Proc. 3rd Int. Conf. "Parallel CFD '96",
Implementations and Results Using Parallel Computers,
May 20–23, 1996, Capri, Italy

Comparison of three parallelization methods for calculation of disperse multiphase flows using the Lagrangian approach

Th. Frank
Technical University of Chemnitz–Zwickau
Faculty of Mechanical Engineering and Process Technology
Research Group of Multiphase Flow
Chemnitz, Germany

Name of the author
Dr. Thomas Frank

Affiliation / Adress
English description of affiliation see header information.
adress : Technische Universität Chemnitz–Zwickau
Fakultät für Maschinenbau und Verfahrenstechnik
Forschungsgruppe Mehrphasenströmungen
Dr. Th. Frank
Reichenhainer Straße 70
09107 Chemnitz, Germany

Type of the problem
Novel parallel algorithms for parallel Lagrangian computation of disperse multiphase flows.

Abstract

In the last years Lagrangian simulation has become an efficient and widely used method for calculation of various kinds of 2– and 3–dimensional disperse multiphase flows (i.e. gas–particle and gas–droplet flows). On the other hand Lagrangian simulations of coupled multiphase flow systems with strong phase interaction are among the applications with the highest demands on computational effort and on system ressources in the field of computational fluid dynamics. Massively parallel computers (MIMD) provide new capabilities for efficient multiphase flow calculations. However, special parallel solution algorithms have to be developed in order to use the computational power of parallel MIMD computers.

The paper deals with the development and comparison of three different parallelization methods for Lagrangian multiphase flow calculations on parallel MIMD computer architectures. All parallelization methods are based on the serial Lagrangian approach developed by the author et al. [2, 3, 4]. For the flow calculation of the continuous phase a modified finite volume Navier–Stokes solver developed by M. Perić and Ž. Lilek [8] is used.

For the case of a steady, incompressible and isothermal two–phase flow the time–averaged form of the governing

fluid phase equations can be cast into the following form of the general transport equation :

$$\frac{\partial}{\partial x}(\rho_F\, u_F\, \Phi) + \frac{\partial}{\partial y}(\rho_F\, v_F \Phi) = \frac{\partial}{\partial x}\left(\Gamma\, \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\, \frac{\partial \Phi}{\partial y}\right) + S_\Phi + S_\Phi^P \tag{1}$$

where Φ stands for the different variables u_F, v_F, k and ε. The terms S_Φ and Γ represent the source term and the effective diffusion coefficient, respectively, and S_Φ^P represents the source term due to the momentum exchange between phases. This last term is calculated by solving the Lagrangian equation of particle motion using the PSI–cell–method [1].

$$S_{u_i}^P = -\frac{1}{V_{ij}} \sum m_P \dot{N}_P \left[u_{Pi,out} - u_{Pi,in} - g_i(1 - \frac{\rho_F}{\rho_P})(t_{out} - t_{in}) \right]$$

The Navier–Stokes solver is operating on structured, non–orthogonal, curvilinear grids with a bisectional refinement strategy for the construction of the various grid levels. Parallelization of the solution algorithm for the set of Navier–Stokes and turbulence model equations (standard k–ε–model) is performed by application of domain decomposition method to the block structure of the numerical grid as proposed by M. Peric in [6, 7].

The disperse phase is treated by the Lagrangian approach where a large number of particles are followed in time along their trajectories through the flow domain. The particle trajectories are determined by solving the ordinary differential equations for the particle location, the translational and rotational velocities. For the formulation of the particles equations of motion only the drag force, the lift force due to particle rotation (Magnus force) and the gravitational force are taken into account. It is assumed that other forces like the Basset history force can be neglected due to a small density ratio ρ_F/ρ_P. The effect of turbulence of the fluid flow on the motion of the disperse phase is modelled by the so–called Lagrangian stochastic–deterministic (LSD) turbulence model proposed by Schönung and Milojević [5].

Three different parallelization methods for the Lagrangian solution algorithm for the equations of motion of the disperse phase are described. The developed parallelization algorithms are applicable to both 2– and 3–dimensional multiphase flow calculations. The main problem in parallelization of Lagrangian solvers is the complex dependence between the fluid flow data and the data requirements of the solution algorithm for the particles/droplets equations of motion. The problem arises from the distributed storage of the fluid flow data over the processor nodes of the parallel computer system in accordance with the domain decomposition method. This data dependence has to be solved by an efficient parallel solution algorithm while introducing a minimum of inter–processor communication. A comparison of three parallelization methods is given for implementations based on PVM message passing library on workstation clusters and dedicated parallel MIMD computers. All three methods are based on a host–node parallelization model, where the host process generates the initial conditions for the dispersed particles and distributes them to the nodes for trajectory and source term calculation.

Method 1 : Each node process calculates trajectories of disperse particles and the corresponding source term contributions in the whole flow domain. Fluid flow data are stored redundant in the node memory of each processor node.

Method 2 : This method is based on the same distributed storage of the grid and fluid flow data as used by the grid partitioning method for the Navier–Stokes solver. One node processor is assigned to one grid block of the numerical grid. Then each node process calculates trajectories and source term contributions on the grid block assigned to this processor node. Particle state at the interface to a neighbouring grid block is submited to the appropriate processor node for continuation of the calculation process.

Method 3 : The same distributed storage of the grid and fluid flow data as in method 2 is used. But in contrast to the second method a processor node calculates a particle trajectory from its entry point to the flow domain to its final exit location at an outlet cross section. While the particle is moving in the processors "own" grid block fluid flow data needed for the particle trajectory calculation can be taken from the processors local node memory. If the trajectory leaves this grid block, fluid flow data have to be made available by node communication. this is accomplished by starting an additional memory handler task on each processor node. The memory handler provides fluid flow data of the grid block corresponding to the processor node number and waits in a permanent message

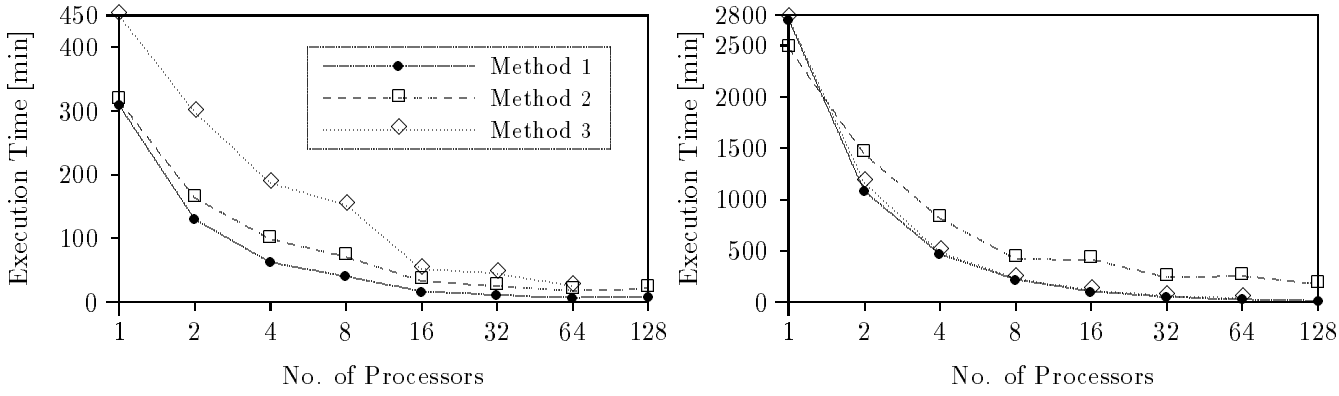

Figure 1: Execution time for parallelization method 1–3 for both test cases.

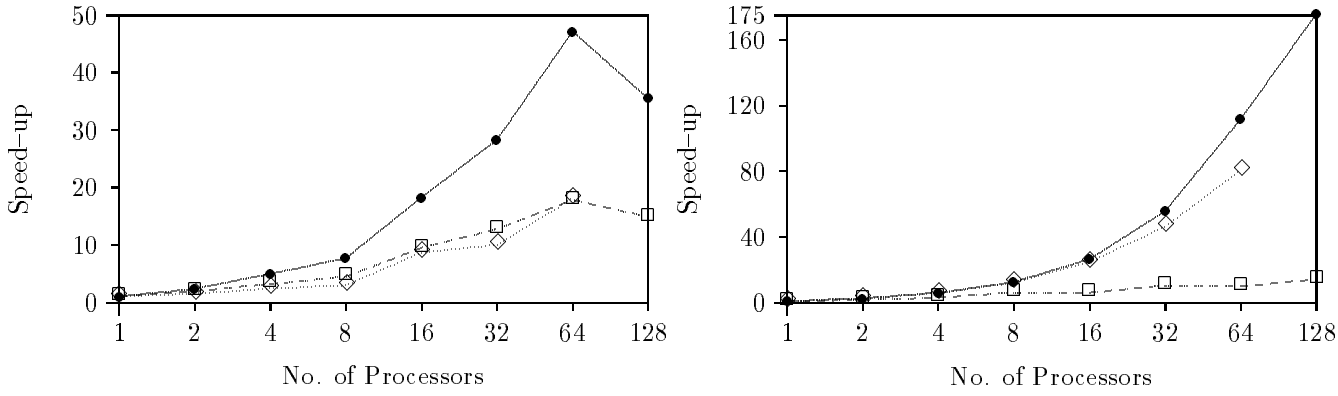

Figure 2: Speed–up for parallelization method 1–3 for both test cases.

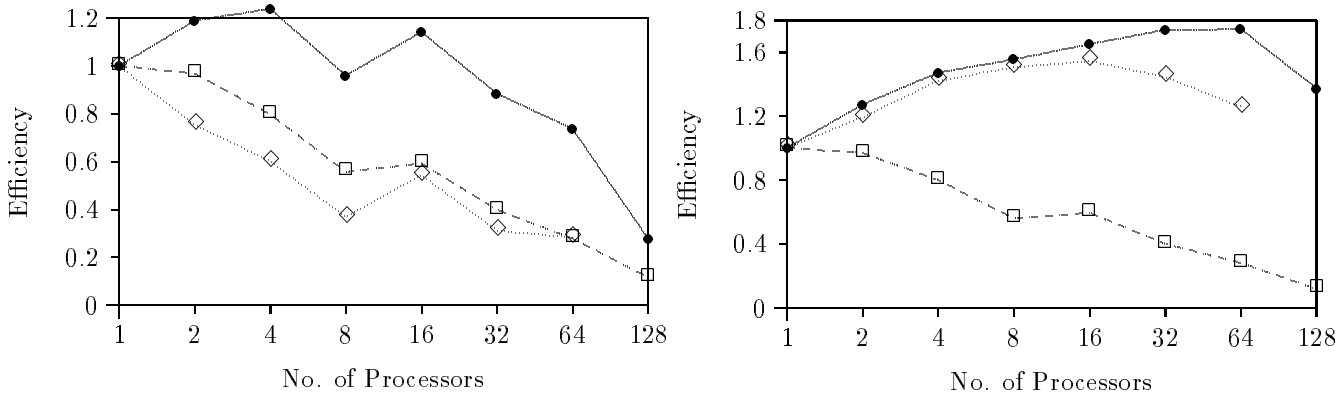

Figure 3: Efficiency for parallelization method 1–3 for both test cases.

loop for requests from other processor nodes. This way requests for fluid flow data can be answered using normal inter–processor communication and without interruption of the particle trajectory calculation of the node program.

The three described parallelization methods were implemented using PVM message passing library on a FDDI–networked workstation cluster and on a massivly parallel MIMD computer Parsytec Power–GC–128. Further the 3 parallelization methods were applied to 2 test cases, which are typical for 2–dimensional simulations of disperse multiphase flows (see Fig. 1—3 for the results of the test cases). In respect to the achieved test case results it is possible to point out the main advantages and disadvantages of the different parallelization methods. So the third method seems to be the most advantageous parallelization method for Lagrangian multiphase flow calculations among the methods with distributed storage of fluid flow data, what is essential for use of MIMD computers. The algorithm of this method can be further enhanced by various caching and look–forward algorithms for the transfered fluid flow data. Corresponding results will be presented at the conference.

References

[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] Frank Th., 1992, "Numerische Simulation der feststoffbeladenen Gasströmung im horizontalen Kanal unter Berücksichtigung von Wandrauhigkeiten", Ph. D. Thesis, Bergakademie Freiberg, Fak. f. Techn. Wissenschaften.

[3] Frank Th., Schulze I., 1994, "Ein numerisches Verfahren zur Berechnung disperser Mehrphasenströmungen auf parallelen Hochleistungsrechnern", Arbeitssitzung des GVC–Fachausschusses Mehrphasenströmungen, 17./18. Februar 1994, Würzburg, Germany.

[4] 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, 19.–23. Juni 1994, Lake Tahoe (NV), USA.

[5] Milojević D., 1990, "Lagrangian Stochastic–Deterministic (LSD) Predictions of Particle Dispersion in Turbulence," *Part. Part. Syst. Charact.,* Vol. 7, pp. 181–190.

[6] Perić M., 1985, "A Finite Volume Method for the Prediction of 3–dimensional Flows in Complex Ducts", Ph. D. Thesis, Imperial College, University of London.

[7] 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.

[8] 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.