

OpenMP optimisation of a parallel MPI CFD code

Fabrice Mathey, Philippe Blaise and Philippe Kloos

CEA/DI, CEA Saclay - Bat. 474, 91191 Gif sur Yvette, France
mathey@cea.fr

1 Introduction

Despite the progress in numerical algorithm and computer power, reliable CFD prediction of complex flows as turbulent flows is still a challenge. Numerical turbulent prediction (Large Eddy Simulation, see [7] for more details) computes the three-dimensional, unsteady turbulent flow directly by use of the Navier-Stokes equations, with a model utilized only for the small scales of motion that are unresolved by the computational mesh (subgrid-scale model). The application of LES to industrial flows is highly attractive as the present increase in computing power of computers brings the necessary computational resources within the reach of industrial research and design groups. However, many situations still require High-Performance Computers. This requirement is due to the use of fine meshes with a huge number of points. A typical simulation requires around 10^6 to 10^7 grid points. In turn, this implies a small size of the time step, increasing the number of iterations needed to compute the flow evolution and the mean quantities of interest.

In a previous work [1], the highly vectorized code LESOCC (Large Eddy Simulation On Curvilinear Coordinates) based on a Finite-Volume method has been parallelized by applying domain decomposition and explicit message passing (MPI). Good results were obtained on a Vector Parallel mainframe (Fujitsu VPP 300). Indeed most of the computing time spent inside the linear SIP solver (Strongly Implicit Procedure [4]) which is used to solve the Poisson equation for the pressure correction. This algorithm is very well suited for vectorization, as very huge loops are involved during the computation. On the contrary, it was shown to be less efficient when implemented on a massively parallel machine.

The focus of this study is on performance test of an OpenMP-MPI implementation of the code on a distributed Shared Memory system: a high performance SMP cluster of 58 Compaq Alpha server ES40 (ev67 chips) installed at the High Performance Computing Center of the Commissariat à l'Energie Atomique in Grenoble (CGCV). The objective is to analyze the scalability of the code when OpenMP directives are used to improve the efficiency of the code by reducing decoupling between each part of the domain. In particular, it seems possible to benefit from previous work made for vec-

torization purpose in order to parallelize the large loops of the solver within each node.

2 Numerical Method

The Large Eddy Simulation code LESOCC developed at the Institute for Hydromechanics [2,3] is based on a Finite-Volume method for solving the incompressible Navier-Stokes equations on general body-fitted, curvilinear grids (LESOCC = Large Eddy Simulation On Curvilinear Coordinates). A non-staggered, cell-centered grid arrangement is used. Both convective and viscous fluxes are approximated by central differences of second order accuracy. The temporal discretization consists of an explicit low-storage Runge-Kutta method (second order in time). The conservation of mass is achieved by a standard pressure correction algorithm (SIMPLE). Pressure is updated at each iteration of an outer loop until mass conservation is reached. The pressure correction variable itself is calculated from a Poisson equation. The Poisson equation is solved by the iterative Strongly Implicit Procedure (SIP), which constitutes the inner loop. In order to avoid decoupling of pressure and velocity on the non-staggered grid the momentum interpolation method of Rhie and Chow [5] is applied. The code is highly vectorized and has been validated extensively [2].

3 OpenMP-MPI implementation

The code has been extended to block-structured grids for parallel execution on distributed memory machines [1]. MPI Parallelization is accomplished by domain decomposition and explicit message passing. Explicit data exchange of the parallel domain boundaries takes place after each iteration of the inner loop inside the implicit step, the SIP procedure. The advantage of this method is to reduce the amount of communication between the parallel process during the implicit step. In turn, this introduces a weak decoupling between each parallel domains. On vector parallel mainframe, high performance can be achieved with the vectorization of the loops which takes place inside the SIP solver. Hence a small number of process are sufficient and good convergence rates are achieved. However, on a massively parallel mainframe, the decoupling of the domain and the increase of parallel boundaries can slow down the convergence. As a result, more than 60% of the computing time is usually spent inside the SIP solver.

With a distributed Shared Memory system, it is possible to take advantage of the vectorization (linearization of the loops) in order to parallelize each loops inside a sub-domain. Hence OpenMP directives have been used to parallelize each 1D loops within the SIP solver, and also within the viscous fluxes calculation routines. It should be noted that only a few directives have been included, leading to a very short time development (around one day).

4 Computational set-up and flow calculations

A two-dimensional channel flow of high $2H$ is simulated for purpose of validation and performance measurement of the parallel algorithm. A computational domain of size $6.4H \times 2H \times 3.2H$ is considered for the simulation, using periodic boundary conditions in the streamwise and in the spanwise direction for the velocity field. The Werner and Wengle [6] wall function approach is employed at solid walls, and the Smagorinsky subgrid-scale model is applied to model the effect of the subgrid scale motion.

The Reynolds number of the flow based on half the height of the channel and the bulk velocity U is $Re = 10000$. The flow is characterized by the presence of time evolving (turbulent) instabilities which pass through the parallel boundaries. This flow calculation correspond to a common test-case for turbulent flow solver validation.

Different calculations will be performed on a high performance SMP cluster of 58 Compaq Alpha server ES40, with various number of MPI processes and OpenMP threads. The results of speed-up comparisons which are not yet available will be reported during the conference.

References

1. F. Mathey, J. Fröhlich, W. Rodi (1998), Large Eddy Simulation of a flow in a channel with surface mounted cubical obstacles, *Proceedings of the Workshop on Refined Flow Modeling*, Istanbul, August 98.
2. M. Breuer, W. Rodi (1994) Large eddy simulation of turbulent flow through a straight square duct and a 180° bend. In P.R Voke, R. Kleiser, and J.P. Chollet, editors, *Fluid Mech. and its Appl.*, **Vol. 26**, Kluwer Acad. publ., pp. 00-00
3. M. Breuer, W. Rodi (1996) Large eddy simulation of complex turbulent flows of practical interest. In E.H. Hirschel, editor, *Flow simulation with high performance computers II*, **Vol. 52** of *Notes on Numerical Fluid Mechanics*, pp. 258-274. Vieweg, Braunschweig.
4. H.L. Stone (1968), Iterative solution of implicit approximations of multidimensional partial differential equations, *SIAM J. on Num. Anal.* **Vol. 21**, pp. 530-558.
5. C.M. Rhie, W.L. Chow (1983) A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, *AIAA Journal*, **Vol. 21**, pp. 1225-1532.
6. W. Werner, H. Wengle (1993), Large-Eddy Simulation of Turbulent Flow over and around a cube in a plate Channel, 8th Symp. on Turb. Shear Flows, (Schumann et al., eds.), Springer Verlag.
7. M. Lesieur (1990) *Turbulence in Fluids*, Kluwer Academic Publisher, 1990.