研 究 速 報

Study on Parallel Computation of CFD Problems

Joshua MORA ACOSTA* , Nobuyuki TANIGUCHI** and Toshio KOBAYASHI***

Large Eddy Simulation

Nowadays LES is the best technique for a wide range of engineering problems due to its application in complex geometry's and phenomenology's e.g. turbulence and combustion, without wasting too much time compared with DNS (Direct Numerical Simulation) and needing less experimental parameters than RANS (Reynolds Average Navier Stokes).

LES is based in the Navier Stokes equations filtered in space obtaining the so called Grid Scale values such as velocities, pressure and kinetic energy. Filtered N-S equations for an uncompressible three dimensional and unsteady flow are expressed in tensor form.

$$\frac{\partial \overline{u}_i}{\partial t} = 0 \quad \cdots\cdots (1)$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j^2} + \frac{\tau_{ij}}{\partial x_j} \quad \cdots\cdots (2)$$

Furthermore in the filtering process appears the term τ_{ij} which due to its nature must be modelled.We refer to the Sub Grid Scale values. A first approximation was proposed by Smagorinsky by modelling the Reynolds stress tensor $R_{i,j}$ and neglecting Leonard L_{ij} and Cross C_{ij} stress tensors.

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j} = L_{ij} + C_{ij} + R_{ij} \approx \nu_{SGS}\,\overline{S_{ij}} \quad \cdots\cdots (3)$$

$$\nu_{SGS} = (C_S \Delta)^2 \left|\bar{S}\right| \quad \cdots\cdots (4)$$

Here S_{ij} and $|S|$ are the strain rate tensor and its norm. And Δ denotes the filter width. C_s stands for the Smagorinsky constant with an estimated value of 0.1 for developed turbulent flows in channels. In complex geometry's, these equations are expressed in a curvilinear system.

In order to obtain good results, the discretization in finite difference is performed in a collocated grid with a modified convective term which improves stability in the simulation calculus [1] and coupling between velocities and pressure correction. Second order of accuracy is used but a fourth order is suggested as a good improvement.

Navier Stokes and mass equations are solved by the fractional step method as a time marching method where a first approximation (first step) of velocities is calculated explicitly using the Adams-Bashforth method for the convective term. The diffusive term is evaluated by Crank-Nicholson scheme and the overall equation is solved by point Jacobi.

From mass equation in the collocate grid is obtained the pressure correction equation which is solved by bicgstab [4]. After that, velocities are updated (second step). A preconditioner in the solver is desired in order to reduce time of computation of this equation. Incomplete LU decompositions [3] can be a feasible option.

Here are presented some results (fgures 1 and 2) of a turbulent channel flow case with Reynolds number 790, grid size 32 × 64 × 32 in stream, normal and span wise directions respectively with a concentration grid parameter equal 2 and its benchmark using DNS data supplied by Moser and Kim [2].

Periodic boundary conditions in stream and span wise directions have been applied while non slip condition in normal direction (walls). Initial conditions are DNS data with random fluctuations in all directions.

Parallel Computing

CFD (Computational Fluid Dynamic) codes contain some bottlenecks related mainly with the pressure correction equation solver. This fact is observed in the rapport of cpu time spent in this

*Heat Transfer Technological Center
Polytechnical University of Catalonia, Spain
**Information Technology Center, The University of Tokyo
***Institute of Industrial Science, University of Tokyo

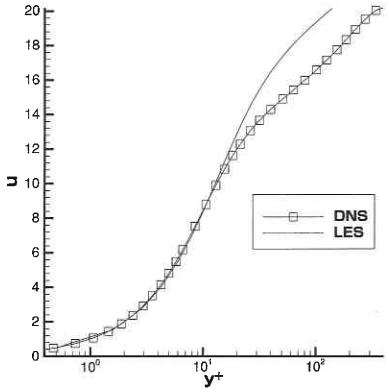

Fig. 1 Figure shows rms stream wise velocity profile near wall.

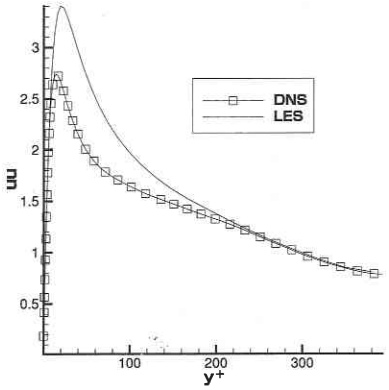

Fig. 2 Figure shows rms turbulent intensity prole in stream wise direction near wall.

small part of code and the overall code which can be estimated in 50–80%. Efforts in parallel computing are the most effective way of reduction of cpu time.

In this sense, a parallel implementation of jacobi and bicgstab [4] solvers has been done. Here are presented (fgures 3,4 and 5) effciencies of both solvers for different ways of partitioning.

It has been found a better performance in those cases when we do a partition in z direction or in y plus z directions instead of simply in x or y directions. We believe that a z direction partition breaks memory arrays in smaller memory arrays avoiding cache missing.

These tables and figures show efficiency of all possibilities of partitioning. We remark that the best partition contains y and z directions.

A second numerical experiment show the scaling properties of both parallel solvers when a given number of equations $N_1 = 32^3$ is increased by 2, 4, 8 and 16 and it is solved with 2, 4, 8 and 16 processors respectively.

In order to implement such solver into a CFD code, a structure

N_{CPU}	1	2	4	8	16
Efficiency X	1.00	0.96	0.88	0.80	0.72
Efficiency Z	1.00	0.99	0.96	0.92	0.86
Efficiency YZ	1.00	0.99	0.97	0.93	0.86
N_{CPU}	1	2	4	8	16
Efficiency X	1.00	0.97	0.89	0.79	0.61
Efficiency Z	1.00	0.99	0.95	0.83	0.65
Efficiency YZ	1.00	0.99	0.97	0.87	0.78

$N_{equations}$	N_1	$2N_1$	$4N_1$	$8N_1$	$16N_1$
N_{CPU}	1	2	4	8	16
$t_{CPU}jacobi$	699	1045	1145	1471	1673
$t_{CPU}bicgstab$	54	82	87	103	138

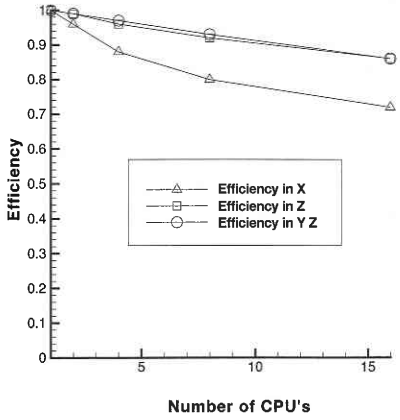

Fig. 3 Eciency of jacobi solver.

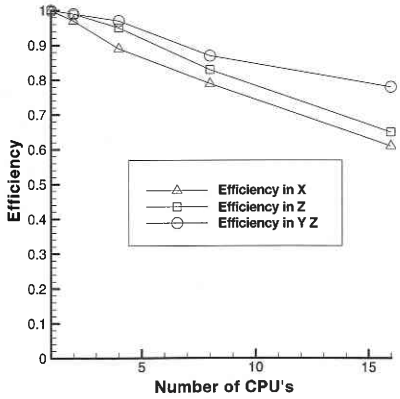

Fig. 4 Eciency of bicgstab solver.

of layers has been designed: communication layer, algebra layer, interface layer and user layer. The first one, contains MPI (Message Passing Interface) [5] subroutines and keeps *hidden* to the final user. The word *hidden* means that the user cannot access directly to this layer or subroutines. Therefore, the user doesn't

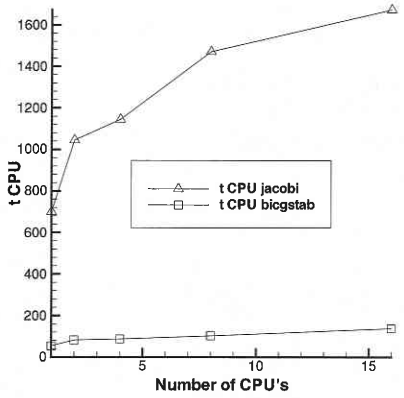

Fig. 5 Time of CPU in each solver when the number of equations and number of CPU's is increasing in the same rapport.

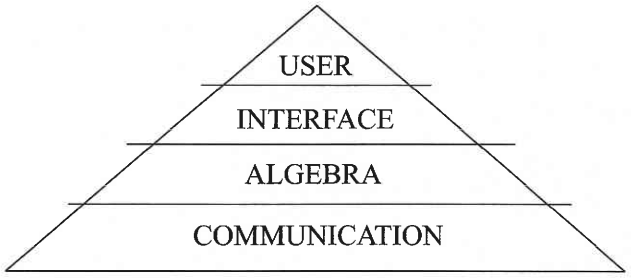

Fig. 6 Pyramid draft of a parallel implementation of a solver.

need to have any knowledge about parallel programming and SIMD (Single Instruction Multiple Data) paradigm. The scope of this layer consist in performing all communication tasks between processors. The algebra layer contains the solver and its auxiliary subroutines. This layer do calls to the previous layer when some operations must be performed in parallel, e.g. scalar product, norm vector, product of an array by a vector. This layer is also hidden to the user.

Finally the interface layer acts as a bridge between the user or the CFD code and the parallel solver. The user layer contains the problem and how to model it. Following this way the user can concentrate efforts in the simulation.

Wavelets and applications

In this section three different applications of a new revolutionary mathematical tool are presented. Due to its powerful features in the analysis of functions and sampled data wavelets are gaining more adepts in the scientific community.

Turbulence's phenomenology is difficult to describe and great efforts are being done in order to obtain good models. Richardson's cascade describe some important features of turbulence scales and the rapport between eddy's hierarchy. Such features can be well analysed by the use of wavelets [6] and [7]. Wavelets able us to extract the hierarchy of eddies easily from a DNS or experimental data, and study the influence and behaviour of each scale in the main flow.

Another interesting application of wavelets is in PIV (Particle Image Velocimetry) data. Data measurements with cameras contains some visualization errors which cannot be reduced: reflection, dirty surface. These errors must be erased before applying correlation between frames. At this point, wavelets acts as an error filtering process of those error measurements over each frame. Furthermore, a significant reduction of memory space is given without loos of accuracy. This reveals a new tool in the pre processing image step.

Finally, wavelet transforms can also be used in multigrid techniques. Transfer operators such a predictor and corrector are expressed algebraically in terms of a two dimensional wavelet transform applied over matrices and vectors [8].

References

1) Kogaki, T.,1999, *Finite dierent schemes in generalized coordinates and their application to large eddy simulation*. Ph.D.Thesis, University of Tokyo, Dept. of Mechanical Engineering.
2) Moser, R.D.,1999, *Direct numerical simulation of turbulent channel flow up to Re = 590* Physics of Fluids, Vol 2, Num 4.
3) Scheneider, G.E. & Zedan. M., 1981, *A Modified Strongly Implicit Procedure for the Numerical Solution of Field Problems*. Numerical Heat Transfer, vol.4, pp.1–19.
4) Vorst, H.A.,1993, *Krylov subspace methods for large linear systems of equations* Report of Mathematical Institute, University of Utrecht, The Netherlands.
5) *MPI, Message Passing Interface* www. mpiforum. org
6) Argoul, F.,1989, *Wavelet analysis of turbulence reveals the multifractal nature of the Richardson cascade*. Nature, Vol 338, pp. 51.
7) Hui, L.,1998, *Identification of Coherent Structure in Turbulent Shear Flow With Wavelet Correlation Analysis*. Transactions of the ASME, Vol 120, pp. 778.
8) Mora, J., 2000, *How to Use Wavelets as a Transfer Operators in algebraic MultiGrid*, Internal report, CTTC.

Acknowledgements

I would like to thank the members of the Industrial Institute of Science for different reasons. Professor Kobayashi who gave me the opportunity to visit the center. Professor Taniguchi who gave me his support in his laboratory. And finally P.h.D. and master course students who produced a friendly-working atmosphere during my stay.

(Manuscript received, November 22, 2000)