und die Zeit, die das Partikel zur Durchquerung des Wirbels benötigt. Zur Überprüfung der ersten Bedingung wird aus einer Gleichverteilung eine Zufallszahl $\alpha \in [0, 1]$ generiert. Falls

$$\alpha < \frac{\Delta t}{T_L} \quad \text{mit} \quad T_L = 0.3 \frac{k}{\varepsilon} \quad (11)$$

erfüllt ist, werden neue Schwankungsgeschwindigkeiten u'_{iF} berechnet.

Die Zeit für die Durchquerung des Wirbels ist abhängig vom Längenmaß des Wirbels L_E und der Relativgeschwindigkeit zwischen Partikel und Fluid. Der Wirbel gilt als durchquert, wenn die vom Partikel in der Zeit Δt zurückgelegte Strecke Δl größer ist als das Wirbellängenmaß:

$$\Delta l > L_E \quad \text{mit} \quad L_E = 0.245 \frac{k^{3/2}}{\varepsilon} \quad , \quad \Delta l = \Delta t v_{rel}$$

In diesem Fall wird angenommen, daß das Partikel in einen neuen Wirbel eintritt und es werden neue Schwankungsgeschwindigkeiten bestimmt.

4 Parallelisierung des Berechnungsverfahrens

Wie in Kapitel 3 dargestellt wurde, besteht die Berechnung einer dispersen Mehrphasenströmung aus der gekoppelten Berechnung der Bewegung der fluiden und der dispersen Phase. Zur Parallelisierung des dort beschriebenen Berechnungsverfahrens gehört deshalb sowohl die Parallelisierung der Fluidberechnung als auch die Parallelisierung der Berechnung der Partikeltrajektorien. In Kapitel 4.1 wird zunächst das parallele Verfahren zur Simulation der Fluidströmung beschrieben. Zur parallelen Berechnung der Trajektorien wurden im Rahmen dieses Projektes drei verschiedene Algorithmen entwickelt. Diese Algorithmen werden in Kapitel 4.2 vorgestellt.

4.1 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. 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.

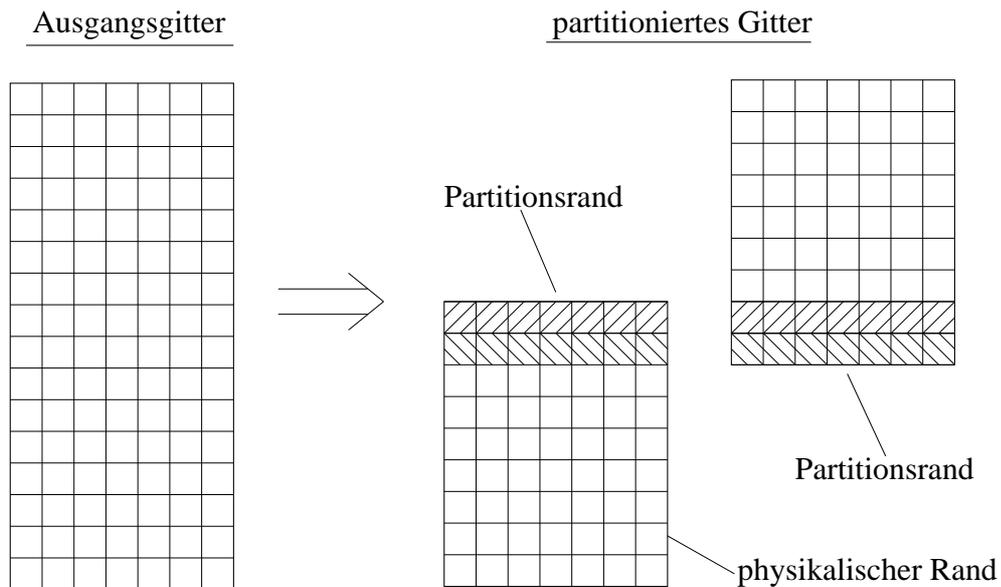


Abbildung 1: Prinzip der Gitterpartitionierung

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 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. 1 gezeigt. Dadurch können die Strömungsgrößen für die unmittelbar 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.

4.2 Die verschiedenen Parallelisierungsalgorithmen für das Lagrange-Verfahren

Bei der Parallelisierung des Berechnungsverfahrens für die Bewegung der dispersen Phase ist die komplexe Abhängigkeit zwischen den Daten der Fluidströmung und den Datenanforderungen des Lagrange-Lösers zu beachten. Für die Berechnung einer Trajektorie werden die Geometriedaten des Gitterblocks benötigt, in dem sich das Partikel gerade befindet. Dies ist notwendig, um die Bahn des Partikels in der Geometrie zu verfolgen und das Stoßen gegen Wände, das Verlassen des Blockes usw. detektieren zu können. Des weiteren müssen die Strömungsgrößen für die Gitterzelle, in der sich das Partikel gerade aufhält, bekannt sein. Unter Beachtung dieser Notwendigkeiten wurden drei parallele Rechenverfahren entwickelt, die in den Kapiteln 4.2.1 – 4.2.3 vorgestellt werden. Sie unterscheiden sich im wesentlichen durch das Speichermanagement, d.h. durch die Weise, wie die zu speichernden Daten unter den Prozessoren verteilt sind, und die sich daraus ergebende Verteilung der Rechenlast.

4.2.1 Das Quasiserielle (QS) Verfahren

In der QS-Methode werden zwei verschiedene Prozeßklassen verwendet (siehe Abb. 2). Der Host-Prozeß verwaltet die Partikelstartbedingungen (Startposition, -geschwindigkeit, usw.), führt aber selber keine Trajektorienberechnungen aus. Diese werden von den Node-Prozessen durchgeführt. Im lokalen Speicher jedes Node-Prozesses sind die Gitter- und Strömungsdaten aller Gitterblöcke des Strömungsgebiets permanent gespeichert. Somit kann jeder Node-Prozeß Trajektorien im gesamten Strömungsgebiet berechnen.

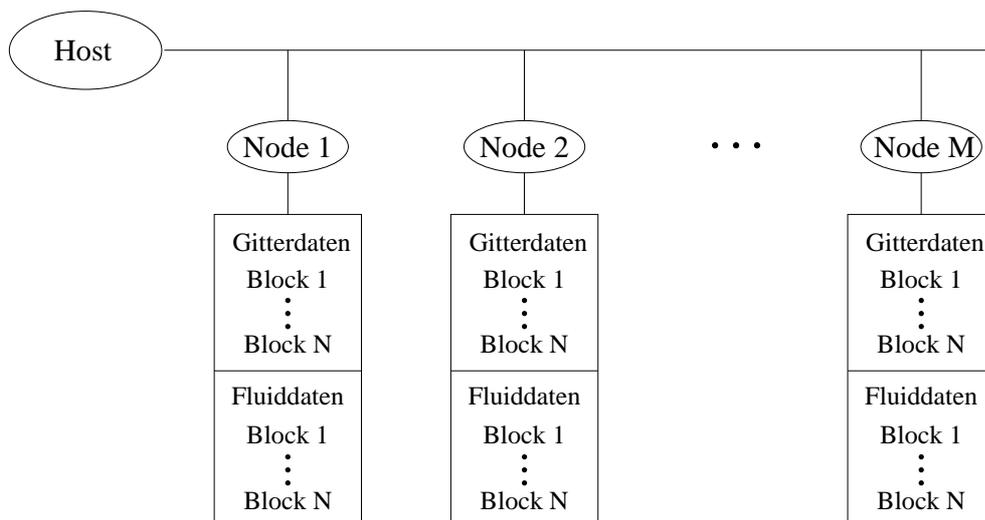


Abbildung 2: Prinzip des Quasiseriellen Verfahrens

Zu Beginn der Rechnung verteilt der Host-Prozeß die Startbedingungen an die Node-Prozesse, so daß jeder Node-Prozeß die gleiche Anzahl Startbedingungen erhält. Die Node-Prozesse be-

rechnen die Partikeltrajektorien bis zum Erreichen einer Abbruchbedingung (Verlassen des Strömungsgebietes, Haften an einer Wand, usw.). Gleichzeitig werden die Quelltermwerte für die Phasenkopplung berechnet. Wenn alle Node-Prozesse ihre Berechnungen beendet haben, senden sie die Quelltermdaten an den Host, der für jede Gitterzelle die einzelnen Beiträge aufsummiert.

Durch die gleichmäßige Verteilung der Partikelstartbedingungen unter den Prozessoren wird beim QS-Verfahren automatisch eine gleichmäßige Verteilung der Rechenlast erreicht. Ein weiterer Vorteil dieses Verfahrens ist, daß die Anzahl der verwendeten Prozessoren unabhängig von der Anzahl der Blöcke des numerischen Gitters ist. Somit kann die Prozessorzahl optimal den zur Verfügung stehenden Rechnerressourcen angepaßt werden. Der Kommunikationsaufwand bei dieser Methode ist gering. Er beschränkt sich auf das Senden der Startbedingungen vom Host an die Nodes am Anfang der Berechnung und das Senden der Quelltermwerte von den Nodes an den Host am Ende der Berechnung. Ein Datenaustausch zwischen den Nodes ist nicht erforderlich. Im Gegensatz zum Kommunikationsaufwand ist der Speicheraufwand relativ groß, da jeder Prozessor lokal die Gittergeometrie- und Strömungsdaten für das gesamte Strömungsgebiet speichern muß. Deshalb ist das QS-Verfahren wenig geeignet für die Anwendung auf massiv parallelen Rechnern mit verteiltem Speicher, da die lokale Speicherkapazität der einzelnen Prozessoren dieser Rechner im allgemeinen relativ klein ist. Es bieten sich jedoch gute Anwendungsmöglichkeiten auf Workstation-Clustern, da die Workstations in der Regel über einen ausreichend großen Arbeitsspeicher verfügen.

4.2.2 Das Domain Decomposition (DD) Verfahren

Um das Lagrange-Verfahren auch auf massiv parallelen Rechnern mit ihrer beschränkten Speichergröße verwenden zu können, wurde das sogenannte Domain Decomposition Verfahren entwickelt (Abb. 3). Das Prinzip dieser Parallelisierungsmethode entspricht dem Prinzip des gleichnamigen Verfahrens zur Berechnung der Fluidströmung. Analog zum QS-Verfahren gibt es einen 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.

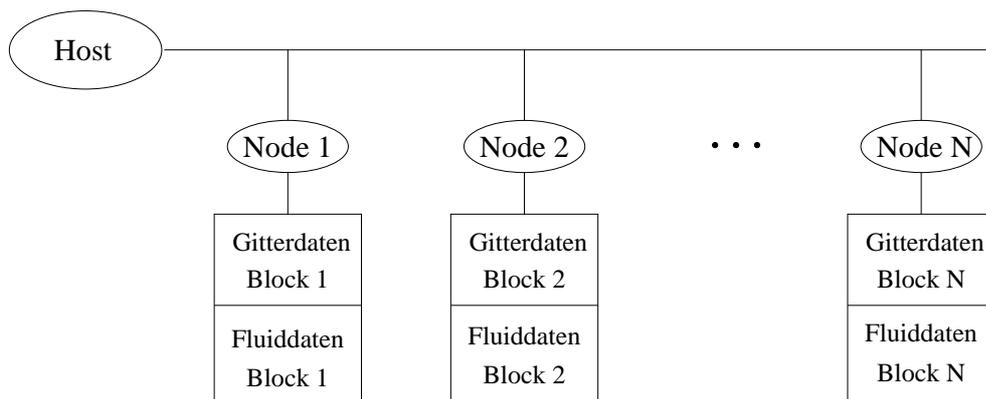


Abbildung 3: Prinzip des Domain Decomposition Verfahrens

Wiederum verteilt der Host zu Beginn der Rechnung die Partikelstartbedingungen an die Node-Prozesse. Allerdings werden diese nicht gleichmäßig verteilt, sondern entsprechend den Startpositionen der Partikel. 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 an den Node-Prozeß gesendet, der den benachbarten Block bearbeitet. 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. Da bei der DD-Methode alle in einem Block anfallenden Quelltermbeiträge von einem einzigen Node-Prozeß berechnet werden, entfällt das bei der QS-Methode notwendige Aufsummieren dieser Werte nach dem Ende der Trajektorienberechnung.

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. Der Kommunikationsaufwand ist höher als beim QS-Verfahren, da zusätzlich Daten zwischen den Nodes ausgetauscht werden. Ein Nachteil des DD-Verfahrens ist, daß eine sehr 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 Freistahlströmungen 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.

4.2.3 Das Distributed Shared Memory (DSM) Verfahren

Die in Kapitel 4.2.2 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. 4). Im Vergleich zu den beiden 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 bei den beiden anderen Verfahren hat der Host-Prozeß die Aufgabe, die Partikelstartbedingungen zu verwalten und an die Nodes zu verteilen.

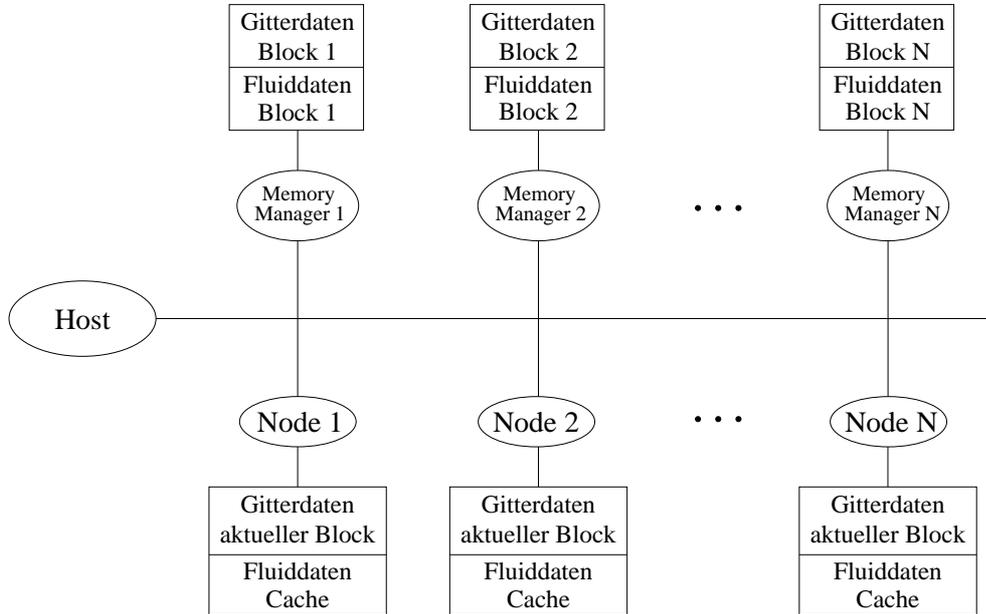


Abbildung 4: Prinzip des Distributed Shared Memory Verfahrens

Im folgenden seien die Aufgaben der einzelnen Prozesse näher beschrieben:

Host:

Der Host verteilt die Partikelstartbedingungen an die Nodes. Dabei ist eine Startbedingung nicht nur der Partikelstatus zu Beginn der Rechnung, 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 Partitions-grenze 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.

Node:

Der Node empfängt jeweils eine Partikelstartbedingung vom Host, berechnet das 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 Partikeldata, 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.

5 Untersuchungen der numerischen und parallelen Effizienz

5.1 Vergleichende Untersuchungen der eingesetzten Parallelisierungsverfahren auf dem Parsytec GC-128

Um einen Vergleich der unter 4.2 beschriebenen parallelen Lösungsverfahren zu ermöglichen, wurden in einer ersten Testserie Berechnungen für zwei typische Testfälle auf dem an der TU Chemnitz-Zwickau verfügbaren massiv-parallelen Rechnersystem durchgeführt.

5.1.1 Die vertikale Kanalströmung

Als ein erster Testfall wurde eine 2-dimensionale, abwärtsgerichtete Kanalströmung eines Gas-Feststoff-Gemisches betrachtet. Das Verhältnis von Kanalbreite zu Kanallänge betrug $H/L = 1/6$. Für die fluide Phase wurde Luft unter Normalbedingungen angenommen. Die disperse Phase bestand aus Feststoffpartikeln mit einem Dichteverhältnis von $\rho_F/\rho_P = 1/2500$ und einer gleichverteilten Partikelgrößenverteilung von $d_P = 100 \dots 300 \mu m$. Die Einströmgeschwindigkeit für das Fluid und die disperse Phase betrug $2 m/s$. Im Eintrittsquerschnitt wurde ein homogenes Fluidgeschwindigkeitsprofil und eine homogene Partikelkonzentrationsverteilung angenommen.

Die Strömungsgeometrie wurde mit einem numerischen Gitternetz diskretisiert, das auf dem groben Gitternetz insgesamt 128×32 CV's und auf dem feinen Gitternetz 256×64 CV's aufwies. Für die parallele Berechnung der Mehrphasenströmung wurde das Gitternetz in $1 \dots 128$ Gitterblöcke zerteilt, ohne die Gesamtanzahl der CV's zu verändern.

5.1.2 Die Umströmung einer Vollkegeldüse

Mit dem zweiten Testfall wurde die Absicht verfolgt, die Abhängigkeit der Effizienz der parallelen Lösungsverfahren von der Art des Strömungsregimes zu untersuchen. Als ein Beispiel einer stark separierten Zweiphasenströmung wurde eine axialsymmetrische, entgegen der

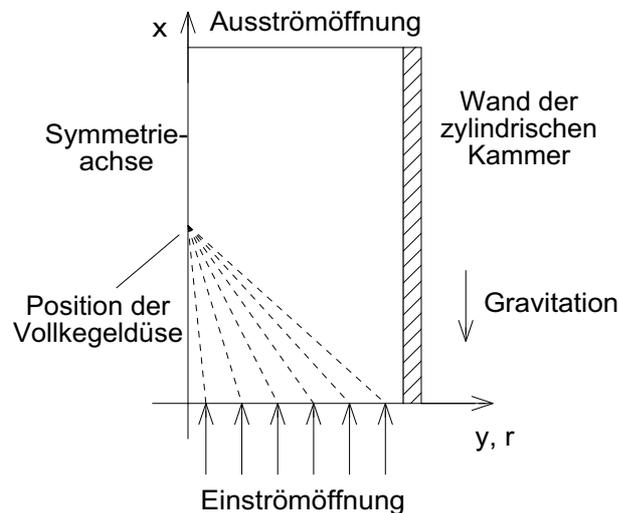


Abbildung 5: Axialsymmetrische Umströmung einer Vollkegeldüse im Gegenstrom

Schwerkraft gerichtete Umströmung einer auf der Symmetrieachse eines Zylinders angeordneten Vollkegeldüse gewählt (Abb. 5). Die von der Vollkegeldüse mit einem Öffnungswinkel von 90° freigesetzte Tropfenströmung ist dem Gasstrom entgegen, d.h. in Richtung der Schwerkraft gerichtet. Das Verhältnis von Zylinderradius zu -höhe betrug $R/H = 1/6$. Die Position der Düse

auf der Symmetrieachse lag bei $x = \frac{1}{3}H$. Für die fluide Phase wurde wiederum Luft unter Normalbedingungen angenommen. Die Einströmgeschwindigkeit der Luft betrug $u_F = 4\text{ m/s}$ mit einem Turbulenzgrad von 10 %. Die disperse Phase wurde durch Wassertropfen repräsentiert, die eine Anfangsgeschwindigkeit von 8 m/s und eine vorgegebene Normalverteilung der Partikelgröße im Bereich von $30\text{ }\mu\text{m} \dots 1400\text{ }\mu\text{m}$ aufwiesen. Die Anzahl der CV's, Gitterblöcke und Verfeinerungsniveaus des numerischen Gitters war mit denen aus dem ersten Testfall identisch.

5.1.3 Das MIMD-Parallelrechnersystem

Für die Testrechnungen wurde das Parallelrechnersystem der TU Chemnitz-Zwickau vom Typ Parsytec GC-128 verwendet. Die 64 Knoten dieses MIMD-Rechnersystems sind aus je 2 mit 80 MHz getakteten Motorola PowerPC-Prozessoren 601 und 4 Kommunikations-Coprozessoren vom Typ IMS T805-30 aufgebaut. Jeder dieser Knoten verfügt über einen lokalen Hauptspeicher von 32 Megabyte. Die Kommunikations- und Ein-/Ausgabe-Bandbreite der Knoten wird mit 35 Mb/s angegeben.

Die Maschine läuft unter dem Betriebssystem Parix 1.2-PPC (einem UNIX-Derivat). Für die Programmentwicklung standen die Parallelisierungsstandards PVM 3.2 und MPI 1.0 zur Verfügung.

5.1.4 Vergleich der parallelen Effizienz der drei Parallelisierungsverfahren

Bei den Testrechnungen wurden die drei unter 4.2 dargestellten Parallelisierungsverfahren auf beide Testfälle angewendet. Es wurde jeweils die Ausführungszeit T_A eines Iterationszyklus des Lagrange-Lösungsverfahrens ermittelt, wobei die Startzeit des PVM-Systems, die Ausführungszeit des Navier-Stokes-Lösers und die zur Ein-/Ausgabe benötigte Zeit bei den Messungen unberücksichtigt blieb. In den Testrechnungen wurden in allen Fällen jeweils 20000 Partikeltrajektorien berechnet, wie es zur Ermittlung konvergenter Lösungen für eine phasengekoppelte Mehrphasenströmung erforderlich ist. Aus den so ermittelten Ausführungszeiten wurde der Speed-up S_N und die parallele Effizienz E_N wie folgt berechnet :

$$S_N = \frac{T_A \text{ auf einem Prozessor}}{T_A \text{ auf } N \text{ Prozessoren}} \quad ; \quad E_N = \frac{S_N}{N}$$

Wie aus den in Abb. 6-11 dargestellten Ergebnissen hervorgeht, spiegeln die so ermittelten Effizienzen E_N nicht den reinen Gewinn an paralleler Effizienz wieder. Vielmehr geht in die Effizienz E_N ebenfalls der Gewinn an numerischer Effizienz ein, der dadurch gegeben ist, daß bestimmte Teilalgorithmen des Lagrange'schen Lösungsverfahrens von dem blockstrukturierten Gitternetz profitieren können und auf numerischen Gitternetzen mit einer höheren Anzahl von Gitterblöcken im numerischen Sinne effizienter abgearbeitet werden können. Dies traf insbesondere auf den in allen drei Verfahren implementierten Algorithmus zur Partikellokalisierung auf dem numerischen Gitternetz zu. Die Summe aus numerischer und paralleler Effizienzsteigerung führt zu den in den Abb. 8 und 11 zu beobachtenden Werten für $E_N > 1$. Da jedoch die Algorithmen, die zu der besagten numerischen Effizienzsteigerung beitragen, in allen drei Parallelisierungsverfahren gleichermaßen enthalten sind, können die ermittelten Testergebnisse trotzdem ohne Einschränkung zu einem Vergleich der drei Verfahren untereinander herangezogen werden.

Abb. 6-8 zeigen die Ergebnisse für die Testrechnungen für die vertikale Kanalströmung auf 1 bis 128 Prozessoren. Das beste Resultat für diese Strömungskonfiguration konnte mit einem Speed-up von 47.1 auf 64 Prozessoren bei Verwendung des quasiseriellen (QS-) Verfahrens erzielt werden. Wie jedoch bereits erwähnt wurde, handelt es sich bei dem QS-Verfahren um den seriellen Lagrange-Löser, der auf jedem der Knoten des Parallelrechners auf der Gesamtmenge der Fluidströmungsdaten abläuft. Das QS-Verfahren unterstützt damit nicht die über die Knoten des MIMD-Systems verteilte Speicherung der Geometrie- und Strömungsdaten und ist somit nicht

anwendbar auf feiner auflösende numerische Gitternetze oder Strömungen in geometrisch komplexeren Strömungsgeometrien. Die hierbei für das QS-Verfahren benötigten Hauptspeicherressourcen würden die lokalen Ressourcen der Knoten eines MIMD-Systems schnell übersteigen. Insofern kann das QS-Verfahren nicht als ein vollständig parallelisiertes Verfahren angesehen werden, obgleich es auf Systemen mit ausreichenden Speicherressourcen (z.B. gut ausgestatteten Workstation-Clustern mit einer geringen Anzahl von Knoten) sehr effizient einzusetzen ist.

Das Domain Decomposition (DD-) Verfahren und das Distributed Shared Memory (DSM-) Verfahren zeigen für den Fall der vertikalen Kanalströmung vergleichbare Ergebnisse mit einem maximalen Speed-up von ca. 18 und einer Effizienz von 0.3 auf 64 Prozessoren. Der starke Leistungsabfall aller drei Verfahren beim Übergang von 64 auf 128 Prozessoren ist nicht auf die Implementierung der Parallelisierungsverfahren zurückzuführen, sondern resultiert aus einer bekannten Beschränkung in der Knoten-Knoten-Kommunikation zwischen der unteren und der oberen 64-Prozessor-Partition des Parsytec GC-128. Der sprunghafte Leistungsanstieg aller drei Parallelisierungsverfahren beim Übergang von 8 auf 16 Prozessoren ist auf die Aufteilung des Einstromrandes ($x = 0$) auf zwei verschiedene Gitterblöcke bei der Vernetzung der Strömungsgebiets mit 16 Gitterblöcken zurückzuführen. Dies führt zu einer Verteilung der Partikelanfangsbedingungen auf zwei Prozessoren und somit zu einer höheren Effizienz bei der Lokalisierung der Anfangsbedingungen auf dem numerischen Gitternetz.

Abb. 9–11 zeigen die Ergebnisse für die Umströmung der Vollkegeldüse. Die für den ersten Testfall diskutierten Besonderheiten für die Testrechnungen auf 128 Prozessoren und für den Effizienzgewinn beim Übergang von 8 auf 16 Prozessoren lassen sich auch in den Ergebnissen des zweiten Testfalls wiederfinden. Auch kann das beste Ergebnis wiederum mit dem quaiseriiellen Verfahren erzielt werden.

Im Unterschied zu den Berechnungen für die vertikale Kanalströmung und auch im Unterschied zu den anderen beiden Parallelisierungsverfahren zeigt das DD-Verfahren jedoch starke Leistungseinbußen (Abb. 11). Dieses Verhalten des DD-Verfahrens ist auf die schlechte Lastverteilung zurückzuführen, die durch die stark separierte Gas-Tropfen-Strömung in diesem Testfall hervorgerufen wird. Im DD-Verfahren (siehe auch 4.2.2) ist die Rechenleistung eines Prozessors fest mit einem der Gitterblöcke des numerischen Gitternetzes verbunden. Tritt nun wie im vorliegenden Testfall eine starke Entmischung der kontinuierlichen und der dispersen Phase auf, so ist die Partikelkonzentration im Strömungsgebiet und damit der numerische Berechnungsaufwand auf den einzelnen Gitterblöcken sehr unterschiedlich. Die Ausführungszeit des Lagrange-Lösers wird nun im Wesentlichen durch die Berechnungszeit für die Bewegung der dispersen Phase auf dem Gitterblock mit der höchsten Partikelkonzentration bestimmt.

Im Gegensatz dazu zeigen die Ergebnisse für das DSM-Verfahren eine mit dem QS-Verfahren vergleichbare Effizienz. Erst mit stark zunehmender Prozessorzahl macht sich das höhere Kommunikationsvolumen im DSM-Verfahren durch ein Absinken der Effizienz bemerkbar (Abb. 11). Die Ergebnisse zeigen jedoch vor allem, daß das DSM-Verfahren weitgehend unabhängig vom Strömungsregime der untersuchten Mehrphasenströmung einsetzbar ist. Die dynamische Zuordnung der Gitterblöcke des numerischen Gitters zu den Prozessoren des Parallelrechnersystems führt zu einer gleichmäßigen, dynamischen Lastverteilung über alle rechnenden Knoten des Systems und dadurch zu einer zufriedenstellenden Gesamtleistung des Verfahrens.

5.2 Vergleich der PVM- und MPI-Implementierungen auf Cray T3D und Parsytec GC-128

Aufbauend auf den unter 5.1 dargestellten Ergebnissen der ersten Tests der drei grundlegenden Parallelisierungsverfahren wurden weitere Veränderungen und Optimierungen vorgenommen, die sowohl das numerische Lösungsverfahren des Lagrange-Lösers als auch die Implementierung des

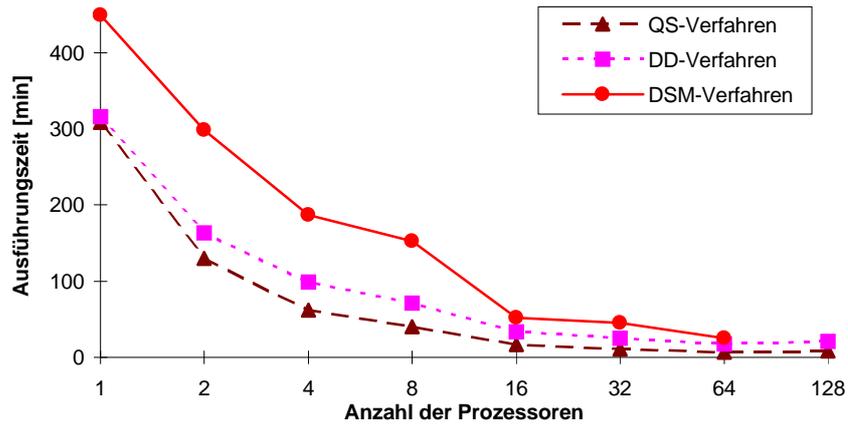


Abbildung 6: Ausführungszeiten T_A für die vertikale Kanalströmung.

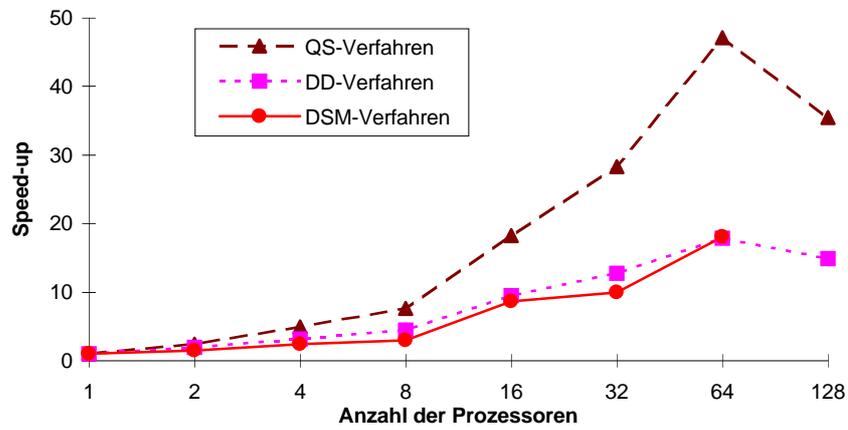


Abbildung 7: Speed-up S_N für die vertikale Kanalströmung.

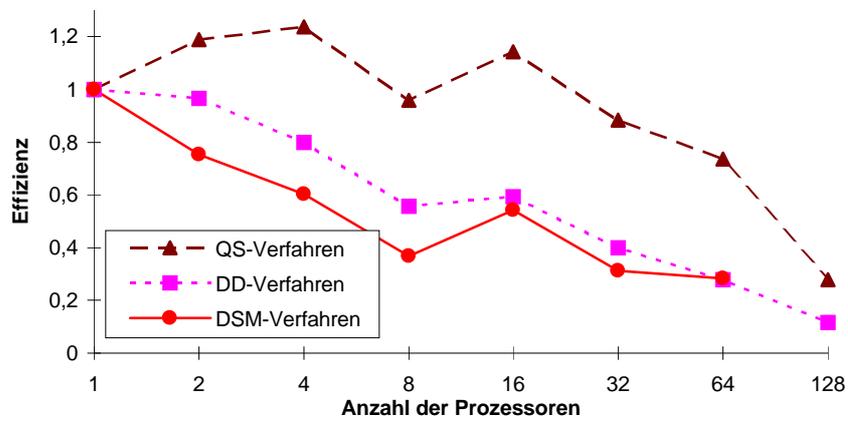


Abbildung 8: Effizienz E_N für die vertikale Kanalströmung.

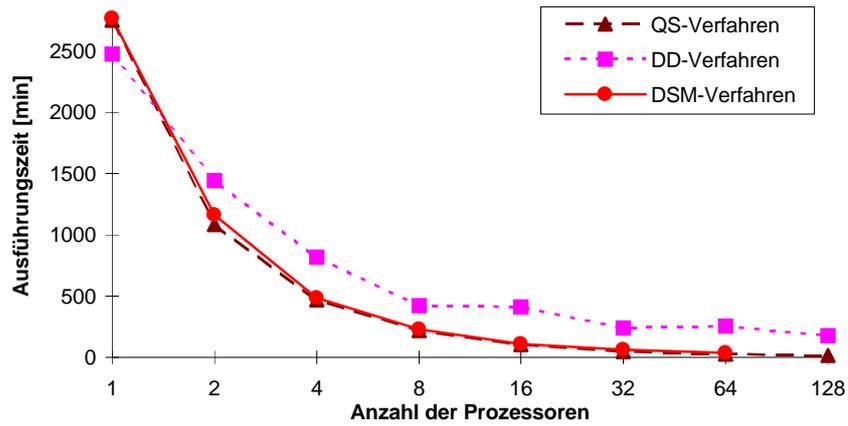


Abbildung 9: Ausführungszeiten T_A für die Umströmung der Vollkegeldüse.

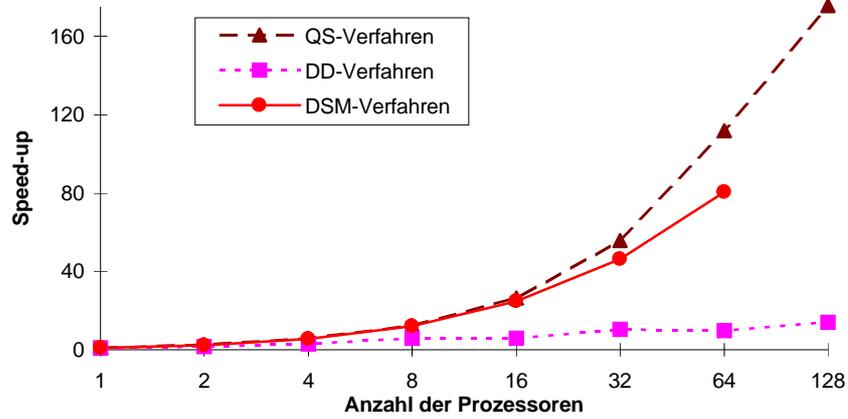


Abbildung 10: Speed-up S_N für die Umströmung der Vollkegeldüse.

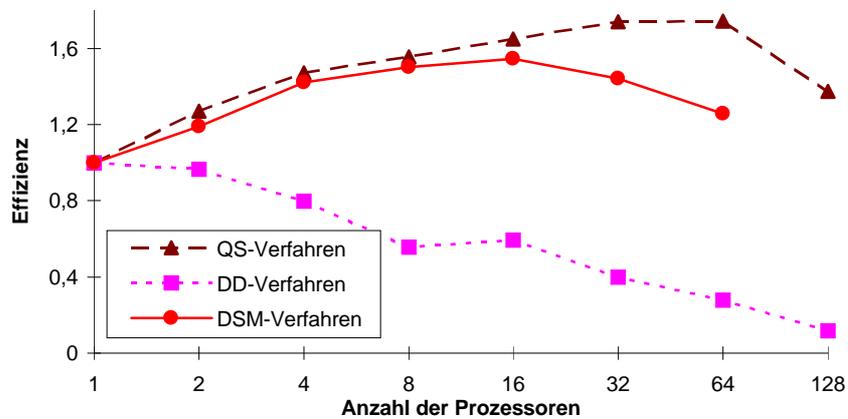


Abbildung 11: Effizienz E_N für die Umströmung der Vollkegeldüse.

parallelen Lösungsverfahrens betreffen. Die Veränderungen umfassen im Einzelnen :

1. Optimierung des Verfahrens zur Lokalisierung der Partikel- Anfangsbedingungen auf dem numerischen Gitternetz.

Unter 5.1 wurden die Anfangsbedingungen für die Partikelphase lokalisiert, indem auf dem betreffenden Gitterblock alle CV's daraufhin untersucht wurden, ob die gesuchte Partikelposition sich innerhalb des jeweiligen CV's befindet. Nunmehr kommt das unter 6.2 beschriebene Strahlverfahren aus [14] zum Einsatz. Der betreffende Gitterblock wird einer fortschreitenden Teilung unterworfen. Erst bei Erreichen einer kritischen CV-Anzahl werden die verbleibenden Gitterzellen einzeln auf die Lage der Partikelposition hin untersucht, da für kleine CV-Anzahlen die Strahlmethode uneffizient wird.

2. Veränderung des Algorithmus zur Partikelverfolgung auf dem numerischen Gitternetz während der Lösung der Bewegungsgleichungen für die disperse Phase durch den Runge-Kutta Löser (Zeitschrittverfahren).

Kam unter 5.1 noch der in [4, 5] beschriebene Algorithmus der "zirkulären Suche" zum Einsatz, bei dem für jeden neu berechneten Zeitschritt des Runge-Kutta-Lösungsverfahrens die benachbarten Gitterzellen auf die Lage der neu berechneten Partikelposition hin untersucht werden mußten, so wird nun in einem direkten Verfahren das mit der neu berechneten Partikelposition korrespondierende CV ermittelt. Dies geschieht, indem alle CV's durch Schnittpunktberechnungen ermittelt werden, die zwischen der alten und der neuen Partikelposition auf dem numerischen Gitter liegen.

3. Einbettung der Berechnung der Quellterm- und Mittelwertanteile für die Phasenkopplung und das Postprocessing in den Lösungsalgorithmus.

Da durch das jetzt eingesetzte Verfahren der Partikelverfolgung auf dem numerischen Gitter alle CV's bekannt sind, die bei einem Runge-Kutta Zeitschritt tangiert werden, ist es möglich, die so gewonnenen Geometrieinformationen für die sofortige Berechnung der Quellterm- und Mittelwertsanteile zu verwenden. Dadurch kann der numerische Aufwand des Verfahrens wesentlich reduziert werden.

4. Übertragung der Arbeitsverwaltung und Lastverteilungsoptimierung an das Host-Programm des Lagrange-Lösers.

Sowohl für das DD-Verfahren als auch für das DSM-Verfahren wird die Berechnung einer Partikeltrajektorie untergliedert in die Berechnung der einzelnen Trajektoriensegmente, die mit den tangierten Gitterblöcken korrespondieren. Nach erfolgter Berechnung eines Partikeltrajektoriensegments für einen Gitterblock wird der Partikelzustand an das Hostprogramm des Lagrange-Lösers zurückgegeben. Insbesondere im DSM-Verfahren ist nun das Hostprogramm mit geeigneten Strategien in der Lage, eine bessere Lastverteilung innerhalb der Parallelrechnerpartition zu erreichen.

5. Zusammenfassung notwendiger Kommunikation zu einer minimalen Anzahl von Kommunikationsvorgängen, wobei alle notwendigen Daten in möglichst wenigen, komplexen Datenstrukturen übertragen werden.

6. Überarbeitung der Kommunikationszwischenschicht.

Zusätzlich zu der bereits vorhandenen, zu PVM 3.2 konformen Implementierung der Kommunikationsroutinen in der vom Verfahren verwendeten Kommunikationszwischenschicht, wurde eine auf MPI 1.1 basierende Implementierung geschaffen. Durch das Setzen logischer Schalter und bedingte Compilierung ist es auf diese Weise möglich, das Programmsystem

mit jedem der beiden am häufigsten verwendeten Parallelisierungsstandards einsetzen zu können. Die Portabilität des Programmsystems auf andere Parallelrechnersysteme konnte durch diese MPI-Implementierung deutlich verbessert werden.

7. Überarbeitung der Zeitmessung.

Durch die Auflösung der absoluten Abarbeitungszeit des Verfahrens in Rechenzeit, Kommunikationszeit, Ein-/Ausgabezeit und Wartezeit wird eine detailliertere Einschätzung der eingesetzten Parallelisierungsverfahren möglich.

Mit dem auf diese Weise modifizierten, 2-dimensionalen Berechnungsverfahren wurde eine weitere Testserie durchgeführt, dessen Ergebnisse im Folgenden diskutiert werden.

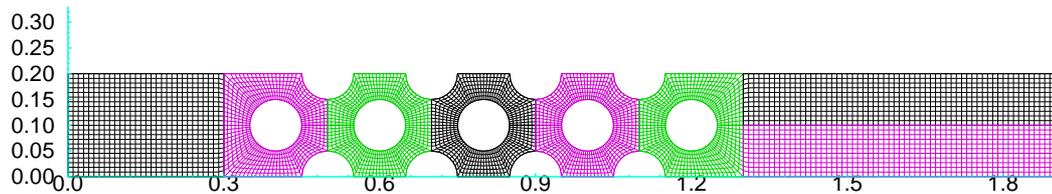


Abbildung 12: Numerisches Gitter für die Strömung durch die gegeneinander verschobenen Rohrreihen; Vernetzung mit 8 Gitterblöcken.

5.2.1 Gas-Feststoff-Strömung durch eine Rohrbündelreihe

Als Testfall wurde die Strömung eines Gas-Feststoff-Gemisches durch eine Reihe zueinander versetzt angeordneter Rohre ausgewählt. Diese Geometrieordnung tritt z.B. in Rohrbündelwärmeübertragern auf, wobei insbesondere Erosion und Anhaftungserscheinungen an den einzelnen Rohrreihen von Interesse sind. Abb. 12 zeigt die im Testfall verwendete Strömungsgeometrie mit 5 hintereinander und versetzt zueinander angeordneten Rohrreihen. Die Strömungsgeometrie weist eine Einströmöffnung auf der linken Seite, Symmetrierandbedingungen am oberen und unteren Geometrierand und eine Ausströmöffnung an der rechten Seite auf. Haftbedingungen sind für die Oberflächen der umströmten Rohre definiert. Für die Tests wurden Vernetzungen der Strömungsgeometrie mit 4 bis 64 Gitterblöcken verwendet. Für jede Vernetzung der Strömungsgeometrie wurden zwei Gitterverfeinerungsstufen verwendet. Abb. 12 zeigt das größere der beiden Gitternetze.

Da sich aus Gründen der Gittergenerierung die in Abb. 12 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.

Für die Testrechnungen wurde eine Gaseinströmgeschwindigkeit von $u_F = 10 \text{ m/s}$ gewählt. Die Gasphase (Luft unter Normalbedingungen) transportiert eine disperse Phase, die aus Feststoffpartikeln mit einer Dichte von $\rho_P = 2500 \text{ kg/m}^3$ besteht und für die eine Eintrittsgeschwindigkeit von $u_P = 9 \dots 11 \text{ m/s}$ angenommen wurde. Die Partikelkonzentration ist im Eintrittsquerschnitt gleichverteilt. Für die Partikeleigenschaften (z.B. Partikel-Wandstoß-Verhalten) wurden Stoffparameter angenommen, wie sie für Siliziumpartikel typisch sind.

Für die Testrechnungen wurde jeweils ein Iterationszyklus bestehend aus der Berechnung des Fluidströmungsfeldes und der Berechnung der Bewegung der dispersen Phase auf dem so ermittelten Strömungsfeld ausgeführt. Für den Navier-Stokes-Löser wurde die Anzahl der äußeren

Iterationen auf 3000 limitiert. Die Anzahl der vom Lagrange-Löser berechneten Partikeltrajektorien lag ursprünglich bei 20000. Auf Grund von CPU-Zeit-Beschränkungen auf der Cray T3D am EPCC in Edinburgh (siehe 8) 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.

5.2.2 Für den Testfall verwendete Hard- und Software

a) Die Parallelisierungsverfahren

Für den Testfall wurden zum Vergleich zwei der unter 4.2 beschriebenen Parallelisierungsalgorithmen ausgewählt. Das erste der beiden Verfahren ist das unter 4.2.2 beschriebene DD-Verfahren. Das zweite Verfahren ist das unter 4.2.3 erläuterte DSM-Verfahren. Dieses tritt im Test in zwei verschiedenen Modifikationen auf, da die Bereitstellung der Fluidströmungsdaten durch die Memory-Manager-Knoten auf verschiedene Art und Weise möglich ist :

1. Wenn bei der Berechnung einer Partikeltrajektorie ein bestimmtes CV durchquert wird, so werden zur Fortsetzung der Berechnung die Strömungsdaten des betreffenden und der angrenzenden CV's benötigt (Strömungsdaten von 5 CV's im 2-d Fall und von 7 CV's im 3-d Fall). Nun können die Strömungsdaten für diese Anzahl von CV's immer dann angeliefert werden, wenn eine Partikeltrajektorie das betreffende CV durchquert (und die Daten noch nicht im lokalen Zwischenspeicher vorhanden sind). Diese Art der Datenbereitstellung wird im folgenden mit DSM-Point-Methode bezeichnet.

Obwohl bei dieser Methode relativ häufig mit geringem Datenvolumen kommuniziert wird, kann diese Art der Datenbereitstellung trotzdem von Vorteil sein in Abhängigkeit von der Kommunikationshardware des Parallelrechnersystems und von der verwendeten Gitterblockgröße. Dies resultiert daher, daß bei dieser Art der Datenbereitstellung nur eine 1-dimensionale Untermenge der gesamten Strömungsdaten auf dem numerischen Gitternetz zu dem rechnenden Knoten transferiert wird. Insbesondere bei 3-dimensionalen Berechnungen kann dies vorteilhaft sein.

2. Andererseits ist es im DSM-Verfahren möglich, die gesamten mit dem aktuell bearbeiteten Gitterblock in Verbindung stehenden Strömungsdaten vom Memory-Manager-Knoten zu dem rechnenden Knoten zu transferieren, 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 kann auf Parallelrechnern mit schnellen Kommunikationsnetzwerken und hoher Kommunikationsbandbreite die effizientere Methode sein.

b) Die MIMD-Parallelrechnersysteme

Als Testplattformen kamen zwei MIMD-Parallelrechnersysteme zum Einsatz. Das erste System ist wiederum der unter 5.1 bereits beschriebene Parsytec GC-128 der TU Chemnitz-Zwickau. Bei dem anderen System handelt es sich um eine Cray T3D am Edinburgh Parallel Computing Centre (EPCC) der Edinburgh University. Diese besteht aus 512 mit 150 MHz getakteten Alpha-Prozessoren 21064, der jeder mit 64 Megabyte lokalem Hauptspeicher ausgerüstet ist. Die Prozessoren sind mit einem sehr leistungsfähigen Kommunikationsnetzwerk in 3-dimensionaler, bi-direktionaler Torus-Topologie miteinander verbunden. Jeder der 6 Links eines Prozessors unterstützt eine Bandbreite von bis zu 300 Mb/s auf Hardware-Ebene.

Auf beiden Parallelrechnersystemen kamen zu PVM 3.2 und MPI 1.1 kompatible Softwareumgebungen zum Einsatz, die nur sehr geringe systemspezifische Modifikationen aufwiesen. Beide Softwareumgebungen unterstützen derzeit keine parallele Verarbeitung auf heterogenen Parallelrechner-Plattformen. Diese Einschränkung gilt jedoch nicht für das hier behandelte Euler/Lagrange-Lösungsverfahren, das auch auf heterogenen Parallelrechner-Umgebungen lauffähig ist.

5.2.3 Ergebnisse der Testrechnungen auf Parsytec GC-128 und Cray T3D

Insgesamt wurden 12 verschiedene Testserien durchgeführt, indem die Testrechnungen für die Strömungsgeometrie aus Abb. 12 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. 13 – 16 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.

Abb. 13 – 16 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-Implementation auf dem Parsytec GC-128 erzielten Meßwerte (Abb. 13) lassen im Vergleich mit den auf der Cray T3D erzielten Ergebnissen (Abb. 15) auf Probleme der PVM-Implementation auf dem Parsytec GC-128 schließen. Generell 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 bereits in den ersten Testrechnungen (siehe 5.1.4) festgestellten und auch von anderen Forschergruppen an der TU Chemnitz-Zwickau belegten Kommunikationsengpaß zwischen den 32-Knoten-Partitionen des Parsytec GC-128 zurückgeführt werden kann. Für die Testrechnungen auf 4 bis 32 Gitterblöcken zeigt Abb. 14 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. 15 und 16) 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

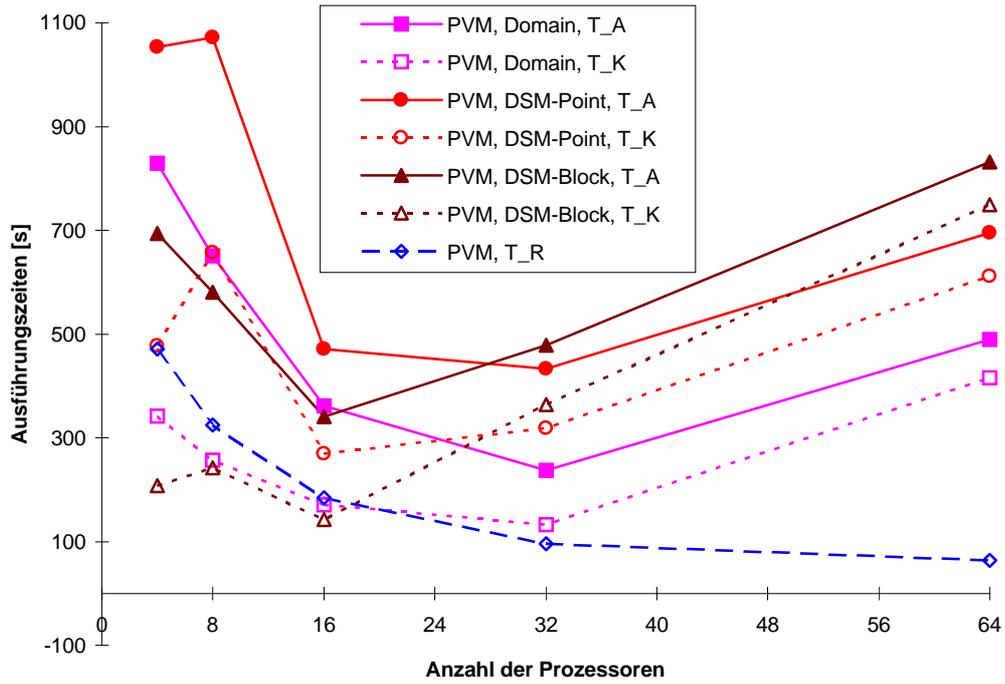


Abbildung 13: Ausführungszeiten der PVM-Implementationen auf dem Parsytec GC-128.

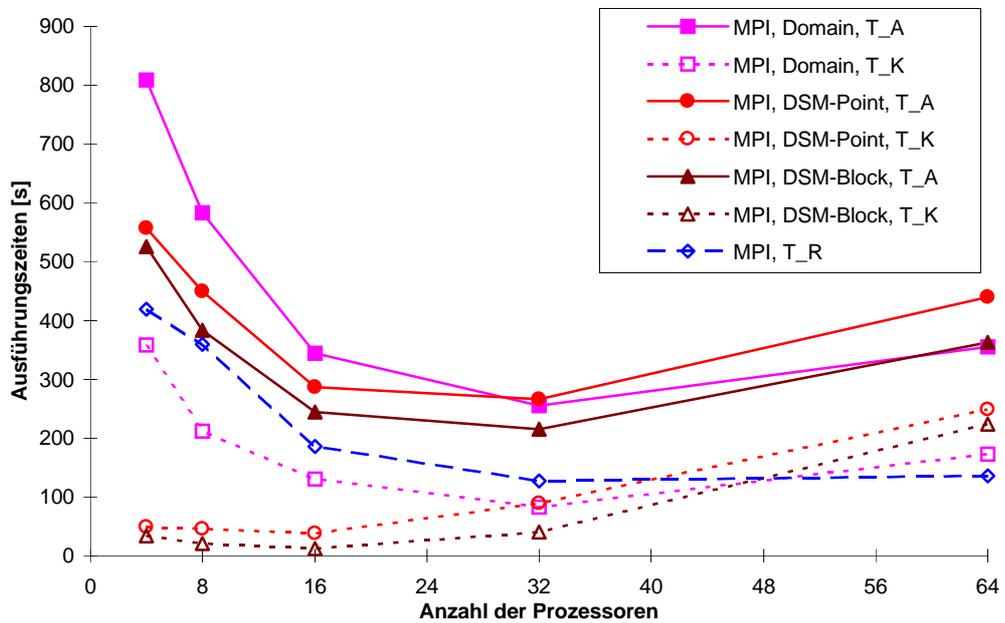


Abbildung 14: Ausführungszeiten der MPI-Implementationen auf dem Parsytec GC-128.

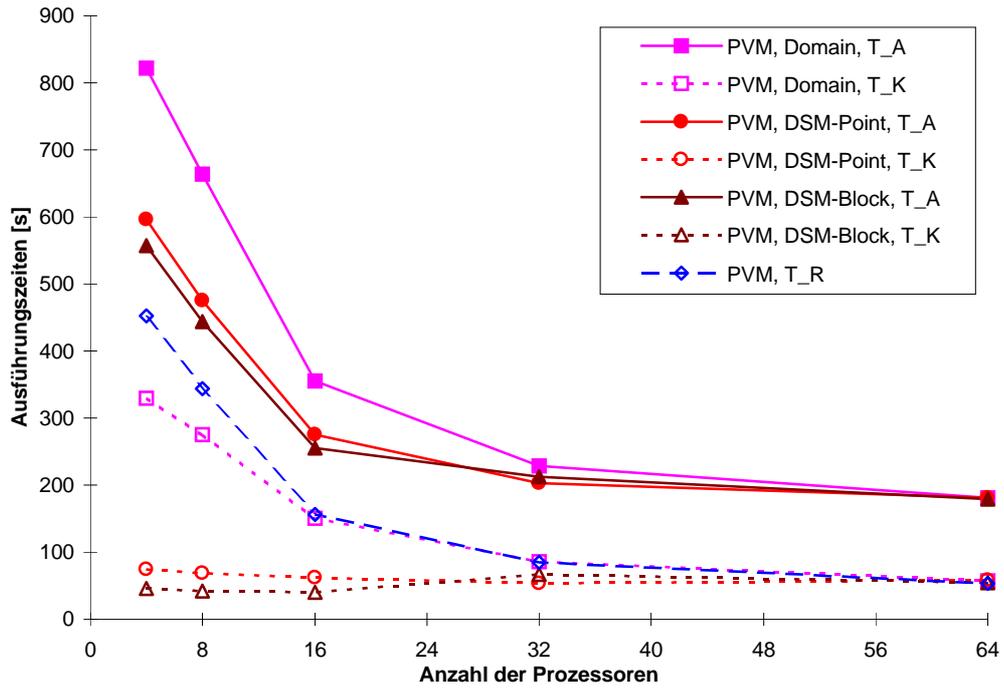


Abbildung 15: Ausführungszeiten der PVM-Implementationen auf der Cray T3D.

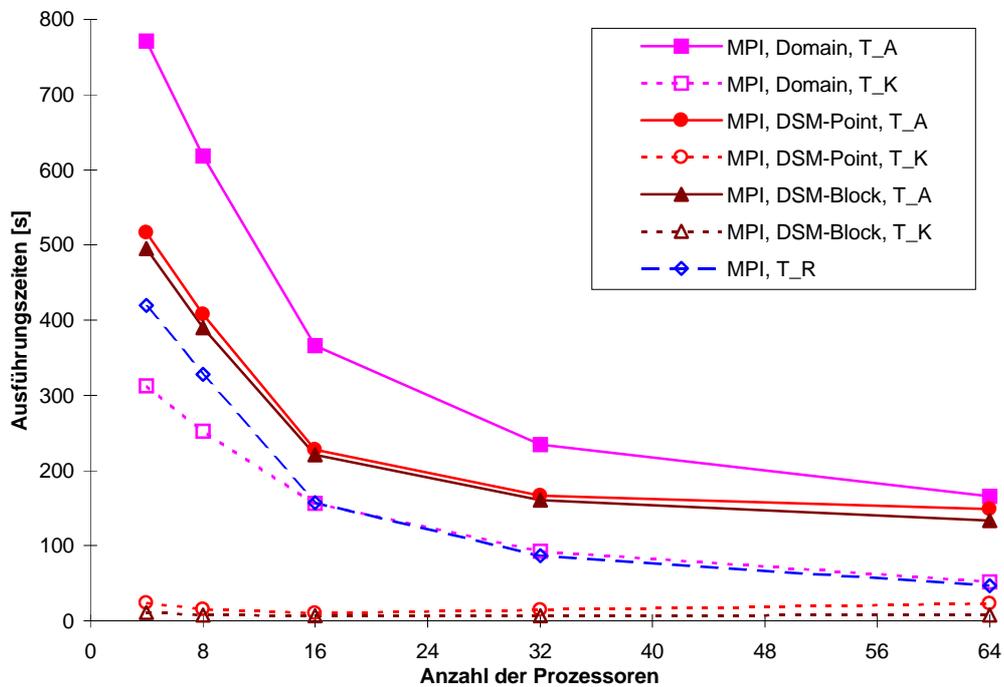


Abbildung 16: Ausführungszeiten der MPI-Implementationen auf der Cray T3D.

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 konnte. 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.

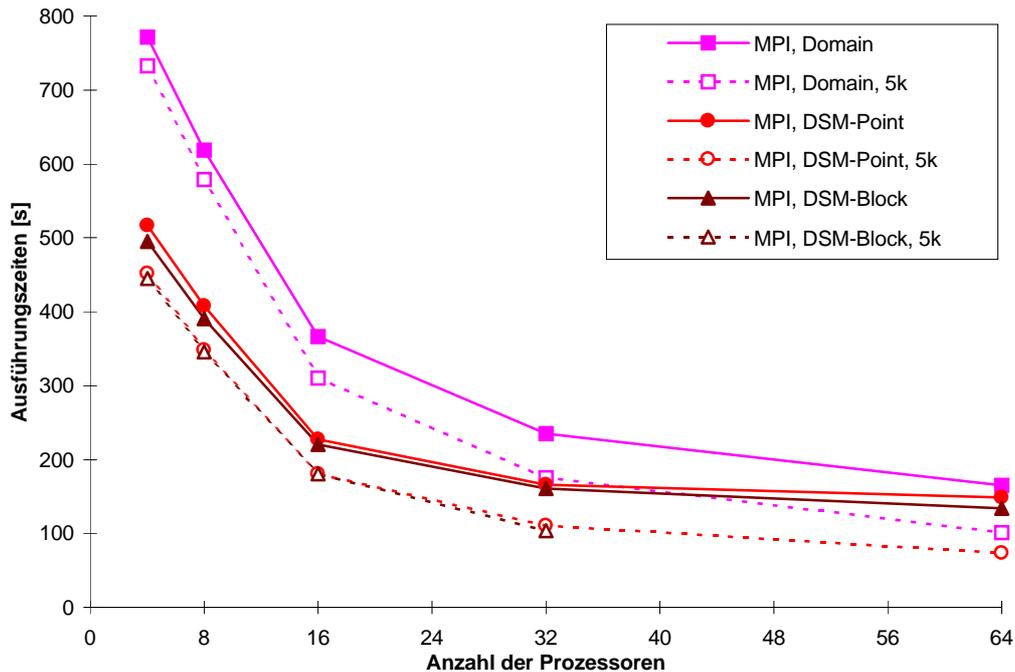


Abbildung 17: Abhängigkeit der Testergebnisse vom Berechnungsumfang (Cray T3D, MPI-Implementationen).

Um den Einfluß dieser Testfallbeschränkung näher quantifizieren zu können, wurden auf der Cray T3D für alle drei Parallelisierungsverfahren in der MPI-Implementierung drei weitere Testserien mit jeweils 5000 Partikeltrajektorien (also dem 5-fachen Berechnungsvolumen) durchgeführt. Abb. 17 zeigt die auf 1000 Partikeltrajektorien skalierten Testergebnisse aus diesen Vergleichsrechnungen. Die so ermittelten Programmausführungszeiten zeigen insbesondere für die hohen Prozessorzahlen ein deutlich besseres Skalierungsverhalten und eine deutlichere Reduktion der Programmausführungszeiten bei fortschreitender Verdopplung der Anzahl der Prozessoren. Dies läßt für sehr große Strömungsberechnungen (große Gitterblöcke mit einer großen Anzahl von CV's und/oder eine hohe Anzahl von zu berechnenden Partikeltrajektorien) ein noch besseres Skalierungsverhalten der entwickelten Parallelisierungsverfahren erwarten.

Abschließend zeigen die Abb. 18 – 20 die Gegenüberstellung der PVM- und MPI-Implementierungen auf dem Parsytec GC-128 und der Cray T3D jeweils für eines der eingesetzten Parallelisierungsverfahren. Abb. 19 und 20 zeigen dabei die zu erwartende bessere Rechenleistung der Cray T3D gegenüber dem Parsytec GC-128. Dieser Unterschied in der Rechenleistung der beiden MIMD-Architekturen wäre jedoch bei gleicher Testfallgröße noch deutlicher ausgefallen (wie schon erwähnt, wurde der Testfall auf der Cray T3D auf 1000 Partikeltrajektorien redu-

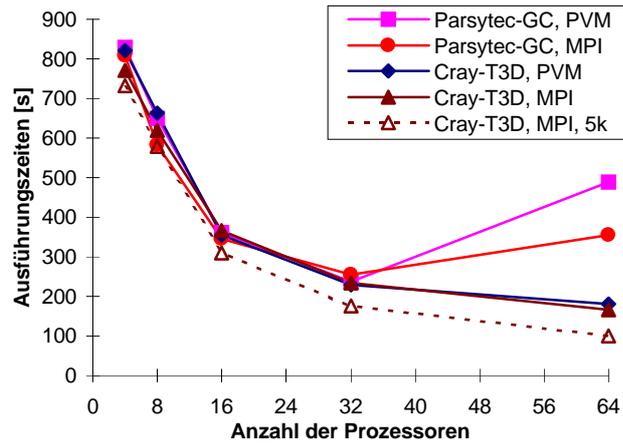


Abbildung 18: Vergleich der Implementierungen des DD-Verfahrens auf Parsytec-GC und Cray T3D.

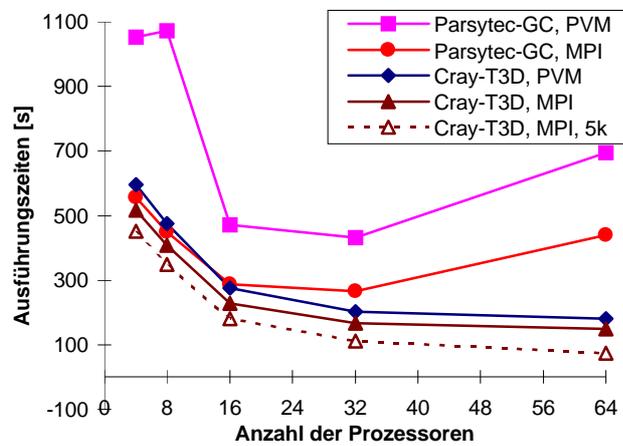


Abbildung 19: Vergleich der Implementierungen des DSM-Point-Verfahrens auf Parsytec-GC und Cray T3D.

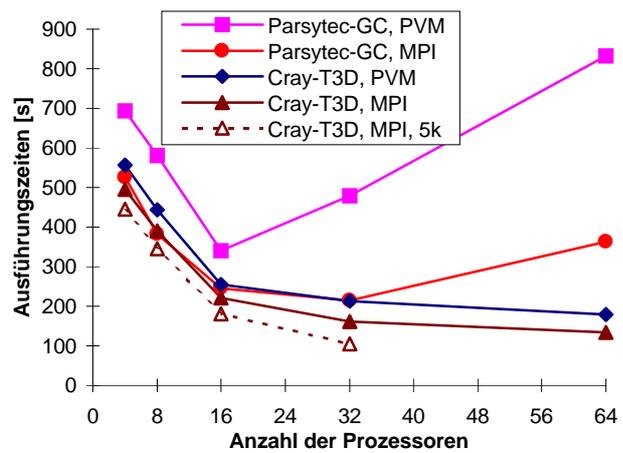


Abbildung 20: Vergleich der Implementierungen des DSM-Block-Verfahrens auf Parsytec-GC und Cray T3D.

ziert). Deutlich zu erkennen ist ebenfalls die etwas bessere Leistung der MPI-Implementierungen der DSM-Verfahren, die auf die bessere Kommunikationsleistung der zur Verfügung stehenden MPI-Standards auf den beiden untersuchten MIMD-Plattformen zurückzuführen ist.

Überraschend sind zunächst die in Abb. 18 dargestellten Ergebnisse für das Domain Decomposition Verfahren. Für die Testrechnungen auf den Gitternetzen mit 4 bis 32 Gitterblöcken läßt sich kaum ein Unterschied zwischen den Ergebnissen für den Parsytec GC-128 und die Cray T3D ausmachen. Eine Erklärung für dieses Verhalten läßt sich in den Abb. 21 und 22 finden. Zunächst läßt sich aus diesen beiden Darstellungen entnehmen, daß :

1. die Differenzzeit T_D im Wesentlichen aus Wartezeiten besteht, die auf den einzelnen Prozessoren anfallen, während Ein-/Ausgabevorgänge zwischen den einzelnen Prozessoren und dem Host-Prozessor ablaufen, und
2. die Ein-/Ausgabe-Zeit $T_{I/O}$ für 8 Prozessoren ca. 9–14% der absoluten Programmausführungszeit T_A ausmacht. Da dieser Teil der Programmausführungszeit unabhängig von der Prozessoranzahl annähernd konstant ist, nimmt sein relativer Anteil an der Gesamtabarbeitungszeit für größere Prozessoranzahlen zu.

In der Hauptsache wird jedoch durch die beiden Abb. 21 und 22 veranschaulicht, wie unterschiedlich die Lastverteilung zwischen den rechnenden Prozessoren im DD- und DSM-Verfahren ist. So zeigt Abb. 21, daß die Programmausführungszeit T_A für das DD-Verfahren hauptsächlich durch die Rechenzeit T_R bestimmt ist, die von den Prozessoren benötigt wird, denen diejenigen Gitterblöcke zugeordnet sind, die die Rohrreihen umschließen. Trotz annähernd gleicher CV-Anzahl kommt es auf diesen sehr krummlinig berandeten Teilblöcken der Strömungsgeometrie durch multiple Partikel-Wand-Stöße und stärkere Fluidgeschwindigkeitsgradienten (Reduktion des Integrationszeitschrittes) zu einem erhöhten Rechenzeitbedarf. Durch die starre Zuordnung der Gitterblöcke und der damit verbundenen Partikelanfangsbedingungen zu den Prozessoren kommt es im DD-Verfahren zu einer sehr ungleichmäßigen Lastverteilung (zur Lastverteilungsproblematik siehe auch 4.2.2).

Im Gegensatz dazu zeigt die Abb. 22 eine homogene Lastverteilung für die rechnenden Prozessoren des DSM-Block-Verfahrens. Dies wird durch die dynamische Zuordnung der Partikelanfangsbedingungen zu den rechnenden Prozessoren erreicht und ermöglicht eine bessere Ausnutzung der Rechenleistung der an der Berechnung beteiligten Prozessoren. Obwohl der durch das DSM-Verfahren verursachte Kommunikationsaufwand deutlich höher ist als für das DD-Verfahren (bei homogener Lastverteilung), können mit dem DSM-Verfahren auf geeigneter MIMD-Hardware mit hoher Kommunikationsleistung die besseren Gesamtergebnisse erzielt werden. Der Hauptvorteil des DSM-Verfahrens besteht jedoch in seiner universellen Einsetzbarkeit unabhängig von Besonderheiten der Mehrphasenströmung oder der Strömungsgeometrie auf Grund der durch das Verfahren gewährleisteten dynamischen Lastverteilung.

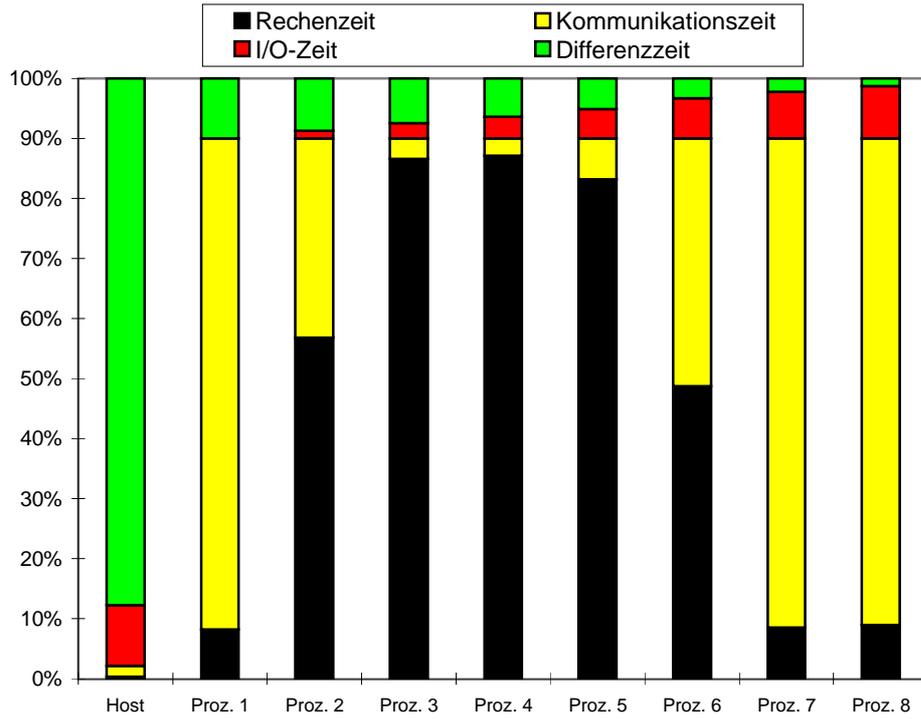


Abbildung 21: Lastverteilung für das DD-Verfahren auf 8 Gitterblöcken.

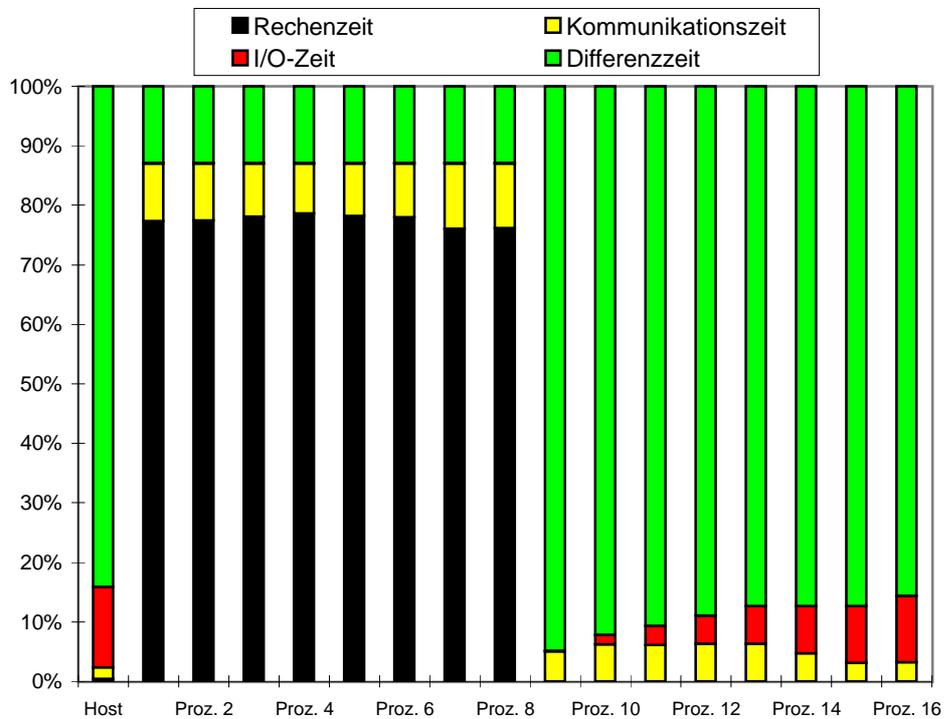


Abbildung 22: Lastverteilung für das DSM-Block-Verfahren auf 8 Gitterblöcken.

6 Arbeiten zur Berechnung 3–dimensionaler, disperser Mehrphasenströmungen

6.1 Grundlagen des 3–dimensionalen Euler–/Lagrange–Verfahrens

Unter 3.1 und 3.2 wurden bereits die Bewegungsgleichungen für die kontinuierliche und die disperse Phase in ihrer Formulierung für den 3–dimensionalen Fall dargestellt. Dabei und auch im Folgenden wird vorausgesetzt, daß es sich bei der dispersen Mehrphasenströmung um eine inkompressible, stationäre und isotherme Strömung handelt.

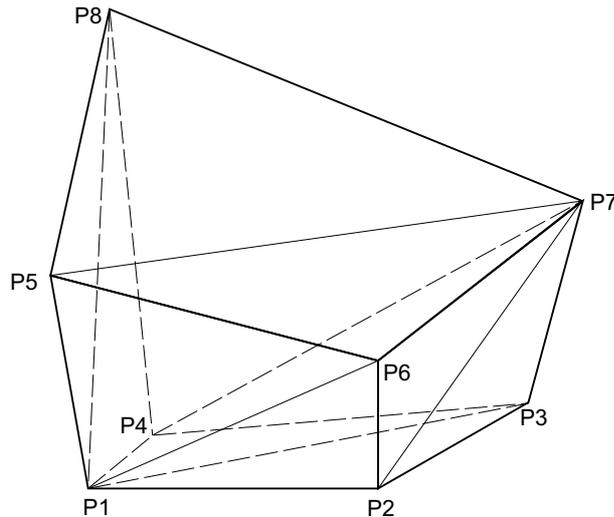


Abbildung 23: Form eines allgemeinen Hexaeder-Kontrollvolumens des 3–dimensionalen, numerischen Gitternetzes.

Ausgangspunkt für die Entwicklung des numerischen Lösungsverfahrens war die Bereitstellung einer Version des 3–dimensionalen Navier–Stokes–Lösers FAN–3D durch Prof. M. Perić vom Institut für Schiffbau der Universität Hamburg im Juni 1996. Die grundlegenden Eigenschaften dieses von Perić und Lilek [13] aus dem 2–dimensionalen Verfahren weiterentwickelten Lösungsverfahrens sind :

- Verwendung allgemeiner nicht–orthogonaler, oberflächenangepaßter, numerischer Gitternetze bestehend aus beliebig geformten Hexaeder-Kontrollvolumina (CV's) (siehe Abb. 23),
- Verwendung lokal strukturierter, global unstrukturierter (blockstrukturierter) Gitternetze für eine vereinfachte Approximation komplexer, 3–dimensionaler Geometrien und für die Parallelisierung des Lösungserfahrens nach der Domain Decomposition Methode (siehe 4.1),
- Diskretisierung der Bewegungsgleichungen auf nichtversetzten Gitternetzen,
- Finite–Volumen–Lösungsalgorithmus, der auf den SIMPLE–Algorithmus von Patankar und Spalding [8] zurückgeht. Im Gegensatz zu anderen Verfahren mit Diskretisierung der Erhaltungsgleichungen auf nichtversetzten Gitternetzen verwendet das Verfahren von Perić und Lilek eine vollkonservative Diskretisierung [11], insbesondere für den Druckgradienten in den Impulserhaltungsgleichungen.

Leider war (und z.T. ist) der Funktionsumfang des bereitgestellten Lösungsverfahrens FAN-3D in einigen wesentlichen Punkten stark eingeschränkt.

- Im Umfang des bereitgestellten Lösungsverfahrens war kein Werkzeug zur Generierung 3-dimensionaler Gitternetze enthalten. Die von dem Programmsystem FAN-3D verwendete Eingabeschnittstelle für die numerischen Gitternetze ist proprietär, so daß zunächst eine Schnittstelle zu einem kommerziell verfügbaren Gittergenerator geschaffen werden mußte, um eigene 3-dimensionale Gitternetze für zu untersuchende, komplexere Strömungsgeometrien erstellen und verwenden zu können. Die Wahl fiel dabei auf das bereits an der Professur vorhandene CFD-Paket CFX 4.1C mit dem darin enthaltenen Gittergenerator MESHBUILD.
- Im Verfahren ist zunächst nur die Lösung des gekoppelten Systems aus Kontinuitäts- und Impulserhaltungsgleichungen implementiert. Ein Turbulenzmodell und eine allgemeine Skalartransportgleichung sind zunächst in FAN-3D nicht enthalten. Auch wenn die nachträgliche Implementierung eines Turbulenzmodells mit vertretbarem Aufwand möglich ist, so ist das Programmsystem zunächst nur auf laminare Strömungen anwendbar.
- Das Programmsystem FAN-3D war in der bereitgestellten Version noch nicht parallelisiert. Auf Grund der Verwendung blockstrukturierter Gitternetze im numerischen Lösungsverfahren konnte – aufbauend auf den Erfahrungen mit der Parallelisierung des 2-dimensionalen Lösungsverfahrens FAN-2D – eine Parallelisierung des Navier-Stokes-Lösers nach der Domain Decomposition Methode realisiert werden. An dieser Stelle wird jedoch nicht näher auf das parallelisierte Verfahren eingegangen, da sich die Implementation noch in der Testphase befindet und abschließende Ergebnisse noch nicht vorliegen.

Grundlage für das 3-dimensionale Lagrange-Verfahren zur Lösung der Bewegungsgleichungen der dispersen Phase aus 3.2 war zum einen das in diesem Bericht ausführlich dokumentierte 2-dimensionale, parallelisierte Lösungsverfahren PartFlow-2D. Weitere Vorarbeiten für ein 3-dimensionales Lagrange-Verfahren waren bereits vor einiger Zeit an einer auf dem Navier-Stokes-Löser FASTEST-3D basierenden Implementierung von PartFlow erfolgt. Da sich jedoch die interne Darstellung des numerischen Gitternetzes, der Randbedingungen und aller damit verbundenen Geometriegrößen recht stark von der in FAN-3D gebräuchlichen Darstellung unterscheidet, mußte die Implementierung nochmals grundlegend überarbeitet werden.

Um eine erfolgreiche Anpassung des 3-dimensionalen Lagrange-Verfahrens an den Navier-Stokes-Löser FAN-3D und dessen Repräsentation der Strömungsgeometrie vornehmen zu können, mußten die folgenden grundlegenden Probleme gelöst werden :

1. Effiziente Lokalisierung der Partikelanfangsbedingungen auf dem numerischen Gitternetz.
2. Implementierung eines Partikelverfolgungsalgorithmus zur Berechnung von Partikeltrajektorien im gesamten Strömungsgebiet unter Berücksichtigung der Blockstrukturierung des numerischen Gitternetzes und einer zukünftigen, auf der Gebietszerlegung basierenden Parallelisierung des Lösungsverfahrens.
3. Effiziente Interpolationsalgorithmen zur Berechnung der Strömungsparameter an der gegenwärtigen Partikelposition.
4. Berechnung der Phasenwechselwirkungs-Quellterme und der mittleren Strömungsparameter der dispersen Phase.
5. Behandlung des Partikel-Wand-Stoßvorganges mit beliebig im Raum angeordneten Strömungsberandungen.

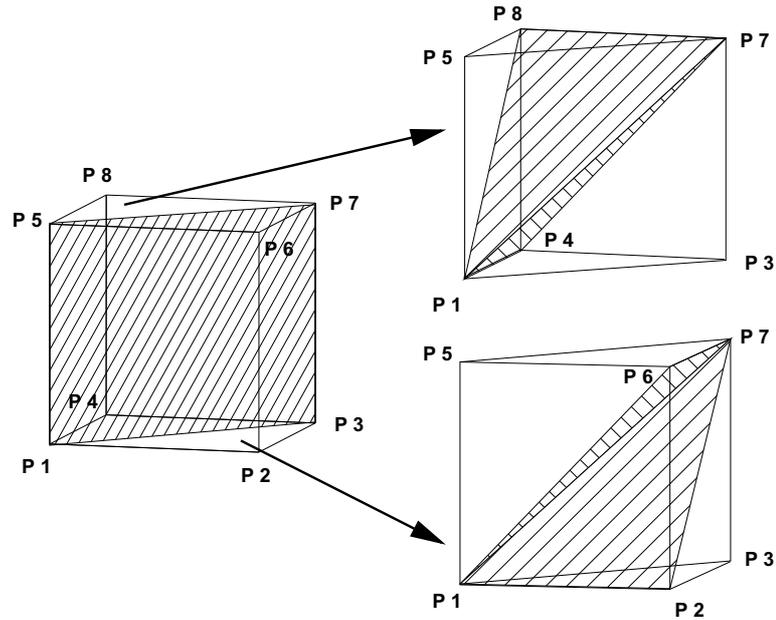


Abbildung 24: Zerlegung eines dodecaederförmigen Kontrollvolumens in sechs tetraederförmige Teilkontrollvolumina.

6.2 Geometrische Grundalgorithmen für das Lagrange-Lösungsverfahren

Wie bereits erwähnt basiert das in FAN-3D implementierte, numerische Lösungsverfahren auf einer Finite-Volumen-Diskretisierung der Kontinuums- und Impulserhaltungsgleichungen auf block-strukturierten Gitternetzen. Die Kontrollvolumen der einzelnen Blöcke sind beliebig geformte Hexaeder, die als zu Würfeln topologisch äquivalent betrachtet werden können (siehe Abb. 23). Die einzelnen Gitterblöcke des numerischen Gitters können beliebig miteinander verbunden werden mit der Restriktion, daß einer Kontrollvolumen-Seitenfläche auf der einen Seite einer solchen Gitterblock-Grenzfläche eine entsprechende Kontrollvolumen-Seitenfläche auf der benachbarten Blockgrenzfläche gegenüberliegt (Paßfähigkeit der Gitterblöcke). Im allgemeinen können die viereckigen Seitenflächen der Kontrollvolumen nicht als eben betrachtet werden. Die einzige Information, die im Rahmen der numerischen Gitterrepräsentation über die Form dieser Seitenflächen zur Verfügung steht, sind die Koordinaten aller acht Eckpunkte eines solchen Kontrollvolumens.

Für das Langrange-Verfahren wurde diese Definition eines Kontrollvolumens des numerischen Gitternetzes ergänzt durch das Hinzufügen der in Abb. 23 eingezeichneten sechs Oberflächendiagonalen, die jedes Hexaeder-Kontrollvolumen in ein Kontrollvolumen in Form eines Dodekaeders umwandeln, das von ebenen dreieckigen Seitenflächen begrenzt wird. Diese Form der geometrischen Diskretisierung des Strömungsgebiets führt zu einer Raumaufteilung, die sich gegenseitig überlappende Kontrollvolumen und nicht von Kontrollvolumen erfaßten Raum ausschließt. Dies ist eine der Grundvoraussetzungen für einen im 3-dimensionalen Raum fehlerfrei arbeitenden Partikelverfolgungsalgorithmus, da durch eine solche geometrische Diskretisierung eine eindeutige Zuordnung zwischen einer beliebigen Partikelposition und dem entsprechenden Kontrollvolumen des numerischen Gitters gegeben ist. Weiterhin erlaubt die Umwandlung der Kontrollvolumen in von ebenen Dreiecksflächen begrenzte Dodecaeder deren Zerlegung in sechs tetraederförmige Teilkontrollvolumen (Abb. 24). Eine solche Zerlegung kann bei der Suche nach dem eine bestimmte Partikelposition einschließenden Kontrollvolumen effizient eingesetzt

werden.

6.2.1 Lokalisierung der Partikelanfangsbedingungen auf dem numerischen Gitternetz

Für das Lagrange-Verfahren ist es zunächst notwendig, die Anfangsbedingungen für die Partikeltrajektorienberechnung auf dem numerischen Gitternetz zu lokalisieren. Hierzu muß für eine vorgegebene Partikelposition $P_I = (x_{PI}, y_{PI}, z_{PI})$ mit den dazugehörigen Partikelgrößen $(\vec{v}_{PI}, \vec{\omega}_{PI}, d_{PI}, \dot{N}_{PI}, T_{PI}, \dots)$ das Kontrollvolumen des numerischen Gitternetzes gefunden werden, daß diese Partikelposition einschließt. Für karthesische Gitternetze mit achsenparallelen CV-Seitenflächen kann diese Aufgabe durch einfache Vergleichsoperationen der Form :

$$x_{i-1} \leq x_{PI} \leq x_i, \quad y_{j-1} \leq y_{PI} \leq y_j, \quad z_{k-1} \leq z_{PI} \leq z_k$$

gelöst werden, indem die Koordinaten der Gitterlinien eines bestimmten Gitterblockes mit den Koordinaten der Anfangsbedingung P_I verglichen werden. Für komplexe 3-dimensionale Gitternetze der hier betrachteten Art ist es bedeutend komplizierter, diese Aufgabe zu lösen. Es existieren zwei grundlegende Methoden zur Lokalisierung einer Partikelposition auf dem numerischen Gitternetz.

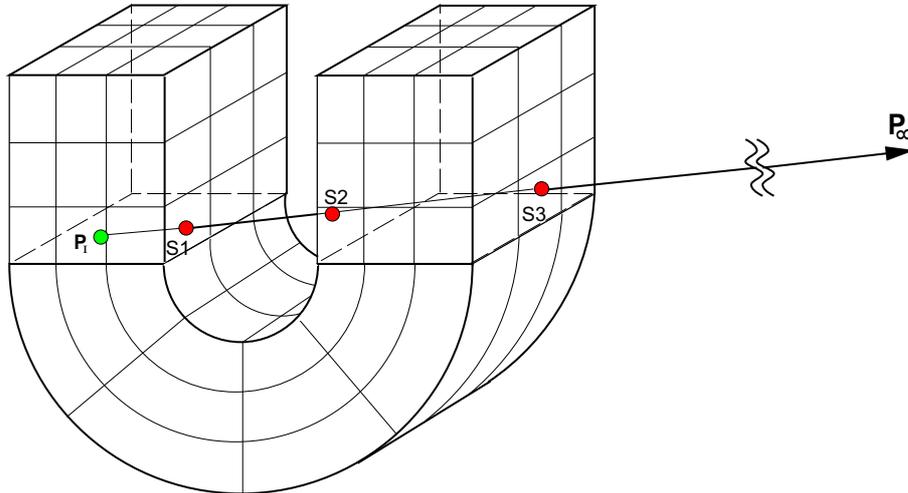


Abbildung 25: Lokalisierung einer Partikelposition auf dem numerischen Gitter mit dem 'Single-shot' oder Strahlverfolgungs-Verfahren.

Methode a) :

Die erste Methode wird von Preparata et al. in [14] vorgeschlagen und wird als 'Single-shot' oder auch Strahlverfolgungs-Verfahren bezeichnet (Abb. 25). Als Aufgabe sei angenommen, daß die Lage der Partikelposition P_I bezüglich eines beliebigen Teils des numerischen Gitternetzes bestimmt werden soll. Nun wird ausgehend von der Position P_I ein Strahl zu einem Punkt P_∞ untersucht, wobei angenommen wird, daß der Punkt P_∞ mit Sicherheit außerhalb der Gesamtgeometrie liegt. In Folge werden alle Schnittpunkte S_i des Strahls $\overrightarrow{P_I P_\infty}$ mit der Oberfläche der Gitternetz-Teilstruktur ermittelt. Wie bereits erläutert, besteht diese Oberfläche aus den dreieckigen Oberflächenelementen der dodecaeder-förmigen Kontrollvolumen der Gitternetz-Teilstruktur. Es kann gezeigt werden, daß der Punkt P_I innerhalb der untersuchten Gitternetz-Teilstruktur liegt, wenn die Gesamtanzahl der ermittelten Schnittpunkte S_i ungerade ist. Beginnend mit den einzelnen Blöcken des numerischen Gitters führt eine fortschreitende Bisektion zu dem CV, das die Partikelposition P_I einschließt (Abb. 26).

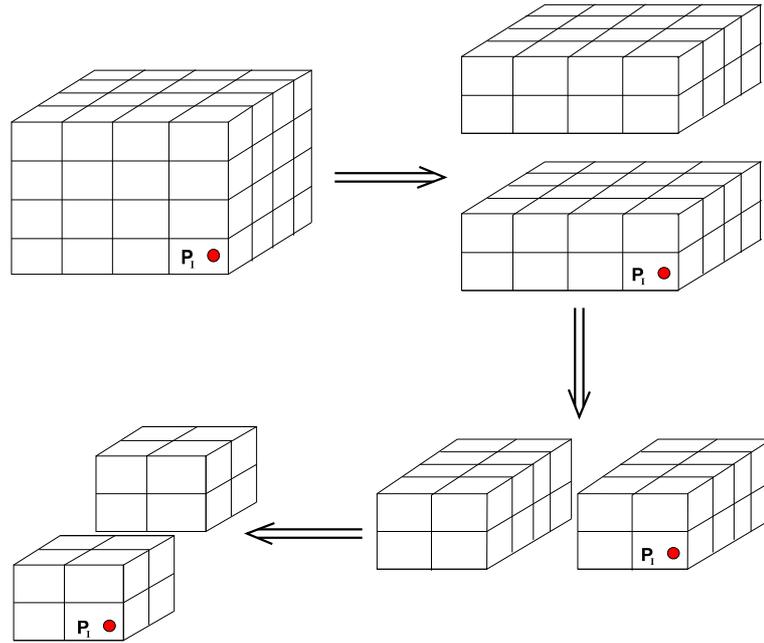


Abbildung 26: Lokalisierung einer Partikelposition P_I auf dem numerischen Gitter durch fortschreitende Bisektion der Gitterteilstruktur.

Wenn N^3 die Gesamtanzahl der Zellen in einem Gitterblock ist, so ist leicht ersichtlich, daß der numerische Berechnungsaufwand $E_N \sim O(N^2)$ ist, wenn die Dimension des Gitterblockes in jeder Koordinatenrichtung ungefähr gleich ist (kubische CV-Anordnung). E_N kann im ungünstigsten Fall auf $O(N^3)$ ansteigen, wenn die CV's eines Gitterblockes linear in einer Koordinatenrichtung angeordnet sind.

Des weiteren muß erwähnt werden, daß die Methode nicht in jedem Fall zu einem Ergebnis führt. Wenn der Strahl $\overrightarrow{P_I P_\infty}$ durch die Kante oder Ecke eines Rand-CV's der zu untersuchenden Gitterteilstruktur führt, oder der Strahl komplanar mit einer CV-Seitenfläche ist, so läßt sich keine Aussage über die Lage der Partikelposition P_I gewinnen. Dieses Problem kann nur über die Variation der Lage des Punktes P_∞ und eine Neuberechnung gelöst werden.

Methode b) :

Dieses Verfahren basiert auf der Tatsache, daß die Lage eines Punktes bzgl. einer Ebene einfach aus deren Normalengleichung abgeleitet werden kann. Wendet man dieses Verfahren auf die vier Seitenflächen eines Tetraeders und in der Folge auf die sechs Teiltetraeder eines dodecaeder-förmigen CV's an (Abb. 24), so kann auf einfache Art und Weise entschieden werden, ob eine zu untersuchende Partikelposition P_I innerhalb eines bestimmten CV's liegt oder nicht. Dieses Verfahren ist im Vergleich zu Methode a) 'robust', d.h. es liefert für alle denkbaren Kombinationen aus Partikelpositionen P_I und beliebig geformten CV's ein Ergebnis. Leider liegt der numerische Berechnungsaufwand jedoch stets bei $E_N \sim O(N^3)$.

Ein Vergleich beider Methoden zeigt, daß Methode a) effektiv auf größere Gitternetz-Teilstrukturen mit einer Gesamtanzahl der CV's größer als 10^2 angewendet werden kann. Sobald der fortschreitende Teilungsprozess aus Methode a) zu kleineren Gitterstrukturen führt, so sind diese mit Methode b) effektiver zu untersuchen.

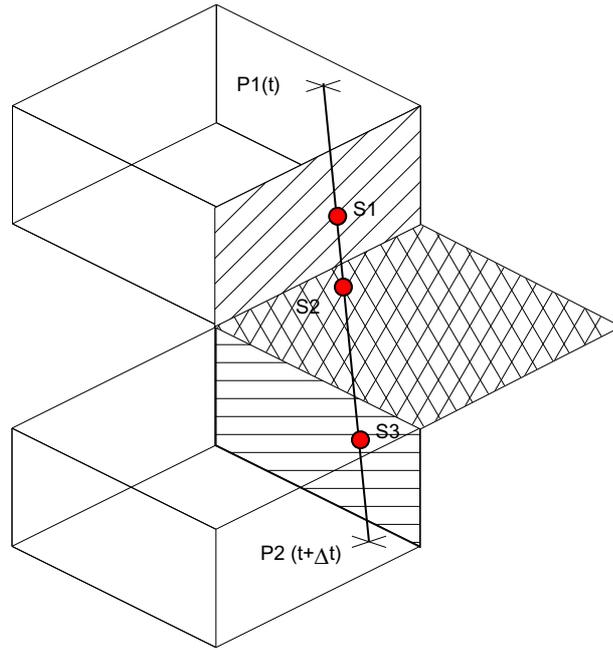


Abbildung 27: Der Partikel–Verfolgungsalgorithmus.

6.2.2 Der Partikel–Verfolgungsalgorithmus

Wenn die Partikelanfangsbedingungen auf dem numerischen Gitter lokalisiert worden sind, so können die Bewegungsgleichungen der dispersen Phase (siehe 3.2, Gl. 3) mit Hilfe eines Runge–Kutta–Lösungsverfahren 4. Ordnung mit automatischer Schrittweitensteuerung gelöst werden. Sobald für eine gegebene Partikelposition $P_1(t)$ die neue Partikelposition $P_2(t + \Delta t)$ berechnet wurde, ist diese auf dem numerischen Gitter erneut zu lokalisieren, d.h. das zu $P_2(t + \Delta t)$ korrespondierende CV ist zu ermitteln. Dies geschieht jedoch nicht durch die unter 6.2.1 beschriebenen Lokalisierungsmethoden, da diese bei bekannter Lage der Partikelposition $P_1(t)$ durch ein effektiveres Partikel–Verfolgungsverfahren ersetzt werden können.

Zunächst werden für das mit der Partikelposition $P_1(t)$ korrespondierende CV alle Schnittpunkte S_i der Strecke $\overrightarrow{P_1P_2}$ mit denjenigen CV–Grenzflächen ermittelt, für die das Skalarprodukt $(\vec{n}, \overrightarrow{P_1P_2}) \geq 0$ ist (Abb. 27). \vec{n} ist hierbei der äußere Normalenvektor auf die CV–Grenzfläche. Die Partikeltrajektorie verläßt das den Punkt $P_1(t)$ einschließende CV nun durch diejenige Seitenfläche, für die der Abstand $|\overrightarrow{P_1S_i}|$ minimal ist. Dies führt zu dem nächsten Nachbar-CV, das von der Partikeltrajektorie $\overrightarrow{P_1P_2}$ durchquert wird. Indem man diese Methode fortlaufend auf alle weiteren von der Trajektorie durchquerten CV's anwendet, gelangt man zur neu berechneten Partikelposition $P_2(t + \Delta t)$ und kann so auf Grund der Strukturiertheit des numerischen Gitters das umgebende Kontrollvolumen ermitteln.

6.2.3 Berechnung der Phasenwechselwirkungs-Quellterme und der mittleren Strömungsparameter der dispersen Phase

Das Partikel–Verfolgungsverfahren ist derart konstruiert, daß es zugleich zur Aufsummierung der Quelltermanteile für die Phasenwechselwirkungs–Quellterme aus 3.1, Gl. 1 entsprechend Gl. 9 verwendet werden kann, da bei der Lokalisierung der Partikelposition $P_2(t + \Delta t)$ alle Schnitt-

punkte des Trajektoriensegments $\overline{P_1P_2}$ mit zwischen den beiden Partikelzuständen liegenden CV–Grenzflächen bereits ermittelt werden. Die für die Berechnung der Phasenwechselwirkungs–Quelltermanteile und der Anteile zu den mittleren Strömungsparametern der dispersen Phase in den Schnittpunkten S_i benötigten Partikelzustandsgrößen werden entlang der Partikeltrajektorie $\overline{P_1P_2}$ linear interpoliert. Hierfür benötigte Werte einer Fluidströmungs–Variable Φ im Punkt $P = (x_P, y_P, z_P)$ werden aus den Werten der Variablen Φ im korrespondierenden CV und den benachbarten CV’s entsprechend Gl. (5) aus Kapitel 3.2.1 interpoliert.

6.2.4 Behandlung des Partikel–Wand–Stoßvorganges

Der Partikel–Wand–Stoß einer kugelförmigen Partikel mit einer ebenen Wand wird durch die Gl. 7–8 aus 3.2.2 beschrieben. Durch geeignete, in der Literatur beschriebene Verfahren [3, 6, 18] können auch Wandrauigkeiten oder Abweichungen in der Kugelform der Partikel im mathematischen Modell berücksichtigt werden.

Wurde nun bisher bei der im Lagrange–Verfahren angewandten Geometriediskretisierung von den durch Dreiecksflächen begrenzten, dodecaeder–förmigen Kontrollvolumen ausgegangen, so ist dies für die Behandlung des Partikel–Wand–Stoßvorgangs nicht ohne Einschränkungen möglich. Wie zu Beginn dieses Kapitels beschrieben, werden die hexaeder–förmigen CV’s des numerischen Gitters durch die mehr oder weniger willkürliche Auswahl je einer Oberflächendiagonale für jede der sechs CV–Grenzflächen in dodecaeder–förmige CV’s umgewandelt. Bei dieser Umwandlung fehlt jedoch der Bezug zur Krümmung der ursächlich zu diskretisierenden physikalischen Strömungsberandung, der nur während der Phase der Gittergenerierung vorhanden wäre. Somit kann der Fall eintreten, daß für eine stark gekrümmte feste Wand durch Wahl der “falschen“ Oberflächendiagonale der Grenzfläche eines Rand–CV’s die tatsächliche Wandkrümmung durch die entstehenden Dreiecksgrenzflächen verfälscht wiedergegeben wird. Um dem daraus resultierenden geometrischen Diskretisierungsfehler zu begegnen, erfolgt die Detektion eines Partikel–Wand–Stoßes zwar noch auf der Grundlage der dodecaeder–förmigen CV’s. Als Stoßebene für die Simulation des Partikel–Wand–Stoßes nach Gl. 7–8 wird jedoch eine mittlere Wandebene herangezogen, die aus den Koordinaten der vier Eckpunkte der ursprünglichen CV–Seitenfläche gebildet wird und somit die tatsächliche Krümmung der ursprünglich diskretisierten physikalischen Strömungsberandung bestmöglich widerspiegelt.

6.3 Zur Übertragbarkeit der Parallelisierungsverfahren

Die Untersuchungen zur Parallelisierung des 2–dimensionalen Lagrange–Verfahrens (siehe 5) haben gezeigt, daß Gebietszerlegungsverfahren auch für die numerische Simulation von dispersen Mehrphasenströmungen auf massiv parallelen MIMD–Rechnerarchitekturen effizient eingesetzt werden können. Dies gewinnt bei den für komplexe, 3–dimensionale Berechnungen benötigten hohen Rechenleistungen um so mehr an Bedeutung.

Es ist festzustellen, daß sowohl das Domain Decomposition Verfahren (4.2.2) als auch das Distributed Shared Memory Verfahren (4.2.3) auf das entwickelte 3–dimensionale Euler–/Lagrange–Berechnungsverfahren für phasengekoppelte, disperse Mehrphasenströmungen ohne Einschränkungen übertragbar sind. Konkrete Implementierungen beider Parallelisierungsverfahren und Untersuchungen zu deren paralleler und numerischer Effizienz für 3–dimensionale Mehrphasenströmungen sind Gegenstand eines gegenwärtig im Rahmen des DFG–Sonderforschungsbereichs 393 “Numerische Simulation auf massiv parallelen Rechnern” bearbeiteten Forschungsvorhabens.

7 Teilnahme an Fachtagungen und Konferenzen

Die erste im Rahmen dieses Forschungsvorhabens besuchte Fachtagung war die *3rd Int. Conference "Parallel CFD '96"* auf Capri, Italien vom 20.–23. May 1996. Dies ist eine europäisch dominierte Konferenz, die jedes Jahr in einem anderen Land stattfindet und die Fachleute auf dem Gebiet der parallelen Anwendungen in der numerischen Strömungsmechanik zu einem Erfahrungsaustausch zusammenbringt. Vom Antragsteller wurde der Beitrag *Th. Frank : "Comparison of three parallelization methods for calculation of disperse multiphase flows using the Lagrangian approach"* präsentiert. Die Konferenz gab eine Übersicht über die aktuellen Aktivitäten im Bereich der Parallelisierung von CFD-Algorithmien und war daher auch in Zusammenhang mit dem vorliegenden Projekt für den Antragsteller von großer Bedeutung, auch wenn ein großer Anteil der vorgestellten Beiträge sich mit Anwendungen aus der Luft- und Raumfahrt befaßte.

Die zweite besuchte Tagung war das *2. International Symposium on Numerical Methods for Multiphase Flows* das Bestandteil des *ASME Fluids Engineering Division Summer Meeting 1996* in San Diego, USA war und vom 7.–11. Juni 1996 stattfand. Diese Tagung der ASME findet jährlich in den USA statt und insbesondere das erwähnte Teilsymposium versammelt regelmäßig die führenden Fachleute auf dem Gebiet der numerischen Methoden für die Berechnung von Mehrphasenströmungen. Neben dem Beitrag des Antragstellers *Th. Frank, E. Wassen : "Parallel Solution Algorithms for Lagrangian Simulation of Disperse Multiphase Flows"* wurde erstmalig auch von einer anderen Wissenschaftlergruppe (Prof. Y. Tsuji, University of Tokio) über parallele Algorithmen zur Berechnung von dispersen Mehrphasenströmungen berichtet. Diese alljährlich stattfindende Konferenz auf dem Fachgebiet des Antragstellers bietet eine hervorragende Gelegenheit zum Erfahrungsaustausch und zum Knüpfen von Kontakten für weiterführende Zusammenarbeit sowie einen guten und umfassenden Überblick über aktuelle Entwicklungen an den anderen internationalen Forschungszentren.

Im Ergebnis des 2-monatigen Forschungsaufenthalts am Edinburgh Parallel Computing Centre (EPCC) an der Universität Edinburgh, GB im Rahmen des von der europäischen Union geförderten TRACS-Programms (Training and Research on Advanced Computing Systems) im Februar/März 1997 (siehe hierzu auch 8) sind 2 Beiträge für internationale Konferenzen entstanden. Der erste wurde auf der vom 26. April bis 1. Mai 1997 in Lochinver, Schottland stattfindenden *1. Euro-Conference on Parallel and Distributed Computing for Computational Mechanics* vorgestellt (*E. Wassen, Th. Frank : "A Comparison of Parallel Algorithms for the Numerical Simulation of Multiphase Flows"*). Diese von der europäischen Union bezuschulte Konferenz bietet vor allem jungen Nachwuchswissenschaftlern die Möglichkeit, ihre Arbeiten in einem internationalen Rahmen zu präsentieren und frühzeitig internationale Kontakte zu anderen Forschern auf ihrem Fachgebiet herzustellen. Der zweite Beitrag wird auf der vom 18.–19. Juli 1997 in Tokio, Japan stattfindenden Konferenz *ISAC '97 Conference on Advanced Computing on Multiphase Flow* präsentiert, die Teil des 10-jährigen Jubiläumskongresses der JSME ist. Der Beitrag trägt den Titel *Th. Frank, E. Wassen : "Parallel Efficiency of PVM- and MPI-Implementations of two Algorithms for the Lagrangian Prediction of Disperse Multiphase Flows"* und wird die in Kapitel 5.2 dargestellten Ergebnisse einem breiten, internationalen Fachpublikum vorstellen.

Durch die Teilnahme an der schon erwähnten ASME Jahrestagung, die in diesem Jahr vom 22.–26. Juni 1997 in Vancouver, Kanada, stattfindet, werden die Entwicklungen von parallelen Lösungsverfahren für komplexe, 3-dimensionale Mehrphasenströmungen publiziert. Titel des präsentierten Konferenzbeitrags ist *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"*.

8 Andere Aktivitäten im Rahmen des Forschungsaufenthaltes an der Edinburgh University im Rahmen des TRACS-Programms

Der Antragsteller hatte sich im Sommer 1996 im Rahmen des TMR-Programms (Training and Mobility of Researchers) der EU für einen 2-monatigen Forschungsaufenthalt am EPCC (Edinburgh Parallel Computing Centre) der Universität Edinburgh, Schottland beworben, der auch für Februar/März 1997 bewilligt wurde. Am EPCC stehen den Gastwissenschaftlern im Rahmen des EU-geförderten TRACS-Programms (Training and Research on Advanced Computing Systems) mehrere parallele Hochleistungsrechner zur Verfügung (Cray T3E, Cray T3D, Meiko CS-2, Cray YMP, Cray J90, SUN/SGI-Workstationcluster). Bei der Cray T3D mit 512 Alpha-Prozessoren und einer Gesamtrechenleistung von bis zu 76.8 GFlops handelt es sich um das derzeit leistungsfähigste Hochleistungsrechnersystem Großbritanniens.

Die Zeit am EPCC wurde vom Antragsteller hauptsächlich genutzt, um die Portabilität der entwickelten Algorithmen zu verbessern und umfangreiche Tests auf der Cray T3D vorzunehmen, um verlässliche Aussagen über die Effizienz der eingesetzten Parallelisierungsverfahren zu erlangen und um effizienzsteigernde Detailverbesserungen vorzunehmen. Unterstützt durch die vom EPCC durchgeführten Anwenderlehrgänge und die direkten Konsultationsmöglichkeiten mit den Mitarbeitern des EPCC, entstand während dieser Zeit auch eine MPI-Portierung der im Euler-/Lagrange-Lösungsverfahren zur Parallelisierung eingesetzten Kommunikationsschicht. Dadurch konnte die Portabilität des Verfahrens nochmals wesentlich verbessert werden, da mittlerweile von allen Herstellern paralleler MIMD-Rechnersysteme PVM- und/oder MPI-Bibliotheken angeboten werden.

Durch die fachliche Anbindung des Antragstellers am Department of Mechanical Engineering in der Forschungsgruppe für Strömungsmechanik von Dr. W.J. Easson ergab sich auch die Möglichkeit eines regen fachlichen Austausches, da diese Forschungsgruppe auch mit Entwicklungen auf dem Gebiet der Particle Image Velocimetry (PIV) und deren Anwendung auf Gas-Feststoff-Strömungen befaßt ist. Insbesondere die Untersuchungen zur Strahlenbildung beim pneumatischen Transport von Gas-Feststoff-Gemischen in Kanälen und Rohrleitungen und zur Untersuchung von Einbauten, die der Auflösung dieser Strahlen dienen sollen, sind von weiterführendem beiderseitigem Interesse. Gemeinsame vergleichende experimentell/numerische Untersuchungen dieser sogenannten 'rope splitter' und des Strahlverschleißes in Anlagen zum pneumatischen Transport wurden diskutiert. Dr. W.J. Easson zeigte starkes Interesse an dem im vorliegenden Projekt entwickelten Berechnungsverfahren für disperse Mehrphasenströmungen, an den Weiterentwicklungen im Hinblick auf 3-dimensionale Mehrphasenströmungen und an den Arbeiten zur Parallelisierung. Ein weiterer Forschungsaufenthalt im Rahmen des TRACS-Programms ist für einen Mitarbeiter am vorliegenden Projekt (E. Wassen) für das Frühjahr 1998 vorgesehen.

Die Ergebnisse der Arbeiten auf der Cray T3D am EPCC wurden am Department of Mechanical Engineering im Rahmen eines Forschungsseminars vorgestellt. Mit den Betreuern vom EPCC und den Mitarbeitern der Forschungsgruppe von Dr. W.J. Easson ergaben sich wertvolle Diskussionen über die entwickelten Parallelisierungsverfahren, deren Effizienz auf MIMD-Rechnerarchitekturen und die Möglichkeiten zu deren weiterer Verbesserung. Während des Forschungsaufenthalts an der Universität Edinburgh wurde des weiteren eine Veröffentlichung der Ergebnisse auf der vom 18.-19. Juli 1997 in Tokio, Japan stattfindenden Konferenz *ISAC '97 Conference on Advanced Computing on Multiphase Flow* (siehe 7) fertiggestellt. Die Ergebnisse flossen auch in den Beitrag zur *1. Euro-Conference on Parallel and Distributed Computing for Computational Mechanics* in Lochinver, Schottland im May 1997 ein. Ein weiterer Beitrag des Antragstellers erscheint in dem jährlichen Forschungsbericht des EPCC.

9 Abschließende Bemerkungen

Die im Rahmen dieser Arbeiten durchgeführten Untersuchungen haben gezeigt, daß Euler-/Lagrange-Verfahren zur Berechnung phasengekoppelter, disperser Mehrphasenströmungen mittels der entwickelten Parallelisierungsverfahren effektiv auf MIMD-Architekturen einzusetzen sind. Wie an anderer Stelle gezeigt [11], ist das dem Berechnungsverfahren für die Bewegung der kontinuierlichen Phase zu Grunde liegende Finite-Volumen-Verfahren sehr einfach und effizient durch Gebietszerlegung parallelisierbar. Die Datenabhängigkeiten bei der Kopplung der Bewegungsgleichungen für die kontinuierliche und die disperse Phase können durch die im vorliegenden Forschungsvorhaben entwickelten Parallelisierungsmethoden für das Lagrange-Verfahren effizient berücksichtigt werden. Das QS-Verfahren stellt dabei die einfachste Form einer Verfahrensparellisierung dar, indem das ursprüngliche serielle Lagrange-Verfahren auf jedem Prozessor des Parallelrechnersystems ausgeführt wird. Die hohen Anforderungen an die lokalen Speicherressourcen der Prozessorknoten stehen jedoch einer generellen Anwendung des QS-Verfahrens auf MIMD-Systemen entgegen. Das DD-Verfahren gestattet für Mehrphasenströmungen ohne Phasenseparation/Entmischung die sehr einfache Parallelisierung auf der Grundlage des gleichen Gebietszerlegungskonzepts, wie es für die Berechnung der Bewegung der kontinuierlichen Phase zur Anwendung kommt. Hauptnachteil des DD-Verfahrens ist jedoch seine Anfälligkeit gegenüber Effizienzeinbußen durch Lastverteilungsprobleme. Das DSM-Verfahren stellt dagegen eine unabhängig von der Art der dispersen Mehrphasenströmung anwendbares Parallelisierungsverfahren dar und kann auf MIMD-Rechnerarchitekturen mit hoher Kommunikationsleistung in einem weiten Bereich von verwendeten Prozessoren effizient eingesetzt werden. Dabei hat sich der DSM-Block-Modus als der effizientere Übertragungsmodus für die benötigten Strömungsdaten erwiesen.

Die Verwendung einer Kommunikationszwischenschicht zur Parallelisierung des numerischen Verfahrens hat eine hohe Portabilität der Implementierung zur Folge. Veränderungen an Schnittstellen der verwendeten Parallelisierungsstandards oder geringe Abweichungen vom definierten Standard (wie z.B. auf der Cray T3D) haben nur wenige Änderungen in der Kommunikationsschicht zur Folge. Neue Standards sind auf diese Weise einfach nachträglich zu implementieren. Durch die Implementierung der z.Zt. gebräuchlichsten Parallelisierungsstandards PVM 3.2 und MPI 1.1 ist eine außerordentlich breite Palette von MIMD-Rechnerarchitekturen abgedeckt, auf denen das numerische Verfahren ohne umfassende Änderungen eingesetzt werden kann.

Die Effekte der parallelen Effizienz sind sehr stark von Hardware und von der Qualität der Implementierung der Parallelisierungsstandards abhängig. Insbesondere Kommunikations-Coprozessoren auf den Knoten des Parallelrechners, eine gute Topologie des Kommunikationsnetzwerks (wie z.B. die Torustopologie der Cray T3D) und hohe Kommunikationsbandbreite mit sehr kurzen Initialisierungszeiten führen zu hoher paralleler Effizienz der entwickelten Parallelisierungsverfahren. Dies ist insbesondere für die Anwendungen auf dem Gebiet der parallelen CFD und für Gebietszerlegungsverfahren im Besonderen von entscheidender Bedeutung, da der Kommunikationsbedarf dieser Parallelisierungsmethoden nicht beliebig minimiert werden kann.

Die zukünftigen Entwicklungen werden in Richtung 3-dimensionaler und instationärer Strömungen verlaufen. Mehrphasenströmungen mit sehr starker Phasenkopplung (Masse-, Impuls- und Wärmeaustausch zwischen den Phasen) stellen für ihre erfolgreiche numerische Berechnung höchste Anforderungen an die Rechenleistung der verfügbaren Hardware und an die Effizienz der eingesetzten numerischen Berechnungsverfahren. Auch Verfahren zur direkten numerischen Simulation (DNS) werden in zunehmendem Maße auf disperse Mehrphasenströmungen angewandt. Die Entwicklung der Parallelrechner läßt erwarten, daß Berechnungen auf parallelen Plattformen (WS-Cluster, MIMD-Computer) in Kürze routinemäßig durchgeführt werden, wenn auch heute die Stabilität und Benutzerfreundlichkeit mancher Systeme noch sehr zu wünschen übrigläßt.

10 Eigene Veröffentlichungen im Zeitraum 1995–1997

1. Schade K.-P., Frank Th., Petrak D., Hädrich Th. : "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, Proceedings, pp.377–388, Nürnberg, Germany, March 21–23, 1995.
2. Frank Th. : "Disperse Multiphase Currents of Air Pollutants",
Final Report on ESPRIT Projekt No. 5404, GP MIMD, Extension of Workpackage 6, Activity 6.2.3, March 1995, TU Chemnitz–Zwickau, Germany.
3. Frank Th. : "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.
4. 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, Vol. 1 (FED–Vol. 236), pp. 11–20.
5. Wassen E., Frank Th., 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.
6. Frank Th., Wassen E., Q. Yu : "A 3-dimensional Lagrangian Solver for Disperse Multiphase Flows on Arbitrary, Geometricaly Complex Flow Domains using Block-structured Numerical Grids",
7th Int. Symposium on Gas–Particle Flows, ASME Fluids Engineering Division Summer Meeting, Vancouver, BC, Canada, July 22–26, 1997.
7. 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.

Literatur

- [1] Crowe C.T., Sharma M.P., Stock D.E., 1977, "The Particle-Source-In Cell (PSI-Cell) Model for Gas-Droplet Flows", *Trans. of ASME, J. Fluids Eng.*, Vol. 99, pp. 325-332.
- [2] Crowe C.T., 1982, "REVIEW — Numerical Models for dilute Gas-Particle Flows," *Trans. of ASME, J. Fluids Eng.*, Vol. 104, pp. 297-303.
- [3] Frank Th, 1992 "Numerische Simulation der feststoffbeladenen Gasströmung im horizontalen Kanal unter Berücksichtigung von Wandrauigkeiten", PhD Thesis, Techn. University Bergakademie Freiberg, Germany.
- [4] Frank Th., Schulze I., 1994, "Ein numerisches Verfahren zur Berechnung disperser Mehrphasenströmungen auf parallelen Hochleistungsrechnern", *Proc. Arbeitssitzung des GVC-Fachausschusses Mehrphasenströmungen*, February 17-18, 1994, Würzburg, Germany.
- [5] Frank Th., Schulze I., 1994, "Numerical simulation of gas-droplet flow around a nozzle in a cylindrical chamber using Lagrangian model based on a multigrid Navier-Stokes solver", *International Symposium on Numerical Methods for Multiphase Flows*, June 19-23, 1994, Lake Tahoe (NV), USA.
- [6] Matsumoto S., Saito S., Maeda S., 1976, "Simulation of gas-solid two-phase flow in horizontal pipe", *Journal of Chemical Engineering of Japan*, Vol. 9, No. 1, pp. 23-28.
- [7] Milojević D., 1990, "Lagrangian Stochastic-Deterministic (LSD) Predictions of Particle Dispersion in Turbulence," *Part. Part. Syst. Charact.*, Vol. 7, pp. 181-190.
- [8] Patankar S.V., Spalding D.B., 1972, "A Calculation Procedure for Heat, Mass and Momentum Transfer in 3-Dimensional, Parabolic Flows", *Int. J. Heat Mass Transfer*, Vol. 15, pp. 1787-1806.
- [9] Patankar S.V., 1980, "Numerical Heat Transfer and Fluid Flow", McGraw-Hill, New York.
- [10] Perić M., 1989, "A Finite Volume Multigrid Method for Calculating Turbulent Flows," *Proc. 7th Symposium on Turbulent Shear Flows*, Vol. 1, pp. 7.3.1.-7.3.6., Stanford University, USA.
- [11] Perić M., 1992, "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.
- [12] Schreck E., Perić M., 1992, "Parallelization of implicit solution methods", *ASME Fluids Engineering Conference*, June 22-23, 1992, Los Angeles (CA), USA.
- [13] Perić M., Lilek Ž., 1993, "Users Manual for the FAN-2D Software for the Calculation of Incompressible Flows", Institut für Schiffbau der Universität Hamburg, Germany.
- [14] Preparata F.P., Shamos M.I. : "Computational Geometry – An Introduction", Texts and Monographs in Computer Science, Springer Verlag, New York.
- [15] Schönung B., 1987, "Comparison of Different Dispersion Models for Particles in Lagrangian and Eulerian Prediction Codes," *In : Proceedings of the International Conference on Fluid Mechanics*, Peking, July 1.-4., 1987, Peking University Press, China.

- [16] Sommerfeld M., 1992, "Modelling of particle-wall collisions in confined gas-particle flows", *Int. Journal of Multiphase Flows*, Vol. 18, No. 6, pp. 905-926.
- [17] Stone, H.L., 1968, "Iterative solution of implicit approximations of multidimensional partial differential equations", *SIAM J. Numer. Anal.*, Vol. 5, pp.530-558
- [18] Tsuji Y., Oshima T., Morikawa Y., 1985, "Numerical simulation of pneumatic conveying in a horizontal pipe", *KONA - Powder Science and Technology in Japan*, No. 3, pp. 38-51.
- [19] Tsuji Y., Shen N.Y., Morikawa Y., 1991, "Lagrangian Simulation of Dilute Gas-Solid Flows in a Horizontal Pipe", *Advanced Powder Technology*, Vol. 2, No. 1, pp. 63-81.
- [20] Tsuji Y., 1996, "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, Vol. 1 (FED-Vol. 236), p. 3.
- [21] Tysinger T.L., Missaghi M., 1997, "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.