A Parallel Unsteady RANS Code for the Numerical Simulations of Free Surface Flows

Riccardo Broglia, Istituto Nazionale per Studi ed Esperienze di Architettura Navale, Roma, Italia
Andrea Di Mascio, Istituto Nazionale per Studi ed Esperienze di Architettura Navale, Roma, Italia
Giorgio Amati, Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Universitá e Ricerca, Roma, Italia

SUMMARY

The parallel code developed at INSEAN for the numerical simulations of unsteady free surface flows is presented in this paper; two test cases are considered as examples of applications and to analyze the performances of the parallel implementation. In the first, the turbulent flow past a circular cylinder is simulated; the Reynolds number is 50,000, the flow is computed in the framework of the Detached-Eddy-Simulations. In the latter, the flow around a ship hull with a practical hull form in a prescribed "pure sway" manoeuvre is presented; free surface effects are taken into account. For both tests, results are compared with both experimental and numerical data, code performances are evaluated by the measurement of the speed up factors and by a preliminary analysis of the required CPU time.

1 INTRODUCTION

In this paper the general purpose code developed at INSEAN for the numerical simulations of the unsteady turbulent free surface flow is presented; focus is on the measurements of the performances of the parallel version of the code. Among a large number of numerical simulations that are performed at INSEAN daily, in this paper two test cases are taken as example.

The first is the numerical simulation of the turbulent flow around a circular cylinder; in this test case the geometry involved is very simple, nevertheless, an accurate simulation is hard to obtain being the flow field very sensible to all the parameters involved (the computational grid, the turbulence model, the adopted scheme, etc...). This numerical simulation has been carried out in the framework of the Detached-Eddy-Simulations and it has been also used to validate the implementation of the DES model in the code. Due to the very simple geometry, the load among available processors (in term of number of point per processor) could be balanced ideally. Moreover, since in this case overlapping grid capabilities were not used, the communications between computational nodes involved only values at the boundary of the blocks, and therefore it was easy to balance the communications exchanged between processors.

The efficiency of the parallel code has been also evaluated in the more practical case of the numerical simulation of the free surface flow around a manoeuvring ship; in this test case, ideal balancing could not be obtained. Moreover, due to the complex geometry and to the movement of the ship hull, overlapping grid capabilities with the research of donors at each time step are involved. This means that a lot of communications between processors are involved and cannot be easily balanced.

The paper is organized as follow: first the mathematical model and the numerical method are briefly recalled, then the numerical simulations will be presented and compared with available experimental data and with similar numerical simulations. In the last section, the efficiency of the code will be shown; conclusions and future works will end the paper.

2 MATHEMATICAL AND NUMERICAL MODEL

The governing equations for the unsteady motion of an incompressible viscous fluid are briefly recalled in this section. The equations are written in an inertial frame of reference; as some blocks on the total grid move to follow possible moving boundaries, the general form of the governing equations are written with respect to a moving control volume. The equations in non-dimensional integral form (with respect to a reference velocity $U_\infty$ and a reference length $L$) are

$$\int_{S(V)} U \cdot n \, dS = 0$$

$$\frac{\partial}{\partial t} \int_V U \, dV + \int_{S(V)} (F_c - F_d) \cdot n \, dS = 0$$

(1)

$V$ being a control volume, $S(V)$ its boundary, and $n$ the outward unit normal. In equation (1), $F_c$ and $F_d$ represent convective (inviscid and pressure) and
diffusive fluxes, respectively:

\[ F_c = p I + (U - V) U \]
\[ F_d = \left( \frac{1}{Rn} + v_t \right) \left[ \text{grad}U + (\text{grad}U)^T \right] \]

where \( V \) is the local velocity of the control volume boundary, \( Rn = U_\infty L/\nu \) the Reynolds number, \( \nu \) the kinematic viscosity, whereas \( v_t \) denotes the non-dimensional turbulent viscosity. In the previous equation and in what follows, \( u_i \) is the \( i \)-th Cartesian component of the velocity vector (the Cartesian components of the velocity will be also denoted with \( u, v, \) and \( w \)); \( p \) is a variable related to the pressure \( P \) and the acceleration of gravity \( g \) (parallel to the vertical axis \( z \), downward oriented) by \( p = P + z/Fn^2 \), \( Fn = U_\infty \sqrt{g/L} \) being the Froude number. In the present work, the turbulent viscosity has been calculated by means of the one-equation model by Spalart and Allmaras [1].

The problem is closed by enforcing appropriate conditions at physical and computational boundaries. On solid walls, the relative velocity is set to zero (whereas no condition on the pressure is required); at the (fictitious) inflow boundary, velocity is set to the undisturbed flow value, and the pressure is extrapolated from inside; on the contrary, the pressure is set to zero at the outflow, whereas velocity is extrapolated from inner points. At the free surface, whose location is one of the unknowns of the problem, both dynamic and kinematic boundary conditions are enforced.

### 2.1 Numerical Model

In this section, the numerical algorithm is only briefly recalled, for more details the readers are referred to the cited papers. For the numerical solution of the equations (1), the fluid domain \( D \) is partitioned into \( N_l \) structured blocks \( D^l \), each subdivided into \( N_i \times N_j \times N_k \) disjoint hexahedrons \( D_{ijk} \). In the numerical scheme adopted here, the blocks are not necessarily disjoint, by can be partially overlapped, as it will be explained in the following sections. Conservation laws are then applied to each finite volume:

\[ \sum_{s=1}^{6} \int_{S_s} U \cdot n \ dS = 0 \]
\[ \frac{\partial}{\partial t} \int_{V_{ijk}} U \ dV + \sum_{s=1}^{6} \int_{S_s} (F_c - F_d) \cdot n \ dS = 0 \]

where \( S_s \) is the \( s \)-th face of the finite volume \( D_{ijk} \), whose measure is \( V_{ijk} \).

The pressure-velocity coupling is obtained by means of the pseudo-compressibility [2] formulation of the Navier-Stokes equations:

\[ \frac{\partial}{\partial t} \int_{V_{ijk}} p \ dV + \beta \sum_{s=1}^{6} \int_{S_s} U \cdot n \ dS = 0 \]
\[ \frac{\partial}{\partial t} \int_{V_{ijk}} U \ dV + \frac{\partial}{\partial t} \int_{V_{ijk}} U \ dV + \sum_{s=1}^{6} \int_{S_s} (F_c - F_d) \cdot n \ dS = 0 \]

Convective and viscous fluxes in the momentum equations, as well as surface integral of the velocity in the continuity equation, are computed by means of trapezoidal rule. Velocity gradients required for the computation of the stress tensor at the interface, are obtained by means of the classical second order accurate finite volume approximation. The computation of the convective fluxes and the surface integral of the velocity in the continuity equation requires the evaluation of pressure and velocity at the face center. To this aim, a second order ENO-type scheme has been adopted [3]. These schemes were originally developed for compressible fluid flows, on the basis of the hyperbolic nature of the inviscid part of the Navier-Stokes equations. The extension to incompressible flows is possible when working in the pseudo-compressible formulation; details of the algorithm can be found in [4]. Very recent high order schemes have been implemented; in particular a third-order upwind scheme and a forth-order centered scheme have been successfully coded and validated [5].

The time derivative in the equation (4) is approximated by means of a second order accurate three-points backward finite difference approximation formula, whereas the integration with respect to the pseudo time is carried out by means of an implicit Euler scheme. The resulting system of equation is solved by means of an approximate factorization [6]. Local dual time step and a multi-grid technique [7, 8] have been used in order to improve the convergence rate of the sub-iteration algorithm.

The presence of the free surface is simulated by mean of a single-phase level-set approach; for the sake of conciseness the algorithm is not described here, the reader is referred to the works by Di Mascio et al. [9] and by Carrica et al. [10] for details on the steady an unsteady single phase level set.

The basic elements of the overlapping grid discretization (or “chimera” method) for both fixed grids and its extension to moving grids are briefly recalled here, for more details and examples of applications.
the reader is referred to [11, 12, 13]. The introduction of chimera capabilities in the RANS code is given through a modification of both the boundary conditions and internal point treatment for those zones where overlapping appears. In particular, besides the natural type (i.e., wall, symmetry, inflow, outflow, ...), a new type of boundary conditions, the “chimera” type, is added for those boundaries where the solution must be interpolated from other blocks. For this cell, the first step is to find a “donor” cell, i.e., a cell that contain the face center for which an interpolation is needed. In order to retain the best possible approximation, if a boundary cell fall within more than one donor cell, the smallest one is picked as basis for interpolation. Once the donor is identified, then a convex set of eight donor cell centers is searched, and a tri-linear interpolation is used to transfer the solution from one block to the boundary of the one under analysis. As to internal points, we look at first for possible overlapping for each cell centers. If this overlapping is found, the cell is marked as a “hole” only if the donor cell is “smaller” than the one we are considering. As for boundary cell centers, is more that one possible donors is found, the smallest is chosen as donor. Differently form standard chimera approaches, however, the cell marked as holes are not removed from the computation but the interpolated solution is enforced by adding a forcing term to the Navier-Stokes equations, in a “body-force” fashion. For the point that fall within a rigid body, the forcing term in the equation is analogous to the previous one, but the velocity in the forcing term is equal to the local velocity of the body.

When dealing with moving grids, of course the grid topology must be re-computed at each time step. In order to speed-up the algorithm, we can exploit the information we have form the previous time step to search donor cells and holes. In particular, for each face center on a chimera boundary, we start the search of the possible donor from the knowledge of the previous ones. For the “holes”, the first check is made to verify if the cell is still overlapped by a cell of the same block. When these initial approximations fails to yield a new donor set, the general search procedure is re-started.

2.2 PARALLELIZATION

The grain/fine parallelization of the unsteady RANS code has been achieved by distributing the structured blocks among the available non-shared (nodes) or shared memory (threads) processors, and by spreading the computational work to be done (mostly in terms of do loop) inside each block among available shared memory processors; the balance of the load among processors is left to the user; useful preprocessing tools, which allows the splitting of the structured blocks and the distribution of them among the processors, are available for a fine tuning of the load balancing. Communication between processors for the grain parallelization is obtained by using standard Message Passing Interface (MPI) library, whereas fine (shared memory) parallelization is achieved by means of Open Message Passing (OpenMP) library. Numerical details of the parallel implementation can be found in [14].

3 NUMERICAL SIMULATIONS AND CODE PERFORMANCES

Some numerical results obtained with the parallel code are presented in this section; these test cases have been taken under consideration in order to analyze the performances of the code. In particular two flow fields have been simulated: the turbulent flow around a circular cylinder, and the free surface flow around a tanker in a prescribed "pure sway" manoeuvre. In the first, since the overlapping grids capabilities have not be used, communications between blocks involve structured ghost cells values only, moreover, after the blocks are splitted, ideal balancing of the load can be obtained by using up to 32 processors, which allows a detailed analysis of the performance of the code in an ideal case. In the latter, the analysis of the performance is carried out for a more practical case: overlapping grids capabilities are used, therefore, communications between blocks involve both structured ghost cells values and internal non structured points; moreover, ideal balancing of the load cannot be achieved, and the use of a large number of processors implies the splitting of the blocks. In the following, the flow fields obtained are first presented, the numerical simulations will be also verified and validate against experimental data and compared with other numerical simulation when available. In the last subsection, the performance of the parallel code will be presented.

3.1 DES PAST A CIRCULAR CYLINDER

As a first test case, the turbulent flow around a circular cylinder has been considered; the diameter of the cylinder \(D\) and the free stream velocity \(U_\infty\) have been used as characteristic length and velocity. The high of the cylinder is \(2D\), periodic boundary condition has been enforced along the cylinder axis (vertical/z direction).
The Reynolds number is 50,000, simulation has been carried out in the framework of the Detached Eddy Simulations [15], with the one-equation by Spalart and Allmaras [1] model as the turbulence model for the RANS region. At this Reynolds number the flow is laminar up to the separation point, where transition takes place. In the simulation, the trip less approach [16] has been used in order to correctly predict the transition to turbulent; in this approach the inflow eddy viscosity is set to zero, whereas non zero values are used as initial conditions. As result, the model is able to predict correctly a self-sustaining eddy viscosity field in the wake, up to the separation region; the solution does not depend on the value used as initial condition, but depends, obviously, on the turbulence model used. Moreover, no information about the location of the transition or its extension have to provide by the user.

In figure (1) the computational mesh used for the simulation is shown. The outer boundary is placed about thirteen diameters far from the cylinder surface, distance that can be considered enough to avoid reflection from the boundary. The mesh is composed by three blocks, the inner block, the wake and the outer block. The number of grid cells in these blocks are $256 \times 128$, $128 \times 64$, and $64 \times 32$, in circumferential and normal to the cylinder direction, for a total of about 43,000 grid cells; 64 cells are uniformly distributed along the vertical direction.

Unsteady numerical simulation is carried out until a statistically steady state is reached (i.e. at time at which mean values and, at least, second order statistics have reached steady values). In figure (2), the time histories of the lift and drag forces (averaged along the vertical direction) are presented; initial solution is from a coarser grid (the zero value of the time denotes the start time of the computation on the finest grid), mean values are computed within the time window bounded by the dashed line (the sample time is around 185 time units). From the time history of the lift force, the presence of a von Kármán shedding phenomena is clear, as well as a modulation of the shedding itself. The drag coefficient does not show a clear periodical behavior. The reported time histories qualitatively agree very well with the results presented by Travin et al. [15].

An overview of the vortical structures are given in figure (3), the presence of a strong two-dimensional structures aligned with the axis of the cylinder, revel a von Kármán vortex shed-
ding; moreover, the presence of some vortical structures transverse to the von Kármán vortices, clear denote the three dimensionality of the flow field.

Fig. 3: DES past a circular cylinder: visualization of vortical structures ($\lambda_1 = 1.0$).

The average values of the Strouhal number $St$, the drag coefficient $C_D$, the root mean square of the lift coefficients $C'_L$, the base pressure coefficient $-C_{pb}$, the separation angle $\Theta_{sep}$, the length of the recirculation bubble $L_r$, the peak value of the $\overline{u'v'}$ Reynolds stress and its location $L_{vv}$ along the centerline, as well as the peak values of the $\overline{u'v'}$ and $\overline{u'^2}$ Reynolds stresses obtained from the present numerical simulations are presented in table (1). In the same table, results from other numerical simulation by Travin et al. [15] ($LS_1$, $LS_2$ and $LS_3$ denote their coarse, medium and fine computations for the same Reynolds number and geometry) and from a collection of experimental data [17, 18, 19, 20, 21] are reported for the sake of comparison. The agreement with experimental data is very good, for both first and second order statistics; the length of the recirculation bubble seems slightly overpredicted, as well as the peak values of the Reynolds stresses. However, the computed values are in the range of the reported numerical simulations; even if, the resolution of the present calculation is close to the finest computation performed by Travin et al. [15] (LS3 case), our computation are in better agreement with their calculation on the medium resolution (LS2). The differences can be due to the different numerical scheme adopted (cited authors use a fifth-order upwind scheme); however, Travin et al. [15] state that their finest calculation does not give the better agreement with experimental data. Deep investigation on the comparison with other numerical and experimental data, as well as, grid convergence study is required.

<table>
<thead>
<tr>
<th>Comp.</th>
<th>$St$</th>
<th>$C_D$</th>
<th>$C'_L$</th>
<th>$-C_{pb}$</th>
<th>$\Theta_{sep}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>0.20</td>
<td>1.17</td>
<td>0.43</td>
<td>1.12</td>
<td>80°</td>
</tr>
<tr>
<td>LS1</td>
<td>0.22</td>
<td>1.05</td>
<td>0.21</td>
<td>0.98</td>
<td>78°</td>
</tr>
<tr>
<td>LS2</td>
<td>0.21</td>
<td>1.26</td>
<td>0.48</td>
<td>1.28</td>
<td>80°</td>
</tr>
<tr>
<td>LS3</td>
<td>0.20</td>
<td>1.32</td>
<td>0.66</td>
<td>1.39</td>
<td>96°</td>
</tr>
<tr>
<td>Exp.</td>
<td>0.18</td>
<td>1.15</td>
<td>0.24</td>
<td>0.12</td>
<td>0.60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comp.</th>
<th>$L_r$</th>
<th>$L_{vv}$</th>
<th>$\overline{u'v'}$</th>
<th>$\overline{u'^2}$</th>
<th>$\overline{v'^2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>0.88</td>
<td>1.0</td>
<td>0.31</td>
<td>0.15</td>
<td>0.60</td>
</tr>
<tr>
<td>LS1</td>
<td>1.3</td>
<td>1.5</td>
<td>0.20</td>
<td>0.13</td>
<td>0.46</td>
</tr>
<tr>
<td>LS2</td>
<td>0.8</td>
<td>0.9</td>
<td>0.34</td>
<td>0.17</td>
<td>0.66</td>
</tr>
<tr>
<td>LS3</td>
<td>0.8</td>
<td>0.9</td>
<td>0.37</td>
<td>0.15</td>
<td>0.62</td>
</tr>
<tr>
<td>Exp.</td>
<td>0.44</td>
<td>0.75</td>
<td>1.1</td>
<td>0.25</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Tab. 1: Comparison between present computation, numerical results from [15] and a collection of experimental results.

Fig. 4: Side view of the KRISO tanker KVLCC2.

3.2 FLOW AROUND A MANOEUVERING TANKER

As a second test case, the flow field around the KVLCC2 tanker (figure (4)) during a prescribed manoeuvres is computed. The imposed motion is of pure sway, which simulates an experiment in the towing tank with the Planar Motion Mechanism (PMM). In the pure sway motion the model travels trough the tank on straight ahead course with constant velocity $U_c$, while it is forced to oscillate from side to side with zero yaw rate and sinusoidal sway velocity. The non dimensional transverse translation is given by $y(t) = S_{MAX} \sin(\omega t)$, where $\omega = 4.95$ and $S_{MAX} = 0.0566$ is the maximum lateral displacement, that correspond to a maximum transversal velocity for the model, that is $V_{MAX} = 0.28$. In order to have an accurate dis-
cretization in time, it has been chosen a fixed time step equal to 0.01 that means about 127 time steps per period. The Reynolds and Froude numbers of the tests are respectively $Rn = 3 \times 10^6$ and $F_n = 0.0643$.

Fig. 5: Flow around the KVLCC2 tanker: an overview of the chimera grid and a zoom in the transom zone.

The overlapping mesh is formed by 28 blocks, eight for the fixed background and 20 fitted around the moving hull. In figure (5) are shown the body-fitted blocks and the background ones for one half of the physical domain at the initial stage of the simulation. The chimera cells have been hidden for the sake of clearness but, as already explained in the previous section, they are not removed from the computation. In particular, it can be observed the clustering of cells around the free-surface position at rest and the division of the background in four zones: two buffer zones in inflow and outflow, a finer central zone wherein the hull will move with the pure sway motion and a coarse bottom zone where few relevant phenomena are expected to take place. In figure (5) a zoom of the transom zone is also shown. Despite the complex geometry of this zone, the use of overlapping blocks permits the construction of a high quality, almost orthogonal mesh. The global mesh is composed by 4.3M cells, 1.7M in the background and 2.6M in the body fitted meshes.

Fig. 6: Flow around the KVLCC2 tanker: wave pattern at $t = 0$, $T/8$, $T/4$, $3T/8$ ($z_{\text{MIN}} = -0.0013$, $z_{\text{MAX}} = 0.0022$, $\Delta z = 0.0001$, dashed lines denote negative values).
Fig. 7: Flow around the KVLCC2 tanker: isobaric on the hull, $p_{\text{MIN}} = -0.44$, $p_{\text{MAX}} = 0.64$, negative values with dashed lines.

The unsteady wave pattern is shown in figure (6) for the smaller amplitude of the motion at the four instants $t = 0$, $T/8$, $T/4$, $3T/8$. The wave pattern in the second half of the cycle, that is for $t = T/2$, $5T/8$, $3T/4$, $7T/8$ is symmetric with respect to the $x$-axis and is not shown for the sake of conciseness. In the top figure the hull is located at the centerline of the basin and is moving with its maximum transversal velocity towards the upper boundary. Due to the composition of the longitudinal and transversal velocities, a run-up is formed on the starboard side of the hull, just behind the bow, whereas on the port side a large throat can be noticed. After $T/8$ the velocity is decreased and correspondingly the run-up is less intense and the extension of the throat on the port side is also reduced. When the hull reaches the apex of its transversal motion at $t = T/4$, the stop generates a large throat on the starboard side, that will become even wider during the downward motion of the hull, while the run-up shifts to the port side.

Fig. 8: Flow around the KVLCC2 tanker: velocity field, $u_{\text{MIN}} = 0.0$, $u_{\text{MAX}} = 1.0$, $\Delta u = 0.1$.

Figure (7) illustrates the isobaric on the hull. For each instant, three views are shown namely from the bottom and on each side of the hull. In the topmost figure, at $t = 0$, the run-up on the starboard side is highlighted as a small, high pressure region just behind the bow, whereas the throat on the port side is marked by a low pressure zone. Moreover, a wide zone of low pressure on the starboard side of the bottom in the bow region marks the point where a high velocity flow turn around the bilge. Another low pressure zone is evident in the stern region, on the port side; it is evidently due to the cross flow detaching from the keel with the consequent formation of an intense vortex (see also figure (8)). This latter zone disappears as soon as the transversal velocity is reduced, see figure for $t = T/8$, whereas the run-up and the
low pressure zone around the bilge are less intense but still persistent. At \( t = T/4 \) the lateral velocity is zero and the pressure distribution on the hull seems roughly symmetric, except near the stern where a low pressure zone appears on the starboard side, probably due to inertial forces acting on the fluid.

In figure (8), the velocity field is reported. The view from the stern shows the longitudinal component on the free–surface, four sections of the field at \( x = 0.0, -0.4, -0.5, -0.6 \) (respectively addressed as S1, S2, S3 and S4 in the following) are presented. The analysis of the sections highlights the vortex pattern underneath. At \( t = 0 \) on S1 the generation of the bilge vortex on the starboard side can be noticed, particularly strong due to the cross flow generated by the sway motion. The trace of this vortex can also be spotted on S2, whereas the formation of a strong vortex where the flow turn around the keel is recorded on S3. This latter vortex embedded in the wake behind the hull is visible on S4. At the next instant the same vortices are still visible. For \( t = T/4 \), the formation of new vortices can be seen. A bilge vortex on the port side develops on S1, together with the one already seen on the starboard side. On S2, apart from the trace of the starboard bilge vortex, that persists also on S3, the formation of two vortices on the sides of the keel in the transom region, which are clearly due to the sudden restriction of the hull in this zone, can be appreciated. Instead, the strong keel vortex still visible at \( t = T/8 \) is no more identifiable. At the last temporal instant, the hull is already moving in the opposite direction and correspondingly the port bilge vortex strengthen, whereas of the starboard bilge vortex only a weak trace remains in S2, S3 and S4.

In figure (9), the time histories (over one and half period, when periodical regime is attained) of the computed lateral force and yaw moment coefficients are presented and compared with experimental data performed at INSEAN [22]; the agreement is fairly good, the error being about 5.5\%, and 20.0\% in the maximum values for the lateral force and for the yaw moment, respectively. Numerical simulation seems under predict both coefficients, with higher error for the yaw moment; however, it has to note that the moment coefficient is computed as a difference between similar measured values, which can be source of high error and uncertainty. Moreover, high uncertainty is related to the position of the PMM device, or in the origin of the axis, and a small error in one of these two parameters can lead to an high error in the computation or in the measurement of the yaw moment.

Fig. 9: Flow around the KVLCC2 tanker: time histories of the lateral force and yaw moment coefficient, comparison with experimental data.

3.3 PERFORMANCE OF THE PARALLEL CODE

In this subsection an overview of the performances of the parallel code are presented; up to now, only a preliminary study has been carried out, therefore, only results in terms of speedups and a global measurements of CPU time will be shown.

In figure (9), the time histories (over one and half period, when periodical regime is attained) of the computed lateral force and yaw moment coefficients are presented and compared with experimental data performed at INSEAN [22]; the agreement is fairly good, the error being about 5.5\%, and 20.0\% in the maximum values for the lateral force and for the yaw moment, respectively. Numerical simulation seems under predict both coefficients, with higher error for the yaw moment; however, it has to note that the moment coefficient is computed as a difference between similar measured values, which can be source of high error and uncertainty. Moreover, high uncertainty is related to the position of the PMM device, or in the origin of the axis, and a small error in one of these two parameters can lead to an high error in the computation or in the measurement of the yaw moment.

The speed up factor measured for the turbulent flow around the circular cylinder is reported in figure (10); the speed up is computed has the ratio between the total CPU time required by the serial code and the CPU time required by the parallel computation, the line of the theoretical speed up is also reported. In the figure, MPI denotes the number of non shared memory CPUs (nodes), whereas, with OMP is indicated the number of the threads used; the total of proces-
sors is the product of the MPI nodes and the OMP threads. Load balancing is obtained by splitting each original block in a number of blocks (with the same number of cells) equal to the available nodes; in this test case ideal load balancing could be obtained. Up to a total of 32 processors have been used. As it can be seen, by increasing the number of the of processors up to 32 and using only the grain parallelization (i.e. the MPI library), the speed up factor is very close to the theoretical one. Deviation from the theoretical value can be observed when 32 processors are used; however, it has to be taken into account that when a large number of computation nodes are used, the CPU time required by the communications becomes a large contribution to the total CPU time. The performances slightly degrade when shared memory parallelization (i.e. OpenMP directives) is used. However the efficiency is satisfactory.

The efficiency of the code has also been measured for the more practical case of the flow around a surface piercing hull with an enforced pure sway motion. For load balancing the 28 blocks are split in up to 76 blocks in total, with a maximum unbalance of $\sim 1.5\%$. This test case involves free surface computation, unsteadiness and overlapping grid motion, i.e. donors and weights for the interpolated values required by both chimera volume and face cells have to be computed at each time step. The non shared parallelization capabilities have not be used for this test case. The CPU time and the measured speedup are reported in figure (11); in this figure, CPU denotes the total CPU required by the simulation (measured on ten time steps), MAP denotes the CPU time required by the construction of the map for the communications, I/O denotes the time required by the Input/Output of the data, and COM denotes the CPU time required for the communications between processors. As it can be seen the efficiency of the code is satisfactory, the speedup being very close to the theoretical value. The speed up measured by doubling the number of processors is $\sim 1.9$, with a total speedup factor of 27.4 when using 32 processors. From figure (11), it can also be seen that the total CPU time remains almost constant, the increase being due to the increase in the time required by the communication.

4 CONCLUSION

In this paper results and performances of the parallel code developed at INSEAN for the numerical simulations of unsteady turbulent viscous free surface flows have been presented. Two tests cases were considered, namely the turbulent flow around a circular cylinder and a more practical example of the simulation of the flow past a tanker in a prescribed manoeuvre. Results compare very well with both experimental and numerical data available in literature; the performance of the parallel code is well satisfactory in term of speed up factor. A deep investigation of the performances is an ongoing activity; future works will involve the development of propeller models as well as the analysis of the hull/propeller interaction by simulation of the operating propeller in real geometry. A long term goal of the research activity is the development of numerical tool capable to simulate a fully 6DoF motion of a surface ship, with either prescribed or predicted manoeuvre.

Fig. 11: Flow around the KVLCC2 tanker: code performances.

5 ACKNOWLEDGEMENTS

This work is supported by the Italian Navy through the project ”MobiProp”. Numerical simu-
lations were carried out on the IBM System p5 575 machine at the supercomputing center CASPUR in Rome, in the framework of the project Simulazioni RANS di scafi navali con appendici mobili e propulsore ad elica. The authors wish to express their appreciation to Dr. Federico Massaioli for the useful help and discussions.

REFERENCES


