Mecánica Computacional, Volumen XIX, pp. 417-422
Compilado por: F. Quintana y S. Felicelli
San Carlos de Bariloche, Noviembre de 2000

FINITE ELEMENT PARALLEL COMPUTATIONS ON A BEOWULF CLUSTER CFD APPLICATIONS

Norberto M. Nigro, Mario A. Storti, Andrea. M. Yommi and Victorio E. Sonzogni

International Center for Computational Methods in Engineering (CIMEC)
Güemes 3450, 3000 Santa Fe, Argentina
e-mail: nnigro@intec.unl.edu.ar, Web page: http://venus.arcride.edu.ar/CIMEC

RESUMEN

Es bien conocida la necesidad de aumentar los recursos computacionales para poder abordar los problemas desafiantes del siglo entrante. Este inconveniente está siendo superado en los países desarrollados mediante el cálculo paralelo. Por nuestras limitaciones presupuestarias no tenemos en la actualidad acceso a supercomputadoras para poder seguir los últimos avances en materia de mecánica computacional. Debido a esto una de las pocas posibilidades de no quedar al márgen la ofrece la arquitectura tipo clúster cada vez mas atractiva aún para aquellos que cuentan con más recursos debido a su gran relación performance/precio. Por este motivo en el CIMEC hemos desarrollado un clúster tipo Beowulf. En trabajos anteriores se han mostrado las principales características técnicas de este clúster y se han realizado ensayos de performance en pos de optimizar la eficiencia en el cálculo.

Este trabajo tiene como finalidad mostrar algunas aplicaciones preliminares en CFD realizadas usando este clúster mencionando cuales serán nuestras futuras aplicaciones aprovechando esta gran fuente de potencia computacional.

ABSTRACT

It is well known that to solve the 21st century challenge problems we need to increase the computational resources In the developed countries this difficulty is circumvented by using parallel computation. Due to our limited budget we have no access to supercomputers to follow the fast development in the computational mechanics field. One of the few feasible current alternatives to follow that tendency is introduced by the concept of cluster, specially that based on domestic microprocessor, called Beowulf cluster, due to its high performance/prize ratio. Due to this fact we have developed our cluster at CIMEC. In previous papers we have shown the main features of our cluster including a performance analysis to measure the efficiency of this parallel architecture. This work focus on simple CFD applications in order to validate our parallel codes.

INTRODUCTION

It is well known the great variety of scales involved in fluid dynamic phenomena playing an important role in complex applications of scientific and industrial interest. Such a situation push the computer simulation to a very large scale problems where the computability is restricted only to parallel programming. Moreover, turbulence and 3D effects, among others, enhance this hypothesis

On the other hand, parallel computing was traditionally performed with very expensive and specific computers. This framework put underdeveloped countries out of the field forbidding to reach the treatment of the most challenge problems in areas such computational mechanics, among others. The fast evolution that PC processors are being experiencing with dramatic cost reduction and at the same time enhanced computing performance has lead many people to the idea of using this class of processors for parallel computing applications. On the other hand, most software for parallel applications like message passing libraries (PVM, MPI) are mostly Unix bounded, so that the use of PC class processors for parallel computations is linked to the generalised use of the GNU/Linux OS which is a free PC based Unix clone. Clusters of Intel x86 processors running GNU/Linux are commonly referred as Beowulf class clusters.

Our project put the target in the implementation of a general purpose, multi-physics library, running on Beowulf clusters and based on the MPI message passing library and the Parallel Extensible Toolkit for Scientific Computations (PETSc). The library is an abstract interface in which the *application writer* can define both the external algorithm (linear/non-linear, steady/unsteady) in term of assembling modules and the internal physics by programming one or several element routines. An object oriented language (C++) was chosen to write the whole program following the current tendencies in the computational mechanics community. In this work we present results for Newtonian and compressible or incompressible fluid flow problems in order to fix ideas about the computational capabilities of this architecture. Finite element with SUPG family methods, originally proposed by Hughes and Tezduyar, have been one of the most used numerical scheme for solving CFD problems. In this paper we have included results with the SUPG-PSPG version for the solution of incompressible viscous flow and the traditional SUPG plus shock capturing formulation for the compressible case. It is remarkable that we have also developed a fractional step algorithm for the incompressible Navier-Stokes equations but we have found less robustness and more user intervention in order to solve problems, specially due to the time step dependency for the solution of stationary problems and the difficulties associated with the fixation of boundary conditions for the intermediate steps. In general incompressible flow problems present two important difficulties for the solution with finite elements. First, the character of the equation becomes more and more advective dominant when the Reynolds number increases. In addition the incompressibility condition represents not an evolution equation but, rather, a constraint on the equations. This has the drawback that only some combination of interpolation spaces for velocity and pressure can be used, namely those ones that satisfy the so called Brezzi-Babuska condition. In the formulation of Tezduyar et.al. advection is stabilized with the well known SUPG stabilization term, and a similar stabilization term called PSPG is included in order to circumvent checkerboard modes. On the other hand for the compressible case the equal order interpolation is widely used but the first mentioned drawback related to the advection dominated problems remains as a spurious oscillation source. The incompressibility is not further involved as a constraint, but may be a source of ill-conditioning in regions where the nearly incompressible regime is established, like in stagnation zones or separation regions. Also, in the transonic case the shock put the equation system in a similar situation requiring some preconditioning in order not to deteriorate the convergence rate. Once these equations are spatially discretized, the resulting system of ODE's need some numerical integration scheme. We have used the standard backward Euler scheme (implicit) for incompressible flow and the forward Euler scheme (explicit) for the compressible one. In the former case, at every time step, the resulting non-linear system of equation is solved iteratively with the GMRES method with Jacobi left or right preconditioning.

NUMERICAL EXAMPLES

In this section we presents some of the numerical results used to validate our parallel codes. We begin with some numerical test for Euler equations to solve inviscid compressible flows.

Parabolic arc bump

This test concerns with a uniform flow that suddenly pass through a parabolic arc bump. For details about the whole data set we cite the work of Aliabadi et.al. [2]. This test was performed at three different Mach numbers, for subsonic M=0.5 regime, for the transonic condition M=0.84 and for the supersonic case M=1.4. Here we only include some features of the results of the two last cases. Figure 1 shows the pressure coefficient for the transonic case and figure 2 shows the density distribution of the supersonic one. In this last example it may be observed the presence of a compression wave that appears close to the leading edge of the bump and travel to the top boundary reflecting on it and it goes down toward the bump again. When it arrives close to the trailing edge it impinges against the expansion wave that appears in this zone producing a density profile with several fronts.

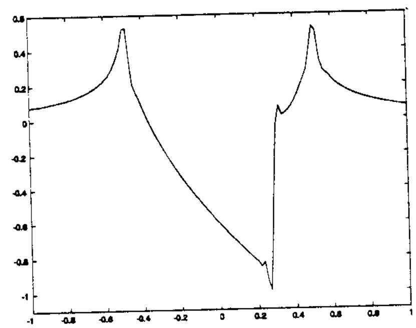

Figure 1: Parabolic arc bump. Transonic case M=0.84 . Pressure coefficient (Cp)

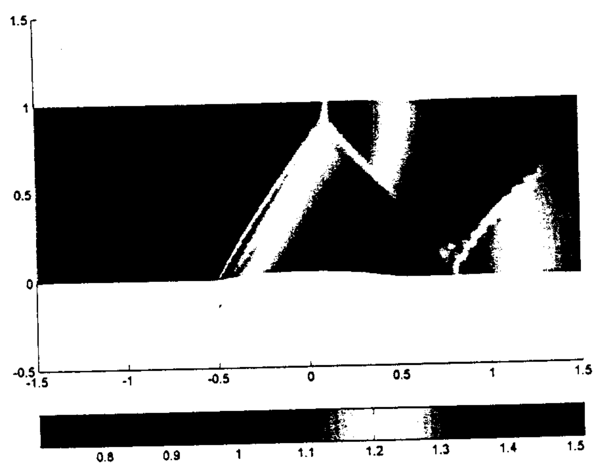

Figure 2: Parabolic arc bump. Supersonic case M=1.4 . Density field

Supersonic flow through a circular cylinder

This test is also included in the above cited reference [2]. It consists of a uniform supersonic flow impinging over a circular cylinder. For symmetry reasons we only compute half the domain and we observe in figure 3 the density field that presents a shock wave shifted upwards to the cylinder and it develops a sharp profile rounding the cylinder.

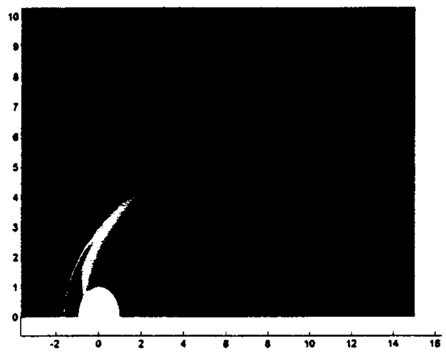

Figure 3: Circular cylinder. Supersonic case M=3.0 . Density field

Incompressible flow through a circular cylinder. Von Karman vortex street.

This example widely used as a test consists of a uniform incompressible flow passing through a circular cylinder at Reynolds number (Re) above the critical one. In this case Re=100 and a lot of reference with numerical results may be found in the literature. We have compared our results, based on a 30,000 elements mesh, with those published by Tezduyar [1] finding a good agreement and a Strouhal number close to 0.16.

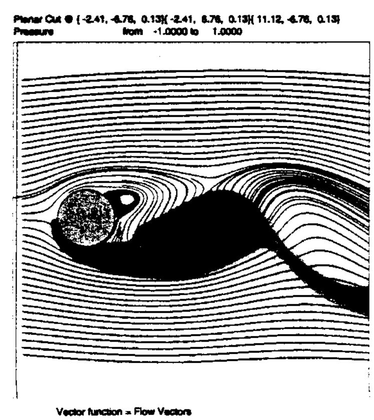

Figure 4: Circular cylinder. Incompressible flow at Re=100 . Streamlines

Incompressible flow in a cubic cavity.

This example consists of an internal incompressible flow inside a cubic cavity. Is an extension of the popular lid-driven square cavity to 3D geometry. We impose on the upper boundary a uniform velocity and the rest of the boundaries are solid wall for which non-slip boundary condition is imposed. Due to symmetry we compute only half of the cube, so we impose on this plane symmetry boundary condition.
Figure 5 shows some streamlines for a computation performed on a mesh of 32,000 elements.

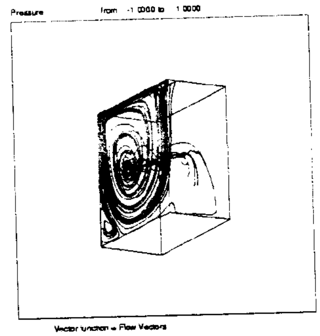

Figure 5: Cubic cavity. Incompressible flow at Re=1000 . Streamlines

Incompressible flow in a quarterbend.

Figure 6 presents a description of the quarterbend benchmark [3] where we have assumed a developed flow at the entrance of a squared section tube and non-slip boundary condition at solid walls. We have compute the flow only on the upper half of the tube using symmetry boundary conditions and a mesh of 50,000 elements.

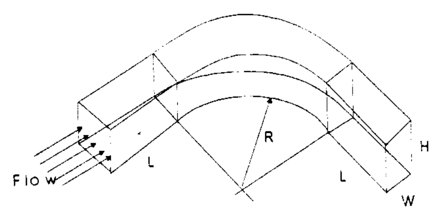

Figure 6: Quarterbend test. Incompressible flow at Re=100 & 500 .

Figure 7 and 8 show several velocity profiles, the former for Re=100 and the later for Re=500 at different stations along the bend. In figure 7 we may appreciate that the flow is developed inside the bend with the maximum peak value slightly apart from the centre of the tube. However, when Reynolds number increases above 100 some instabilities appear causing a secondary flow that deserves an experimental and numerical treatment out of the scope of the present work. As an interesting result we may observe in figure 8 that the flow is not developed at the end of the bend for this Reynolds number.

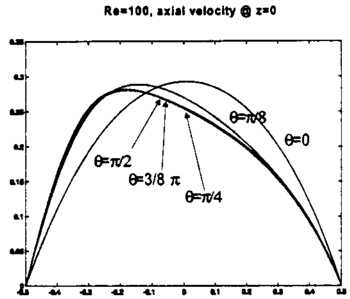

Figure 7: Quarterbend test at Re=100 Velocity profiles at different stages in the bend.

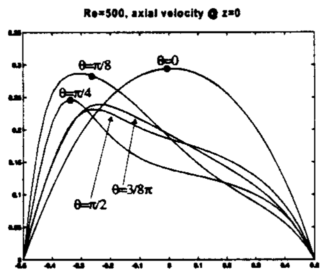

Figure 8: Quarterbend test at Re=500 Velocity profiles at different stages in the bend.

ACKNOWLEDGEMENTS

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET,Argentina) through grant BID 802/OC-AR PID Nr. 26, and from Universidad Nacional del Litoral (Argentina). For graphical post-processing we have used Visual3 Version 2.35 software from MIT

REFERENCES

[1] **Tezduyar T., Mittal S., Ray S. and Shih R.**, *Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity-pressure elements*, Comp. Meth. Applied Mechanics and Engineering, 95, 1992.

[2] **Aliabadi S., Ray S. and Tezduyar T.** ,*SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations* ,Computational Mechanics, 11, pp. 300-312, (1993)

[3] **Hassager O., Henriksen P., Townsend P., Webster M. and Ding D.**, *The quarterbend: A three-dimensional benchmark problem*, Computers & Fluids, 4, pp.373-386 (1991)