OBLIQUE–SHOCK–WAVE/BOUNDARY–LAYER INTERACTIONS ON THE QUADRICS PARALLEL COMPUTERS

B. FAVINI†, F. MASSAIOLI‡, R. BROGLIA†

† *Dip. Meccanica e Aeronautica, Univ. La Sapienza*
V. Eudossiana 18, I-00184 Roma

‡ *CASPUR, c/o CICS Univ. La Sapienza*
P.le Aldo Moro 5, I-00185 Roma
massaif@itcaspur.caspur.it

ABSTRACT

The present paper aims to analyze the fluid dynamic problem of oblique–shock–wave/laminar–boundary–layer interaction. Our concerns regard the influence of boundary conditions, the stability of the time integration technique, the actual accuracy of the discrete model, and the suitability of the algorithm to the Quadrics architecture.

1 Mathematical Model and Integration Technique

The Navier–Stokes equations define the mathematical model of the phenomenon: the Stokes' hypothesis is assumed for the viscosity second coefficient, while the viscosity first coefficient is governed by the Sutherland Law, a good approximation in the temperature range realized in the present study. The Navier–Stokes equations are numerically solved in integral form by a fractional step techniques. The operator is splitted into two parts, the Euler subsystem and the Dissipative operator: an explicit four step predictor–corrector technique has been adopted [1]. The essential properties of the integration scheme are: *i*) the Euler operator is integrated by an ENO–type scheme with second order accuracy both in time and space; *ii*) the reconstruction step is performed in terms of primitive variables; *iii*) the Riemann problems are solved exactly [2] or approximately [3]; *iv*) the Dissipative operator is integrated by a centered scheme second order accurate both in time and space; *v*) the time integration step is chosen in order to satisfy the stability of each operator; *vi*) the boundary conditions are imposed following the approach proposed in [4].

2 Implementation on the Quadrics Architecture

Parallelization has been achieved with a porting of an existing Fortran 77 code on the Alenia Spazio Quadrics architecture. This is an high performance and cost effective SIMD architecture, directly derived from the INFN APE100 supercomputer family. The machine [5] is built with single precision FPU nodes, ranging in number from 8 to 2048, and a single controller responsible of integer operations, addressing and program flow control. Every FPU has 4 or 16 MB of local memory and yields a peak performance of 50 MFlops. The FPUs are arranged on a 3D cubic mesh and the machine is capable of parallel synchronous communications between nearest neighbouring FPUs at speeds up to 12.5 MB/s per link.

The algorithm lends naturally to a multiblock parallelization, i. e. every FPU takes charge of a patch of the whole computational grid, with local boundary conditions obtained from the neighbouring FPUs. We used a Q1 (8 nodes, 400 MFlops peak) machine for development and small grids, and a Q16 (128 nodes, 6.4 GFlops peak) machine for bigger grids. The first version of the parallel code yielded a speedup factor per computational cell of 1.4 on the Q1

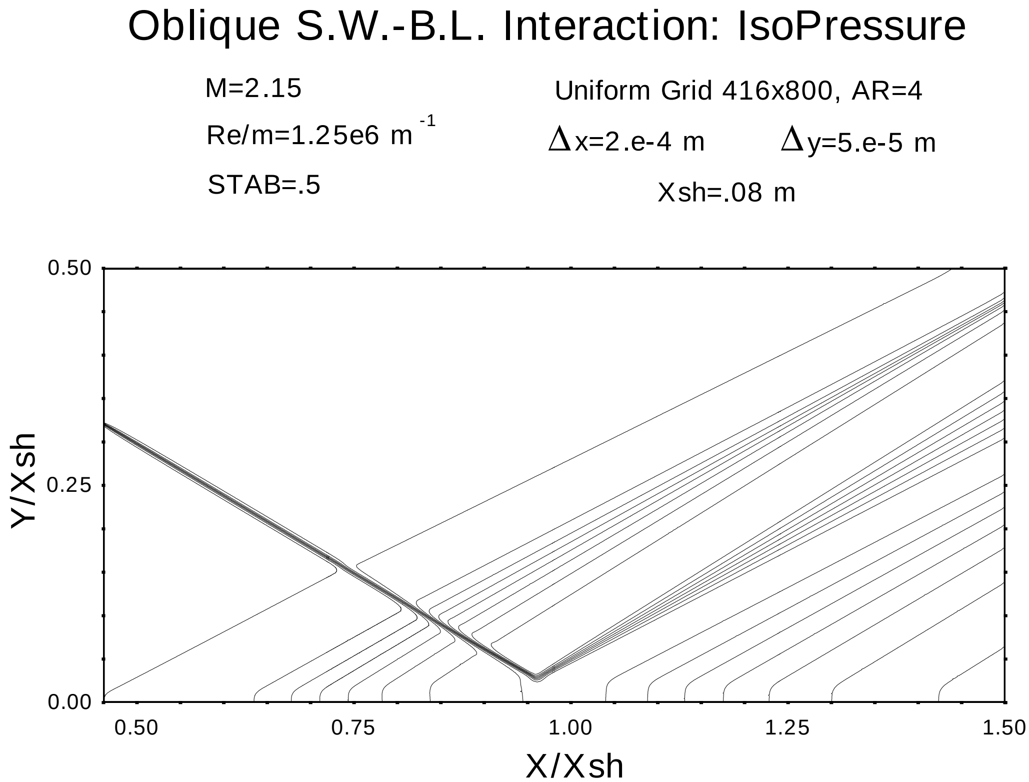

Figure 1: *Pressure field, normalized to the pressure at the infinity. The isolines range from* 1.0 *to* 1.5*, with* $\Delta = 3.6 \times 10^{-2}$

and of 19.6 on the Q16, with respect to a Sun SPARCstation 10/41, on small and medium sized grids. Very large grids take advantage of the large real memory of the Q16 (512 MB) and do not suffer from the huge amount of pagination which hampers a workstation.

Two factors affect these results. First, the Quadrics FPUs are mainly highly pipelined multiplier and adder devices, and the algorithm isn't able to sufficiently fill the pipe. Second, the original program adopted an exact solver of the Riemann problem [2], which finds a first guess of the solution with an iterative method. This badly affects a SIMD machines, because the cost of every step is dictated by the node with the slowest convergence. The switch to an approximate solver [3] (which didn't affect the numerical results in this class of problems), while resulting in a 10% speed improvement on the workstation, doubled the speed of the Quadrics code.

3 Numerical Results

The essential features of the two-dimensional interaction of an oblique shock with a laminar boundary layer on an adiabatic flat plate are (Fig. 1): *i*) the pressure increase across the shock influences the boundary layer, which results retarded and, for sufficiently strong pressure rise, as in the present case, separated; *ii*) upstream the oblique shock impinging location, the boundary layer growth induces a compression fan, which eventually degenerates in a shock; *iii*) another compression fan originates downstream where the boundary layer thickness decreases; *iv*) between these two compression waves, the impinging shock reflects as an expansion fan, indicating that the boundary layer reacts as a constant pressure surface. The recirculating bubble presents an extremely flat structure, corresponding to a plateau

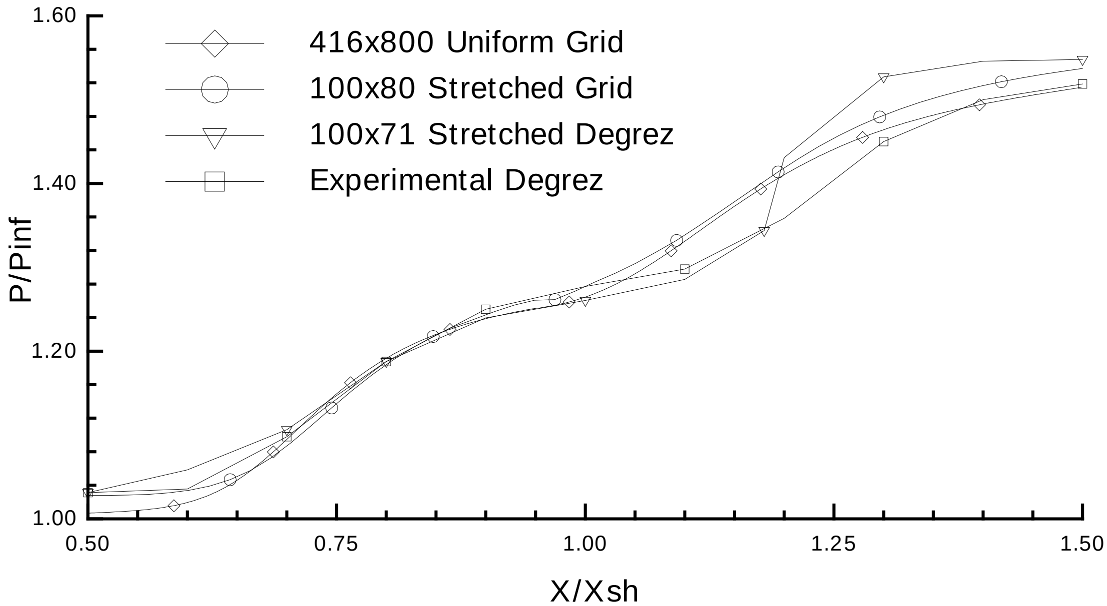

Figure 2: *Comparison of the pressure distributions along the plate in the two simulations discussed with the experimental and simulation data reported by Degrez*

in the pressure distribution along the plate (Fig. 2). The wall shear stress has generally a peculiar behaviour inside the reverse flow region where a local relative maximum appears corresponding to an asymmetric structure of bubble.

We compare the numerical results obtained with two different grids: *i*) 100×80 grid, highly stretched in the y–direction, uniformly spaced in the x–direction, and a minimum cell $(\Delta x, \Delta y) = (1.6 \times 10^{-3}, 5.0 \times 10^{-5})$; *ii*) 416×800 grid, uniform resolution in both direction (its features are given in fig. 1). In the latter case the integration domain has been reduced and, therefore, the resolution increases by a factor of 10, considering the number of cells enclosed in the separation bubble as a criterium for measuring the fineness of the grid.

The pressure distribution at the plate (Fig. 2) shows that the fine grid furnishes only minor improvement with respect to the coarse one; particularly the critical region at the end of the recirculating bubble, where flow reattaches, is not sufficiently resolved. A strong discrepancy in the reattachment region is exhibited with respect to the numerical solution presented in [6], where a grid corresponding to the present coarse mesh was adopted, using a quite different integration tecnique. The difference appearing in the inflow region depends on the fact that the integration domain for the fine grid starts on the plate and the boundary layer solution is assumed as inflow conditions, whereas for the coarser grid the integration domain starts upstream of the plate and a weak shock occurs at the plate leading edge causing a small pressure rise. The dimension of the separation region in the present computation is egual to 36.8 mm to be compared with 40 mm reported in [7] and 35.2 mm reported in [6]. The analysis of the friction coefficient confirms that the major differences are located in the reattachment region.

4 Conclusions

The preliminary numerical results obtained have shown that the resolution needed for a satisfactory simulation of S.W.–B.L. interactions has to be increased. The solution is deeply influenced by the boundary conditions, in the sense that a higher resolution reduces the numerical dissipation and emphasizes the strenght of the reflected waves at the boundaries: this fact opens some questions about the technique adopted for imposing the boundary conditions and more numerical experiments are required. The stability limit has been progressively reduced for increasing grid resolution: at present we are not able to give any explanation, although we charge the dimensional splitting nature of the discrete model.

The implementation on the Quadrics architecture can be improved. A new version is in the works, taking full advantage of the capability of the Quadrics TAO language [8] to be freely modified and augmented in the source code. The new version, written in an object based fashion, has one fourth the lines of codes than the original one, is highly modularized and more flexible. The preliminary results show that this approach allows for efficient and localized optimizations in the code of the newly defined operators, thus allowing for a better exploitation of the Quadrics power. A 3D version of the code is also planned.

Acknowledgements

The authors thank the INFN APE100 Project group, especially R. Tripiccione and R. Sarno, for useful discussions, and G. Todesco for the invaluable support to this work. The authors thank INFN Sez. di Pisa, where part of the simulations were performed.

References

[1] Di Mascio, A. and Favini, B., in preparation.

[2] Gottlieb, J.J. et al., J. Comput. Phys., **78**, 1988, 437.

[3] Roe, P.L., J. Comput. Phys., **43**, 1981, 357.

[4] Poinsot, T.J. and Lele, S.K., J. Comput. Phys., **101**, 1992, 104.

[5] Alenia Spazio S.p.A. *Introductory Notes to the Quadrics Parallel Supercomputers Family*, ALS/MKG011/1.0

[6] Degrez, G. et al., J. Fluid Mech., **177**, 1987, 247.

[7] Katzer, E., J. Fluid Mech., **206**, 1989, 477.

[8] Alenia Spazio S.p.A., *The TAO Language*