Mecánica Computacional Vol. XXIII, pp. 3119-3132
G.Buscaglia, E.Dari, O.Zamonsky (Eds.)
Bariloche, Argentina, November 2004

PARALLEL IMPLEMENTATION OF FREE SURFACE FLOWS

Laura Battaglia*, Jorge D'Elía*, Mario Storti*and Norberto Nigro*

*Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC)
Instituto de Desarrollo Tecnológico para la Industria Química (INTEC)
Universidad Nacional del Litoral - CONICET
Parque Tecnológico Litoral Centro (PTLC) s/n, 3000-Santa Fe, Argentina
e-mail: (jdelia, mstorti, nnigro)@intec.unl.edu.ar
web page: http://venus.ceride.gov.ar/CIMEC

Key Words: free surface flows, sloshing, mesh-movement, finite elements, large scale and distributed computing, fluid mechanics.

Abstract. *In this work, transient free surface flows of a viscous incompressible fluid are numerically solved with a parallel computation. Transient free surface flows are boundary-value problems of moving type that involves geometrical non-linearities. In contrast to CFD more conventional problems, the computational flow domain is partially bounded by a free surface which is not known a priori, since its shape must be computed as part of the solution. In steady-flow the free surface is obtained by an iterative process but the problem is more difficult when the free surface evolves with time, generating large distortions in the computational flow domain. In this work, an incompressible Navier-Stokes numerical solver is based on the finite element method with equal order elements for pressure and velocity (linear elements), and it uses a Streamline Upwind Petrov Galerkin (SUPG) scheme combined with a Pressure Stabilized Petrov Galerkin (PSPG) one. At each time step, the fluid equations are solved with constant pressure and null viscous traction conditions at the free surface. The velocities obtained in this way are used for updating the positions of the surface nodes. Then, a pseudo elastic problem is solved in the fluid domain in order to relocate the interior nodes so as to minimize mesh distortion. This has been implemented in PETSc-FEM by running two parallel instances of the code and exchanging information between them. A numerical example is presented.*

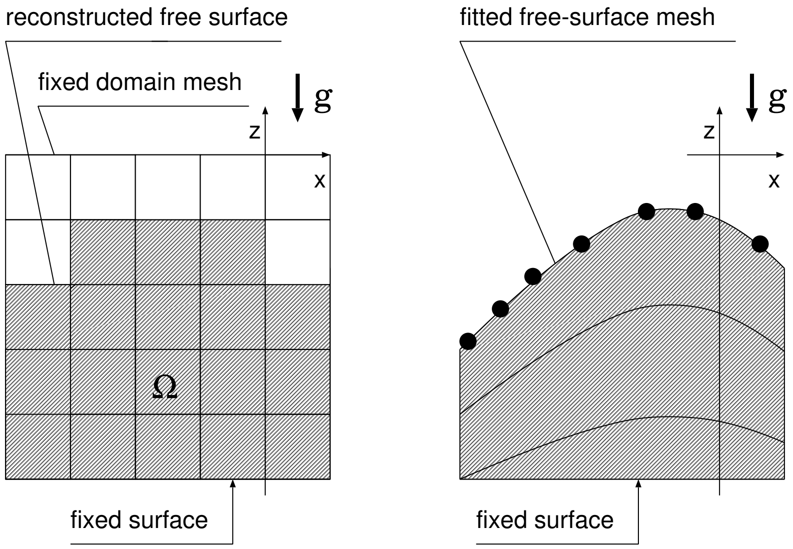

Figure 1: A flow domain with a free surface discretized by domain-like schemes: Euler-type (left) and Lagrangian-type (right) methods.

1 INTRODUCTION

In this work, transient free surface flows of a viscous incompressible fluid are numerically solved with a parallel computation. Free surface flows include the sloshing case which can appear, for instance, in vehicle, ship or aerospace engineering, where the back-and-forth splashing of a liquid fuel in its tank can lead to problems of stability and control in ground vehicles or launch ones.

Another case is the sloshing in a liquid storage tank subjected to seismic action where high impact loads on the tank roof and walls which can damage the liquid storage tank. Early simulations of the liquid sloshing problem in liquid carrier or storage tanks have mostly been performed with waves of small steepness. The sloshing height is assumed to be too small so the nonlinear boundary conditions may be neglected. The most commonly applied classic idealization for estimating liquid response in excited rigid tanks is due to Housner[1] where the hydrodynamic pressures are divided into two components: the impulsive pressure caused by the portion of the liquid accelerating with the tank and the convective pressure caused by the portion of the liquid sloshing in the tank. These hydrodynamics pressures result in added masses[2] which can duplicate forces and moments exerted by a liquid on a vibrating tank.

From a numerical point of view, several techniques have been developed for the solution of free surface flows as initial value problems. These techniques are roughly classified by Shy[3] *et al.* as Eulerian, Lagrangian or mixed Eulerian-Lagrangian. In Eulerian-like (volume-tracking) approaches, see Fig. 1, the mesh remains stationary or moves in a predetermined manner, the free surface is not explicitly tracked, so it is reconstructed from other field properties such as the fluid fractions, and the fluid moves in/out of the computational flow domain. It can handle large displacements without loss of accuracy, but it is rather difficult to impose the free boundary conditions, since a a sharp definition is lacking, e.g. see Nickell[4] *et al.*, Siliman/Scriven[5], Ruschak[6] and Kawahara/Miwa[7]. In Lagrangian-like (surface-tracking) approaches, the mesh is configured to conform the shape of the free surface and, thus, it adapts continually to it. The free surface is a discontinuity and we explicitly track its evolution as an $(n-1)$ dimensional entity in an n-dimensional space. No modeling is necessary to define the free surface or its effects on the flow field. The grid points move with the local fluid particles, so the free surface is sharply defined. However, mesh refinement or remeshing usually is necessary for large deformations, e.g. see Bach/Hassager[8] and Ramaswamy/Kawahara[9]. In mixed Eulerian Lagrangian-like approaches, the advantages of both methods are taken into account, e.g. see Chiapada[10] *et al.*. Other mixed approaches are also proposed. For example, the "emplicit" method uses an explicit-implicit time integration oriented to seakeeping ship motions, e.g. see Huang/Sclavounos[11], while the "material point" method uses unconnected Lagrangian points and a background Eulerian mesh for solving fluid-membrane interaction, e.g. see York[12] *et al.*. Aliabadi[13] *et al.* perform a numerical simulation of sloshing in tanker trucks during turning with a stabilized finite element formulation which is implemented in parallel using the message-passing interface libraries.

In a previous work[14], a Lagrangian-type panel method in the time domain was proposed for inviscid potential flows with a moving free surface where, after a spatial semi-discretization with a low-order scheme, the instantaneous velocity-potential and normal displacement on the moving free surface were obtained by means of a time-marching scheme. Later[15] a surface reallocation strategy for the instantaneous wetted hull surface caused by changes in the position of the intersection curve between the free surface and hull surface was shown.

In this work, a mesh-movement technique is addressed for transient flow domains with a free surface of a viscous and incompressible fluid in the context of a finite element approach and with a parallel computation.

2 GOVERNING EQUATIONS

Let us consider the flow of a viscous and incompressible Newtonian fluid. The governing flow equations are the Navier-Stokes (NS) ones,

$$\begin{aligned} \rho\left(\partial_t \mathbf{v} + \mathbf{v}\cdot\nabla\mathbf{v} - \mathbf{f}\right) - \nabla\cdot\boldsymbol{\sigma} = 0 \; ; \\ \nabla\cdot\mathbf{v} = 0 \; ; \end{aligned} \tag{1}$$

on the flow domain $\Omega_t = \Omega(t)$ at time t, for all $t \in [0, T]$, where $\mathbf{v}$ is the fluid velocity, $\mathbf{f}$ is the body force, ρ is the fluid density and T is a final time. The fluid stress tensor $\boldsymbol{\sigma}$ is decomposed into its isotropic $-p\mathbf{I}$ and deviatoric $\mathbf{T}$ parts

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{T} \; ; \tag{2}$$

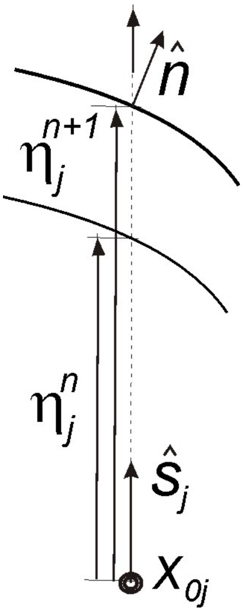

Figure 2: Notation for the spines-like employed in the mesh-movement.

where p is the pressure and $\mathbf{I}$ is the identity tensor. As only Newtonian fluids with constant physical properties are considered, its deviatoric part $\mathbf{T}$ is related linearly to the strain rate tensor with

$$\mathbf{T} = 2\mu\boldsymbol{\epsilon} \quad ; \quad \boldsymbol{\epsilon} = \frac{1}{2}\left[\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right] \ ; \tag{3}$$

where μ and $\nu = \mu/\rho$ are the dynamic and kinematic viscosity of the fluid and $(...)^T$ denotes the transpose. The boundary conditions are

$$\begin{aligned} \mathbf{v} &= 0 \quad &&\text{at } \Gamma_{wall}; \\ p &= P_{\text{atm}} &&\text{at } \Gamma_{FS}; \\ \boldsymbol{\tau}\cdot\mathbf{n} &= 0 &&\text{at } \Gamma_{FS}; \end{aligned} \tag{4}$$

where Γ_{wall} is the boundary on the solid walls while Γ_{FS} is the free surface. Note that as no restriction is imposed on velocity at the free surface, then the normal velocity there can be non-null. This normal velocity is responsible of the free surface movement. The boundary conditions at the free surface are similar to those normally imposed at an outlet boundary.

Discretization of this system of Partial Differential Equations (PDE) by a finite element method in space leads to an Ordinary Differential Equation (ODE) in time which is discretized by a finite difference method and, at each time step, we have a non-linear system of equations of the form

$$\mathbf{F}\left(\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t}, \mathbf{p}^{n+1}\right) = \mathbf{0} \ ; \tag{5}$$

so that, having the state of the fluid at time t^n and a mesh for the domain $\Omega(t^n)$ we can solve for the velocity and pressure unknowns at time t^{n+1}.

As the velocity may be non-null at the free surface, in a Lagrangian approach we should move the nodes there with this velocity

$$\frac{\mathbf{x}_j^{n+1} - \mathbf{x}_j^n}{\Delta t} \approx \mathbf{v}_j^{n+1} \ ; \tag{6}$$

however, displacements of the nodes tangential to the free surface are irrelevant so that we need only to verify the normal component of this equation

$$\left(\frac{\mathbf{x}_j^{n+1} - \mathbf{x}_j^n}{\Delta t} - \mathbf{v}_j^{n+1} \right) \cdot \ \mathbf{n}_j^n = 0 \ . \tag{7}$$

In addition, if we constrain the movement of the free surface nodes to be along a fixed direction, say $\hat{\mathbf{s}}_j$ then

$$\mathbf{x}_j(t) = \mathbf{x}_{0,j} + \eta_j(t) \ \hat{\mathbf{s}}_j \ ; \tag{8}$$

where η_j is a scalar coordinate along some *"spine"* whose direction is $\hat{\mathbf{s}}_j$ and starting point $\mathbf{x}_{0,j}$, see Fig. 2. Then Eq. (5) gives an equation for the increment in η coordinate

$$\Delta\eta_j^{n+1} = \eta_j^{n+1} - \eta_j^n = \Delta t \frac{\mathbf{v}_j^{n+1} \cdot \hat{\mathbf{n}}_j^n}{\hat{\mathbf{s}}_j \cdot \hat{\mathbf{n}}_j^n} \ . \tag{9}$$

Note that the spines $(\mathbf{x}_{0,j}, \hat{\mathbf{s}}_j)$ do not change with time. The only requirement is that the spine direction and the normal must be non-orthogonal at each node at each time step, which means that the fixed direction could be defined almost arbitrary, but it is convenient to chose them as parallel as possible to the expected surface normal. For example, the spines for a spillway are usually drawn perpendicular to the main profile of the structure. These spines are only used for the movement of the free surface nodes, that is, the interior ones are relocated by solving the pseudo elastic problem in an independent fashion from spines directions. On the other hand, the normal to the free surface at node $\mathbf{x}_j$ is computed at each time step using

$$\begin{aligned} \hat{\mathbf{n}}_j &\propto \int_{FS} N_j(\mathbf{x}) \, \hat{\mathbf{n}}(\mathbf{x}) \, \mathrm{d}\mathbf{\Sigma}_{FS} \ ; \\ \|\hat{\mathbf{n}}_j\| &= 1 \ ; \end{aligned} \tag{10}$$

where $N_j(\mathbf{x})$ is the finite element interpolation for the j-node and $\hat{\mathbf{n}}(\mathbf{x})$ is the normal to the free surface at point $\mathbf{x}$. The integration is carried out over the whole free surface, but due to the local support of the finite element interpolation function it involves only those elements that are connected to the j-node. For linear tetrahedral elements in the fluid, this amounts to the weighted average of the normals of the triangular panels around the j-node.

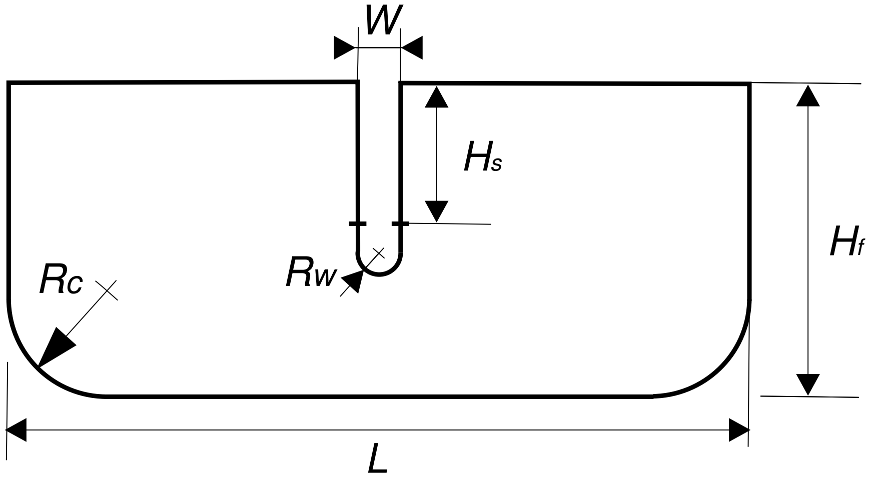

Figure 3: Main geometrical dimensions of a truck-like container with a separating wall.

3 MESH MOVING STRATEGY

Once the displacements of the free surface nodes at time step $n+1$ are known, we are free to relocate the positions of the internal nodes in such a way so as to reduce the distortion of the mesh. Several possibilities have been proposed, we use here a simple strategy based on solving an artificial elastic problem with imposed displacements at the free surface, and slip or non-slip conditions at solid walls and other boundaries, e.g. see Güler[16] *et al.*. For instance, consider a typical case of a truck container as in Fig. 3 with a separating wall. Under longitudinal accelerations of the truck, the fluid tends to go from one half to the other causing large displacements of the fluid surface with extreme positions as shown in Fig. 4. Under these circumstances, best boundary conditions for the pseudo elastic problem may be to let the nodes to move freely at the solid walls $ABCD$, GH and IE (slip boundary condition). However, these can cause large distortions of the elements near the tip F of the separator, so we switch to a non-slip one in a small region around the tip, let us say in the portion HFI. So that the pseudo elastic problem may be posed as

$$\begin{aligned} \sigma_{ij,j} &= 0 \ ; \\ \sigma_{ij} &= 2\tilde{\mu}\epsilon_{ij} + \tilde{\lambda}\delta_{ij}\epsilon_{kk} \ ; \\ \epsilon_{ij} &= \frac{1}{2}(u_{i,j} + u_{j,i}) \ ; \end{aligned} \tag{11}$$

where $\mathbf{u}$ is the mesh node displacements

$$\mathbf{u}_j = \mathbf{x}_j^{n+1} - \mathbf{x}_j^0 \ ; \tag{12}$$

where $\tilde{\mu}$ and $\tilde{\lambda}$ are the Lamé elastic constants for the fictitious elastic material and δ_{ij} is the Kronecker tensor. Of course, the pseudo elastic problem is invariant under a multiplicative

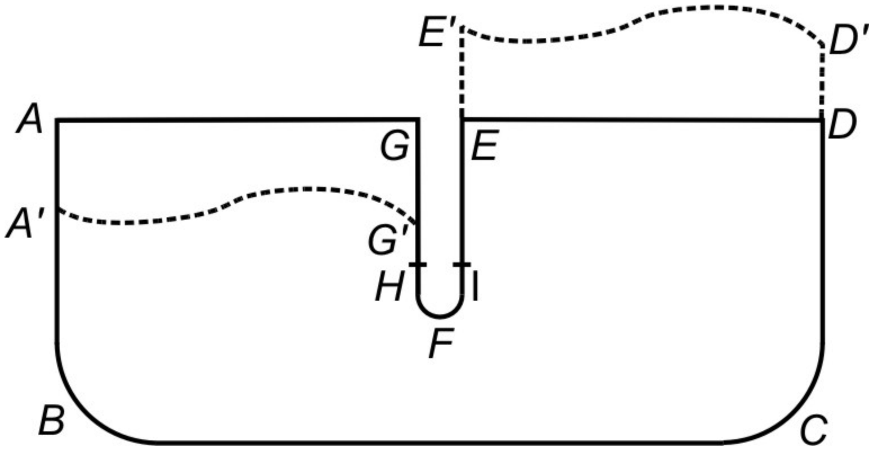

Figure 4: Boundary conditions for the pseudo elastic problem for a mesh movement: nodes can move freely at the solid walls $ABCD$, GH and IE (slip boundary condition) and non-slip one in portion HFI to prevent large distortions of elements near the tip F of the separator.

constant in the elastic coefficients since only Dirichlet conditions are used. The only relevant parameter is the ratio between them, or which is equivalent, the Poisson ratio $\tilde{\nu}$. The Lamé constants $\tilde{\mu}$ and $\tilde{\lambda}$ can be expressed in terms of more familiar modulus of elasticity $\tilde{E}$ with

$$\tilde{\mu} = \frac{\tilde{E}}{2(1+\tilde{\nu})} \qquad ; \qquad \tilde{\lambda} = \frac{\tilde{\nu}\tilde{E}}{(1+\tilde{\nu})(1-2\tilde{\nu})} \; . \tag{13}$$

It is not clear which values are more appropriated for $\tilde{\nu}$ but for $\tilde{\nu} \to 1/2$ the material is nearly incompressible and the pseudo elastic problem will become ill conditioned. In the example we used a value of $\tilde{\nu} = 0.3$. The boundary conditions are

$$\begin{aligned} \mathbf{u} &= \mathbf{x}^{n+1} - \mathbf{x}^0 && \text{at the free surface } AG + ED; \\ \mathbf{u} &= 0 && \text{at a non-slip boundary like } HFI; \\ \mathbf{u} \cdot \hat{\mathbf{n}} &= 0 && \text{at a slip boundary like } ABCD + GH + IE. \end{aligned} \tag{14}$$

The pseudo elastic problem is solved in the reference mesh Ω_0, where the choice between slip and non-slip boundary condition at solid walls is problem dependent and is specified by the user. Once this problem is solved, the position of the internal nodes x^{n+1} are updated with Eq. (12).

Several alternatives for the mesh relocation problem could be devised. Non-linear elastic material behavior could be used in order to reduce distortion but, the linear version shown here has the advantages that the computing time per time step and memory requirements is much lower than that one needed for the fluid, and it was able to solve problems with relatively large distortions as shown in the example below.

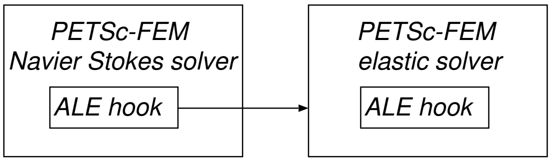

Figure 5: PETSc-FEM hooks that exchange information and data for the synchronization of the global execution of the fluid and pseudo elastic solvers.

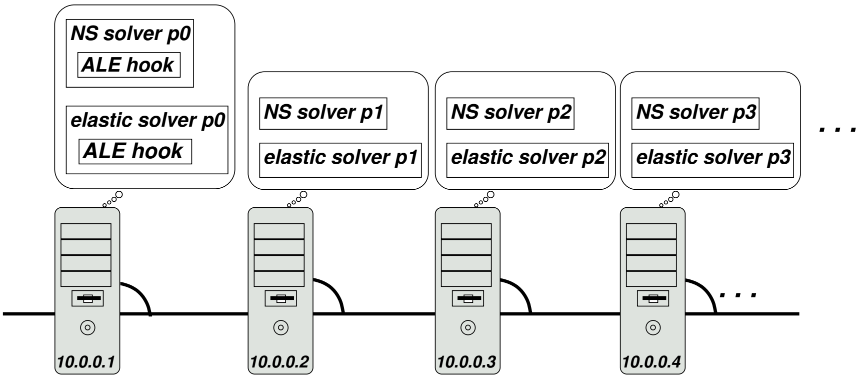

Figure 6: The master processes of both PETSc-FEM (fluid and mesh-movement) are executed at the same computing node.

4 FREE SURFACE ELEVATION SMOOTHING

The whole algorithm as described so far is unstable for gravity waves of high frequency and must be stabilized, mainly due to the fully explicit character of the free surface update given by Eq. (9). This has been also reported by other authors[16]. In this work we applied a smoothing operator to the free surface elevation so that Eq. (9) is replaced by

$$\begin{aligned} \Delta\tilde{\eta}_j^{n+1} &= \frac{\mathbf{v}_j^{n+1} \cdot \hat{\mathbf{n}}_j^n}{\hat{\mathbf{s}}_j \cdot \hat{\mathbf{n}}_j^n} \ ; \\ \Delta\eta_j^{n+1} &= \mathcal{S}(\Delta\tilde{\eta}_j^{n+1}) \ ; \end{aligned} \tag{15}$$

where $\mathcal{S}$ is a smoothing operator based on solving the heat equation with a diffusivity α adjusted so as to have a characteristic spreading length γh, where h is a characteristic global mesh size, and γ is a user chosen parameter. In the example below we choose $\gamma = 2$.

5 MOVING CONTACT LINE

As described so far, the nodes at the contact line (i.e. the intersection of the free surface with a wall boundary, also called waterline) have null velocity and so they do not move. This would lead to large elevation gradients near the wall. The non-slip condition may

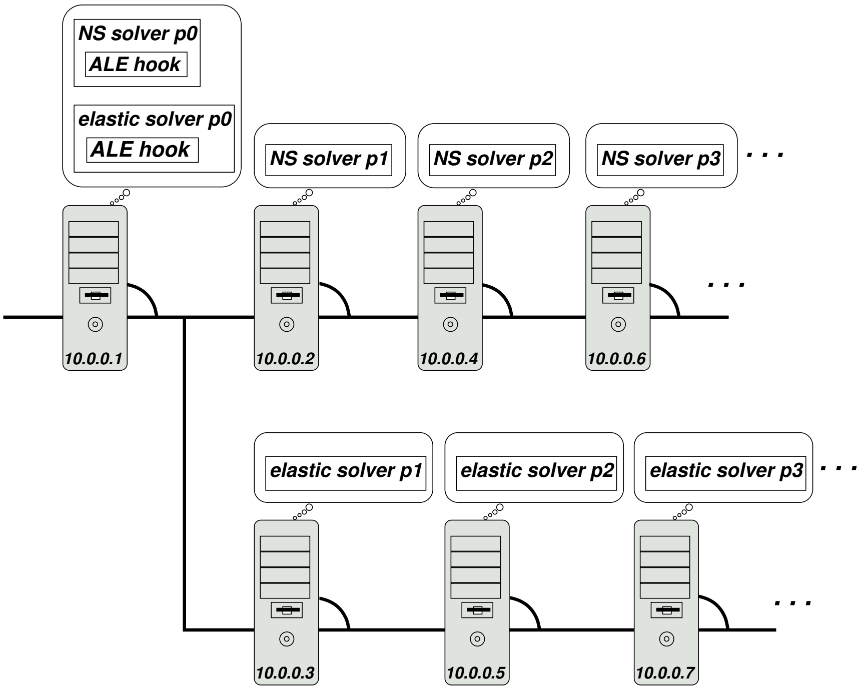

Figure 7: The PETSc FEM parallel runs (fluid and mesh-movement) are running in different node sets but their master processes (MPI rank 0) must be the same.

be relaxed at the contact line and replaced by the Navier slip condition

$$(\mathbf{I}-\mathbf{nn})\cdot(\mathbf{n}\cdot\boldsymbol{\sigma}) = -\frac{1}{\beta}(\mathbf{I}-\mathbf{nn})\cdot(\mathbf{v}-\mathbf{v}_{\mathrm{wall}}) \ ; \tag{16}$$

where $\mathbf{v}$ is the fluid velocity at the contact line, $\mathbf{v}_{\mathrm{wall}}$ is the wall velocity, $\mathbf{I}-\mathbf{nn}$ is the projector onto the tangent plane and β is an empirical slip coefficient. For $\beta = 0$ we recover the non-slip condition, whereas for $\beta \to \infty$ we recover the perfect slip condition. In the example below we used the perfect slip condition on the contact line and the nodes on a thin strip near the contact line.

6 PARALLEL AND MULTI-PHYSICS IMPLEMENTATION DETAILS

Besides the relevance of this problem from the physical and engineering point of view, this problem has interest as a paradigm of multi-physics programming. Even if it could be perfectly possible to implement this problem as a module, it is interesting to see how it can be implemented reusing preexistent fluid and elastic modules, not specifically oriented to the free surface case. We implemented the proposed algorithm in the PETSc-FEM[17,18] code, which is a parallel multi-physics finite element program based on the Message Passing Interface MPI[19] and the Portable Extensible Toolkit for Scientific Computations PETSc[20].

Both pseudo-elastic and fluid problems are run in independent PETSc-FEM instances. Both instances are run in parallel so that, in general, we have at each computing node a

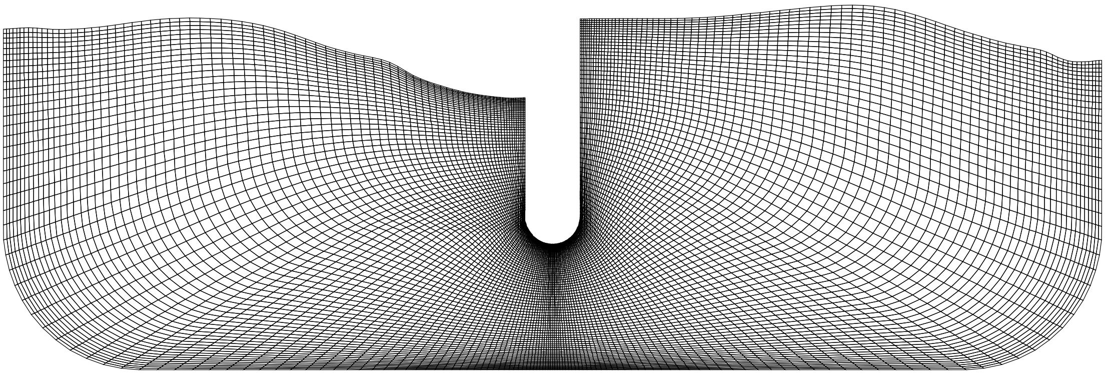

Figure 8: Updated mesh with a pseudo elastic strategy.

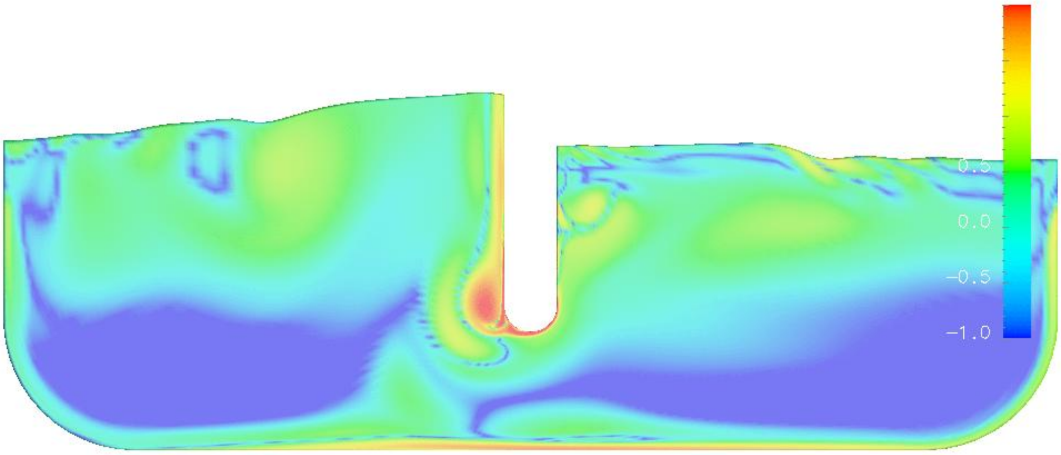

Figure 9: Vorticity at time step $n_t = 422$. A forming vortex is clearly formed on the left wall of the separator near the tip.

PETSC-FEM process for the flow problem and another for the pseudo elastic one. The key point in the implementation is the data exchange and synchronization between both parallel processes. This could be done by modifying the PETSc-FEM sources or well by writing a small script or C++ external code that communicates with both PETSc-FEM processes. But PETSc-FEM has a feature called *"hooks"* that are C++ modules, or may be shell scripts (`bash`, `Perl`, `Python` or other) as well, that are run at certain specific points in the program. This concept has been borrowed from the `GNU Emacs` editor and also from the `Linux`[21] kernel. The C++ hooks are compiled and dynamically loaded at runtime, so that it is not necessity of linking them against PETSc-FEM or modifying the sources. Currently, the Navier-Stokes PETSc-FEM module launches hooks at 4 points in the execution thread: before the beginning of the time step loop, at the beginning and end of each time step, and after the time step loop.

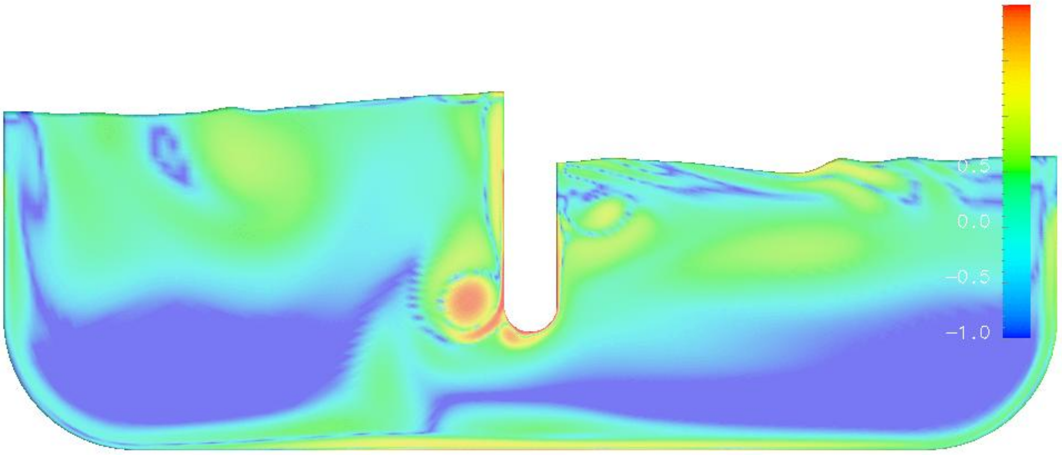

Figure 10: Vorticity at time step $n_t = 434$. The previously formed vortex has been separated from the wall.

For the problem at hand two PETSc-FEM C++ hooks have been written, one that is executed from the NS process and the other from the elasticity, mesh-relocation process. Both hooks exchange information and data for the synchronization of the global execution through a *FIFO* (First Input First Output), also called a *"named pipe"*, with an ad-hoc protocol, see Fig. 5. This is an efficient and portable way of communication between processes and is part of the standard C library (*"libc"*). There is a restriction related with this implementation: inter-process communication via FIFO can be only done between processes in the same host, so that this constrains the master processes of both PETSc-FEM to be executed at the same computing node. That is, the parallel runs (fluid and mesh-movement) can be executed on the same set of nodes, see Fig. 6, or in different ones but their master processes (MPI rank 0) must be the same, see Fig. 7. This restriction could be removed by replacing the FIFO by a socket.

7 NUMERICAL EXAMPLE

A truck-like container tank with an internal buffer subject to an impulsive deceleration is considered, see Fig. 3. The container is moving toward right with velocity 0.5 m/s and suddenly stopped at $t = 0$ s. The tank length and width are $L = 1.20$ m and $L_y = L/2$ m, respectively, and the curvature radius of tank corners is $R_C = 0.15$ m. The length of wall edge separation is $W = 0.15$ m. The starting height of fluid in tank is chosen to be $H_f = 0.36$ m. The width of the strip with slip boundary condition near the contact line (see section §5) was $H_s = 0.2\ H_f = 0.072$ m. The number of elements in fluid movement direction is 80, the same as in the wall-to-wall directions and in the crossing one. The time step is $\Delta t = 0.02$ s, the gravity acceleration $g = 9.81 \text{m/s}^2$ and the kinematic viscosity is

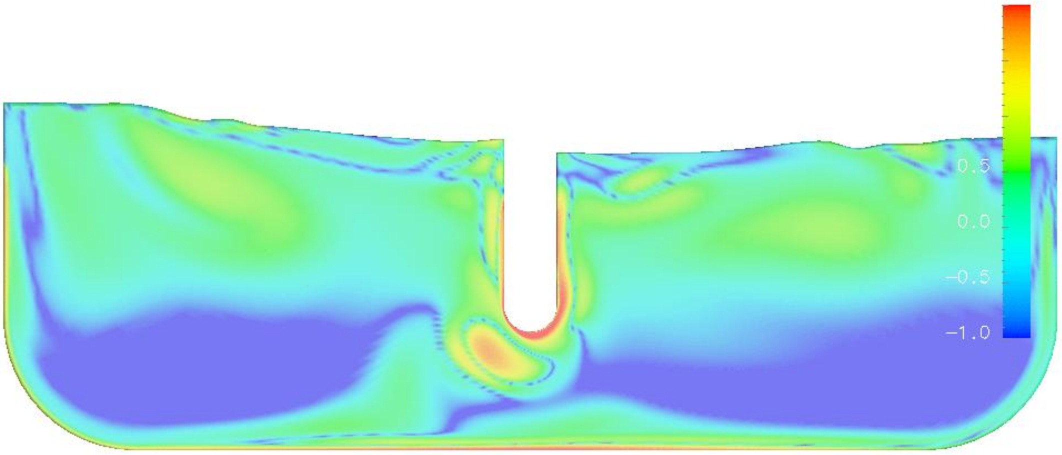

Figure 11: Vorticity at time step $n_t = 448$. Once the vortices are shed they are transported by the fluid.

$\nu = 3\ 10^{-5}\text{m}^2/\text{s}$.

The internal buffer is a wall edge separation that is placed to break the fundamental longitudinal sloshing mode. Nevertheless with this geometrical container configuration and under longitudinal accelerations or decelerations, there is a transient back-and-forth splashing of the liquid, as an hydraulic pendulum. The observed period of the main mode is $T_h \approx 1.7$ s. As the fluid passes from the right half to the left one there is a strong viscous friction near the tip of the separator. This causes emission of vortices, which are the main energy dissipation mechanism. The Figs. 9 and 10 correspond to time instants near the point of maximum height in the left half. A forming vortex is clearly seen in Fig. 9 on the left wall of the separator near the tip. In Fig. 10 it has been already separated from the wall. Once the vortices are shed, they are transported by the fluid and in Figs. 11 and 12 we see the vortex passing to the right half and a new vortex forming on the right half.

8 CONCLUSIONS

In this work we have shown a mesh moving technique for transient free surface flows of a viscous and incompressible fluid in the context of a stabilized finite element approach and parallel computation. The combined fluid and mesh moving problem has been formulated within the picture of the multi-physics programming paradigm, and was implemented reusing preexistent fluid and elastic modules which are not specifically oriented to the free surface case. Future modeling of sloshing applications would be focused on the problem of how to couple the dynamics of the fluid with the container dynamics, for instance, simulate sloshing in containers in the case of tanker trucks with accelerations during turning.

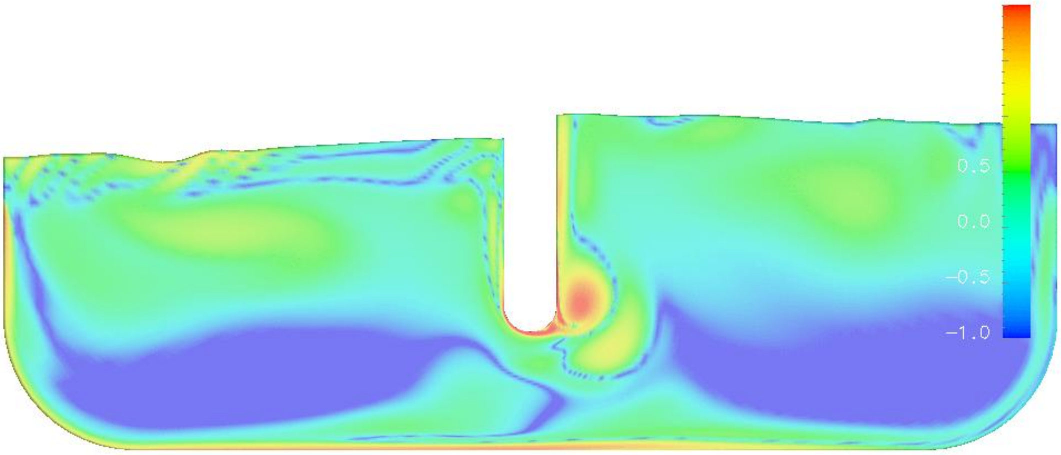

Figure 12: Vorticity at time step $n_t = 462$. A new vortex forming on the right half.

Acknowledgment

This work was partially performed with the *Free Software Foundation/GNU-Project* resources as `GNU/Linux OS` and `GNU/Octave`, as well another Open Source resources as `PETSc`, `MPICH` and `OpenDX`, and supported through grants CONICET-PIP-02552/2000, ANPCyT-FONCyT-PME-209 **Cluster**, ANPCyT-FONCyT-PID-99-74 **Flags**, ANPCyT-PICT-6973-BID-1201/OC-AR **Proa** and CAI+D-UNL-2000-43.

REFERENCES

[1] Housner G.W. Dynamic pressures on accelerated fluid containers. *Bulletin of Seismic Society of America*, **47**, 15–35 (1957).

[2] Storti M. and D'Elía J. Added mass coefficients of an oscillating hemi-sphere at very low and very high frequencies. *ASME Journal of Fluids Engineering*, (2003). to be published.

[3] Shy W., Udaykumar H.S., M.M., and Smith R.W. *Computational Fluid Dynamics with Moving Boundaries.* Taylor & Francis, (1996).

[4] Nickell R.E., Tanner R.I., and Caswell B. The solution of viscous incompressible jet and free-surface flows using finite element method. *J. Fluid Mech.*, **65**, 189 (1974).

[5] Siliman W.J. and Scriven L.E. Separating flow near a static contact line: slip at the wall and shape of a free surface. *J. Comp. Physics*, **34**, 287 (1980).

[6] Ruschak K.J. A method of incorporating free boundaries with surface tension in finite element fluid flow simulation. *Int. J. for Num. Meth. in Engng.*, **15**, 639 (1980).

[7] Kawahara M. and Miwa T. Finite element analysis of wave motion. *Int. J. for Num. Meth. in Engng.*, **20**, 1193 (1984).

[8] Bach P. and Hassager O. An algorithm for the use of lagrangian specification in

newtonian fluid mechanics applications to free surface flows. *J. Fluid Mech.*, **152**, 173 (1985).

[9] Ramaswamy B. and Kawahara M. Lagrangian finite element analysis applied to viscous free surface fluid flow. *Int. J. Num. Meth. Fluids*, **7**, 953 (1987).

[10] Chiapada S., Jue T.C., S.W. Joo, Wheeler M.F., and Ramaswamy R. Numerical simulation of free-boundary problems. *Computational Fluid Dynamics*, **7**, 91 (1996).

[11] Huang Y. and Sclavounos P.D. Nonlinear ship motions. *J. of Ship Research*, **42**(2), 120–130 (1998).

[12] York A.R., Sulsky D., and Schreyer. Fluid-membrane interaction based on the material point method. *Int. J. for Num. Meth. in Engng.*, **48**, 901–924 (2000).

[13] Aliabadi S., Johnson A., and Abedi J. Comparison of finite element and pendulum models for simulation of sloshing. *Computers and Fluids*, **32**(4), 535–545 (2003).

[14] D'Elía J., Storti M., Oñate E., and Idelsohn S. A nonlinear panel method in the time domain for seakeeping flow problems. *International Journal of Computational Fluid Dynamics*, **16**(4), 263–275 (2002).

[15] D'Elía J., Storti M., and Idelsohn S. A surface remeshing for floating-like bodies with a moving free surface. In *ENIEF 2001*, Córdoba, Argentina, (October 30th - November 2nd 2001). XII Congress on Numerical Methods and their Applications.

[16] Güler I., Behr M., and Tezduyar T. Parallel finite element computation of free-surface flows. *Computational Mechanics*, **23**(2), 117–123 (1999).

[17] Sonzogni V.E., Yommi A., Nigro N., and Storti M. Cfd finite element parallel computations on a beowulf cluster. In *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2000*, (11-14 September 2000).

[18] PETSc-FEM: A general purpose, parallel, multi-physics FEM program. GNU General Public License (GPL), http://minerva.arcride.edu.ar/petscfem.

[19] Message passing interface (MPI). http://www.mpi-forum.org/docs/docs.html.

[20] Balay S., Gropp W., McInnes L.C., and Smith B. Petsc 2.0 users manual. Technical Report UC-405, Argonne National Laboratory. Mathematics and Computer Science, (1997).

[21] The Linux Documentation Project, http://wwww.gnu.org.