

FREE-SURFACE FLOWS IN A MULTI-PHYSICS PROGRAMMING PARADIGM

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. *Unsteady free surface flows of an incompressible and viscous fluid are numerically solved by a finite element computation. In a previous communication (e.g. see Battaglia¹ et al.), a mesh-movement technique was addressed for flow domains with a transient free surface of a viscous and incompressible fluid. The combined fluid and mesh moving problem was solved within the picture of a multi-physics programming paradigm, and was implemented reusing preexistent fluid and linear pseudo-elastic modules which were not specifically oriented to the free surface case. The “dialog” (data exchange and synchronization) between the fluid and pseudo-elastic solvers was performed by means of “hooks”. These were C++ modules (or shell scripts like bash, Perl or Python) that run at certain specific points in the program. Nevertheless, when the free surface performs non-small displacements, there can be a numerical breakdown at some mesh update due to larger distortions in some elements close to the solid boundaries. The larger distortions are related to the non-slip boundary conditions imposed in some portion of the boundary. The objective of this work is to employ the non-linear pseudo-elastic formulation for the mesh-movement proposed by López² et al. in order to reduce mesh distortions and thus simulate larger free surface displacements.*

1 INTRODUCTION

Problems involving unsteady free surfaces such as sloshing into fluid containers submitted to accelerations or unstable initial conditions, as in liquid transport carriers are common in physics and engineering.

As performed in previous work (see Battaglia¹ *et al.*), a finite element code for solving transient free surface flows of viscous and incompressible fluids by a time-marching procedure was used. The code involves two different instances: (i) a Navier-Stokes (NS) equations solver which determines the fluid state, and (ii) a mesh-movement process, which provides an updated mesh according to the new free surface position by solving a pseudo-solid mechanical one. A surface tracking method, or Lagrangian-like, was used to determine an updated position of the interface for the next time step, where a linear elastic solver calculated the nodal positions for the updated mesh. This implementation, based on the proposal made by Güller³ *et al.*, can be employed when the small-deformations paradigm is still valid, but tends to fail when the deformation amplitudes grow or complicated geometries have to be solved. Thus, a better mesh update is necessary for non-small deformations in such cases.

Among other strategies, Behr⁴ *et al.* mention three approaches for a mesh-update:

- *Algebraic*: the displacement of interior nodes of the domain is function of the boundaries nodes. It can be useful for simple geometries and structured meshes;
- *Elastic*: the free surface displacements are used as data for a pseudo-elastic problem which gives the new nodes positions through a solid mechanic solver. This proposal offers several alternatives for controlling mesh quality;
- *Remeshing*: a new mesh is generated for the modified domain. However, it is usually expensive and implies an interpolation of data between old and new mesh.

In this work, the elastic approach for a mesh-update is selected in two cases, but faced in different ways, i.e., applying a linear elastic solver and a non-linear one. Finally, a procedure based on solving an *optimization problem*, out of this classification, is applied. It was proposed by López² *et al.*, based on certain mesh quality indicators, which became an interesting alternative for the mesh-update. The algorithm is implemented in the PETSc-FEM^{5,6} code, a finite element program for parallel computing, which counts on several modules, as the NS and the elastic ones. The purpose of this work is to evaluate the abilities of different algorithms for mesh-updating when applied to unsteady free-surface problems.

2 FINITE ELEMENT IMPLEMENTATION

The finite element code PETSc-FEM is based on the Message Passing Interface⁷ (MPI) and the Portable Extensible Toolkit for Scientific Computations⁸ (PETSc), and it involves several modules.

The general scheme for the implementation of the whole process is independent from the mesh-update approach selected. It consists in the synchronization of two processes, one that calculates the state in the fluid, the NS solver, and the second one which controls the mesh-update, see Fig. 1. The latter could be an elastic one, linear or non-linear, or the mesh-move application. For certain cases, where instabilities are registered over the free-surface, a smoothing operator is applied over the fluid results before the mesh-update process.

The fluid-problem and mesh-update are run in different instances, so communication between them must be effective. This communication is made through C++ modules or shell scripts called *hooks*, that are executed at certain points of the main process.

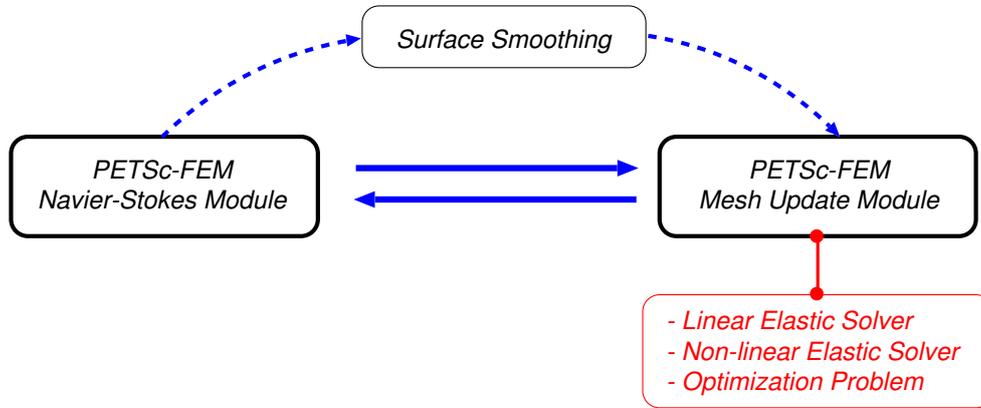


Figure 1: General scheme for solving free surface flows in PETSc-FEM.

The C++ hooks used in this case, one executed from the NS solver and the other from the mesh update one, exchange information and data through a FIFO (First Input First Output). The following sections give details about each module involved in the process described so far.

3 FLOW PROBLEM

As the flow is considered viscous, incompressible and Newtonian, the NS equations are applied over the flow domain $\Omega_t = \Omega(t)$ at time t , $t \in [0, T]$,

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

where \mathbf{v} is the fluid velocity, \mathbf{f} the body force, ρ the fluid density, T a final time considered, and $\boldsymbol{\sigma}$ the fluid stress tensor, composed by an isotropic $-p\mathbf{I}$ and a deviatoric part \mathbf{T} , i.e.

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{T} ; \quad (2)$$

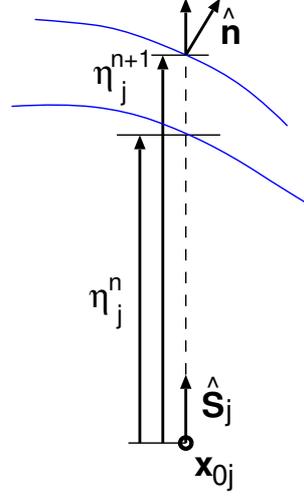


Figure 2: Directions and magnitudes for free surface nodes displacement.

where p is the pressure and \mathbf{I} representing the identity tensor. The deviatoric part in the Newtonian fluid case can be expressed as

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

considering μ and $\nu = \mu/\rho$ as dynamic and kinematic viscosity of the fluid, respectively, and $(\dots)^T$ indicates the transpose. The boundary conditions are expressed by

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

where Γ_{wall} is the boundary on the solid-walls and Γ_{FS} is the free-surface. The third expression allows free surface movement in its normal direction. Then, from time step t^n , nodal velocities in time t^{n+1} are approximated as

$$\mathbf{v}_j^{n+1} \approx \frac{\mathbf{x}_j^{n+1} - \mathbf{x}_j^n}{\Delta t} \quad ; \quad (5)$$

The free-surface node movement is restricted to a fixed direction, \hat{s}_j , in this case assumed vertical, and is calculated as

$$\mathbf{x}_j(t) = \mathbf{x}_{0,j} + \eta_j(t) \hat{s}_j \quad ; \quad (6)$$

where η_j is the scalar quantity along one “spine” of direction \hat{s}_j associated to the nodal initial position $\mathbf{x}_{0,j}$ and constant in time, showed in Fig. 2. Then, from Eq. 5,

$$\Delta\eta_j^{n+1} = \eta_j^{n+1} - \eta_j^n = \Delta t \frac{\mathbf{v}_j^{n+1} \cdot \hat{\mathbf{n}}_j^n}{\hat{s}_j \cdot \hat{\mathbf{n}}_j^n} \quad . \quad (7)$$

Finally, the normal to the free surface at nodes x_j is approximated each time step considering the finite element approximation function $N_j(\mathbf{x})$ corresponding to the node, integrated over neighbor elements, and according to the nature of the free surface, i.e., over linear elements for bidimensional problems, or triangles (or quads) in three dimensions.

At the waterline, which is the intersection of the free surface with a wall, the non-slip boundary condition for the fluid imposes null velocities, and would translate into large gradients at free surface in the proximity of the wall. So, the condition is relaxed by replacing the latter with the Navier slip condition, expressed as

$$(\mathbf{I} - \mathbf{nn}) \cdot (\mathbf{n} \cdot \boldsymbol{\sigma}) = -\frac{1}{\beta}(\mathbf{I} - \mathbf{nn}) \cdot (\mathbf{v} - \mathbf{v}_{\text{wall}}) ; \quad (8)$$

where $\mathbf{I} - \mathbf{nn}$ projects $(\mathbf{n} \cdot \boldsymbol{\sigma})$ onto the tangent plane, \mathbf{v} is the velocity in the fluid and \mathbf{v}_{wall} is the velocity of the wall. The slip parameter β , empirically determined, allows the condition to vary from the perfect slip condition ($\beta \rightarrow \infty$) to non slip condition ($\beta = 0$). For detailed information, see previous work.¹

4 MESH-UPDATE

As mentioned before, the mesh-update has been done in different manners, by applying one of the methods described below. Once the velocity field over the free-surface is known, nodal displacements are calculated, and became input data for the mesh-update problem.

Regardless of the process selected to get the new nodes position, boundary conditions for this stage are common for the alternatives tested. As an example, the boundary conditions considered for the problem below are: (i) the nodes are fixed at the bottom of the container, and (ii) a perfect slip boundary condition at the lateral walls. The nodal displacements over the interface determined after the fluid step are the imposed ones for this problem. Obviously, the selection of one boundary condition over the other must always be in accordance to the problem considered, as perfect slip in sections close to moving boundaries.

All methods applied in this scope keep the initial topology of the mesh, because they update nodal positions but do not modify conectivities.

At the beginning, the selection of one over the other was made by taking into account computational costs. As complexity of the geometry grows or higher deformations are registered, more robust tools are needed, so both criteria must be considered at the time to choose between the alternatives considered in the Sec. 4.1-4.2.

4.1 Pseudo-elastic Problem

The mesh-update consists in calculating the new nodal positions by solving an artificial elastic problem over the domain Ω_0 , where the boundary conditions may be slip or non-slip over the solid walls, but are always of the Dirichlet type. This pseudo-elastic problem

may be formulated as a standard elastic one,

$$\begin{aligned}\sigma_{ij,j} &= 0 ; \\ \sigma_{ij} &= 2\tilde{\mu}\epsilon_{ij} + \tilde{\lambda}\delta_{ij}\epsilon_{kk} ; \\ \epsilon_{ij} &= \frac{1}{2}(u_{i,j} + u_{j,i}) ;\end{aligned}\tag{9}$$

where $\tilde{\mu}$ and $\tilde{\lambda}$ are the artificial Lamé elastic constants for the material, δ_{ij} is the Kronecker tensor and node displacements are

$$\mathbf{u}_j = \mathbf{x}_j^{n+1} - \mathbf{x}_j^0 ;\tag{10}$$

and corresponds to the boundary conditions over the free surface. For solid contours, the boundary condition is imposed as $\mathbf{u} = 0$ for the non-slip part and $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ for the slip ones.

The artificial material properties can also be put in terms of the Poisson ratio $\tilde{\nu}$ and the elasticity modulus \tilde{E} , which are the parameters to be set as input data. Independently of \tilde{E} , and according to boundary condition types, $\tilde{\nu}$ is the relevant one. Usually, $\tilde{\nu} = 0.3$ is used, considering that for $\tilde{\nu} \rightarrow 0.5$, i.e., tending to incompressibility, the problem is ill-conditioned.

The linear solver is able to deal with relatively large deformations, but could get distorted meshes that damage the numerical results and eventually make the update fail. The time employed for solving this update is about 35% or 40% of the time for the fluid problem.

Xu⁹ *et al.* resume some different stiffening methods usually applied to mesh-moving methods by expressing the element stiffness matrix \mathbf{K}_e for the pseudo-elastic problem as

$$\mathbf{K}_e = \int_{\Omega_t^e} \mathbf{B}^T \mathbf{D} \mathbf{B} |\mathbf{J}|^e \tau^e d\Omega_t^e ;\tag{11}$$

with \mathbf{B} as the derivative matrix of shape functions, \mathbf{D} the constitutive matrix, $|\mathbf{J}|^e$ the Jacobian of the element and τ^e a factor that controls stiffening. Following the formulation proposed by Tezduyar and Stein,^{10,11} τ^e is taken as

$$\tau^e = \left(\frac{|\mathbf{J}|^0}{|\mathbf{J}|^e} \right)^s ;\tag{12}$$

where the non-negative number or *stiffness exponent* s is a scaling parameter chosen by the user, $|\mathbf{J}|^e$ is the Jacobian for element e and $|\mathbf{J}|^0$ an arbitrary scaling parameter considered for dimensional consistency. In this way, smaller elements become more stiffened than the larger ones in a degree given by s .

In this work, a null value for the stiffness exponent was set only for the linear elastic case (i.e. $s = 0$), while the remaining ones were analyzed with $s = 1$.

The application of this procedure allows a reduction of distortion of the elements. The main disadvantage of this method in relation to the linear one is that computational costs are higher, about the same as the required for the fluid solver step.

The pseudo-elastic problem is solved by means of the PETSc-FEM elasticity module, but only the internal nodal displacements obtained are useful for this implementation.

4.2 Mesh-move strategy applying an optimization problem

This algorithm has been proposed and implemented by López² et al., and was developed for moving boundary problems with imposed displacements and is applied here to free-surface motion in particular.

The method consists in solving an optimization problem, where the functional to minimize is expressed in a general way as

$$F = F(\{x_j^\alpha\}); \quad (13)$$

x_j^α being the α coordinate of node x_j and $\{x_j^\alpha\}$ the set of mesh coordinates. There are several requirements to obtain a functional appropriate for minimization, considering that the problem is solved by Newton-like methods. The criterion is implemented so far as

$$F = C_v \left(\frac{V}{V_{ref}} - 1 \right)^m + C_q q^n ; \quad (14)$$

where V and V_{ref} are the element and the target volume, respectively, q is a quality indicator for each element, C_v and C_q are weight-coefficients and m, n indicate the norms to apply to size and shape measures. For two-dimensional problems, the area is evaluated instead of the volume.

It must be considered that in order to get coherence between the terms in Eq. 14 m can take even values and then $n < 0$ is taken, so both terms are minimized simultaneously, considering that in this case the distortion index $\frac{1}{q}$ will be reduced.

The quality indicator implemented is

$$q = \frac{CV}{\sum_i l_i^p} ; \quad (15)$$

i.e., it is the quotient between the elemental volume V and the sum of its edge lengths l_i weighted to a power p , which is the space dimension, and scaled by a constant C chosen in such a way that $0 \leq q \leq 1$, where $q = 1$ corresponds to the equilateral element. Then, it is $C = 4\sqrt{3}$ for triangles and $C = 36\sqrt{2}$ for tetrahedral elements.

The algorithm is formulated in order to get the best possible mesh each time step but, of course, under the exposed criteria about quality of elements. Different values given to C_v and C_q allows the user to personalize the mesh-update criteria.

The method is implemented for two-dimensional and three-dimensional problems, but is restricted so far to triangular and tetrahedral elements, respectively.

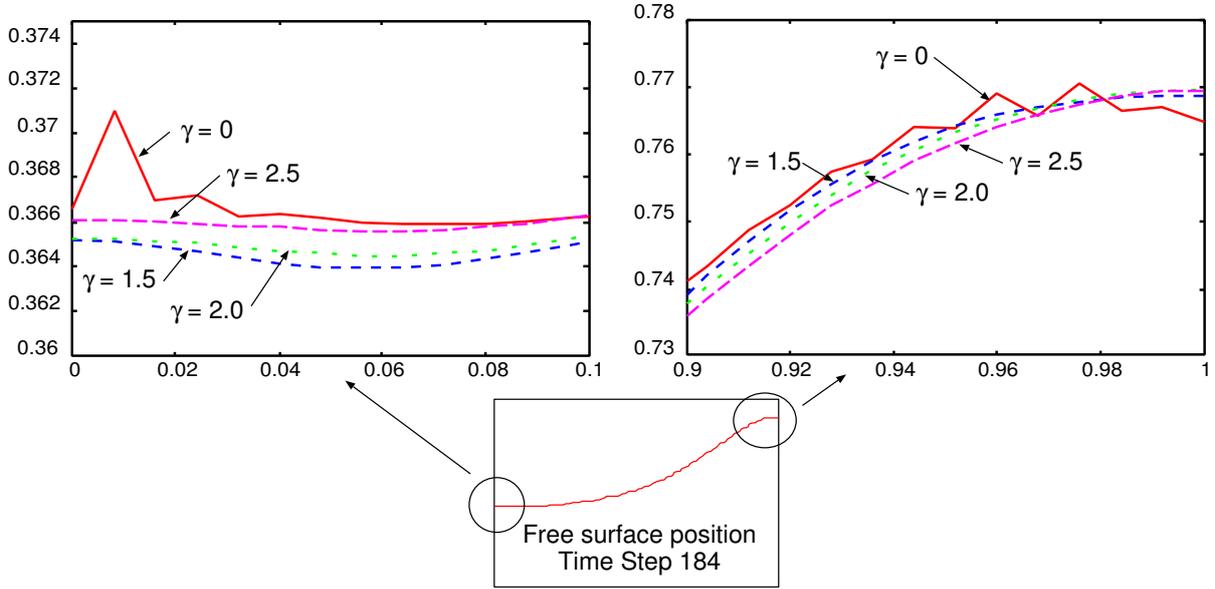


Figure 3: Influence of different γ values over free surface smoothing, scale in meters.

As a general rule, the beginning of the process requires a valid topology, and for the beginning of each updating step must be $q \neq 0$ for $n < 0$.

In the cases developed up to now, for non-small deformations, an initial mesh is obtained by deforming the initial geometry imposing the free surface displacement progressively through this method, even for elastic update cases.

5 FREE SURFACE SMOOTHING

The explicit formulation of the free surface expression given by Eq. 7 proves unstable for gravity waves of high frequency. This circumstance is corrected by means of a smoothing operator \mathcal{S} , in such a way that, calling $\Delta\tilde{\eta}_j^{n+1}$ at $\Delta\eta_j^{n+1}$ from last equation, the increment in η coordinate is

$$\Delta\eta_j^{n+1} = \mathcal{S}(\Delta\tilde{\eta}_j^{n+1}) . \quad (16)$$

This application consist in solving the heat equation with artificial parameters for diffusivity (α) and characteristic length γh , h being a characteristic mesh-size and γ a parameter proposed by the user.

As an illustration of the effects of high frequency over free surface near solid walls, Fig. 3 shows results obtained with different γ values, and not smoothed solution, in the case of an elastic non-linear mesh update.

6 NUMERICAL EXAMPLE

The numerical example chosen is a two dimensional container, whose geometry is shown in Fig. 4, and its length and width are $L = 1.00$ m and $H_{fl} = 0.50$ m, respectively.

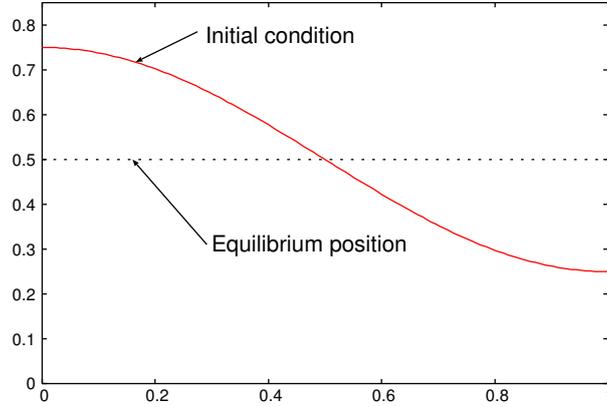


Figure 4: Container dimensions and initial condition for the example (in meters).

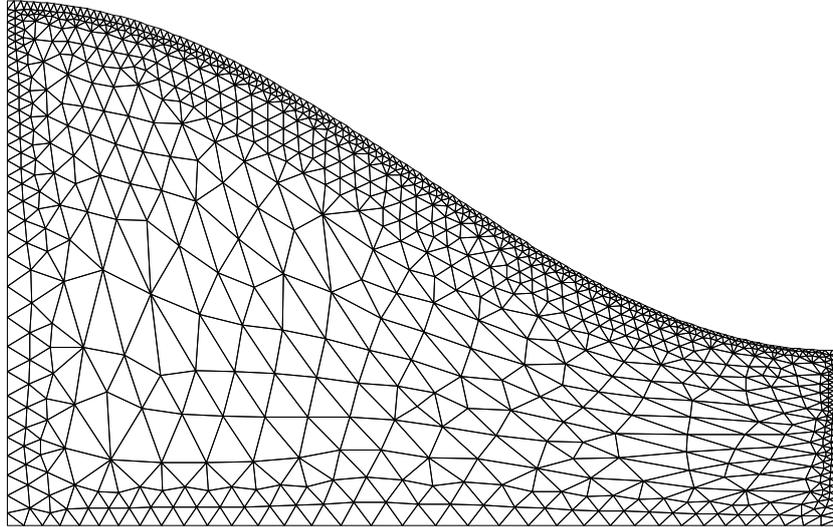


Figure 5: Initial mesh configuration for the example.

The example is similar to one showed by Rabier and Medale,¹² but instead of a small initial free-surface displacement and perfect slip hypothesis over the solid boundaries, the fluid is liberated from a sinusoidal shape of maximum elevation of $\eta_0 = 0.50 H_f = 0.25$ m, relatively large with respect to the channel height, see same figure, and a Navier-slip condition is considered over the walls except on a strip of $H_s = 0.20 H_f = 0.10$ m, where a perfect-slip is assumed.

The employed mesh has 1734 triangular elements and 980 nodes, refined near the free-surface in order to get elements of $h \approx 0.008$ m, while in the interior domain they are proposed as $h = O(0.05$ m), see Fig. 5.

The gravity acceleration and the fluid kinematic viscosity are $g = 1$ m/s² and $\nu = 3 \cdot 10^{-5}$ m²/s, respectively. The time-step chosen is $\Delta t = 0.01$ s, though other analy-

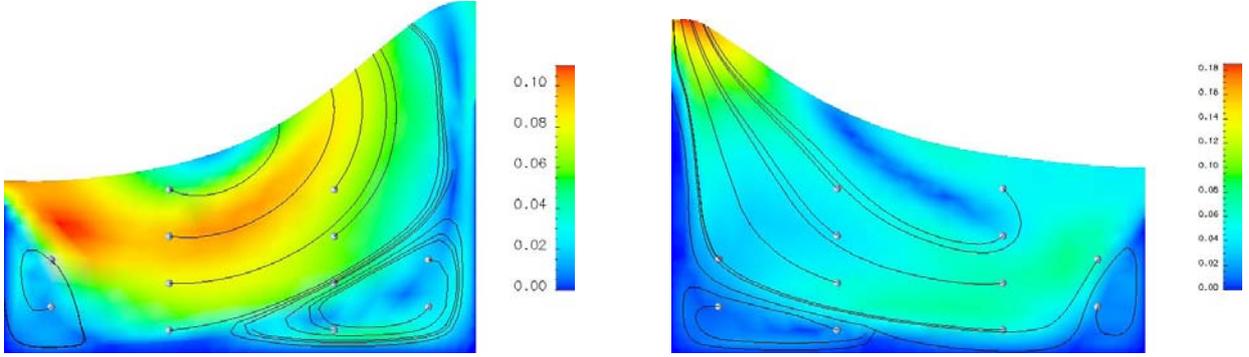


Figure 6: Velocity field and streamlines for the test problem with linear pseudo elastic mesh update. Left: time step 184, $t = 1.84$ s. Right: time step 368, $t = 3.68$ s.

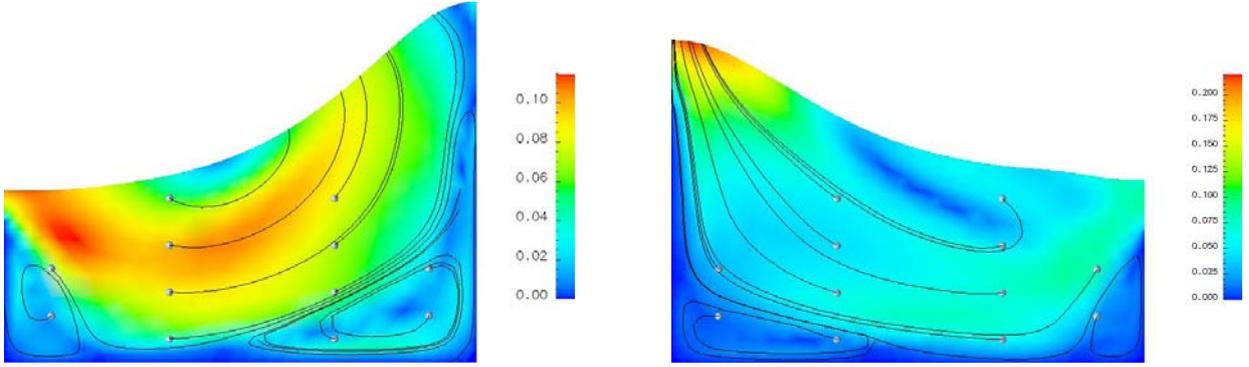


Figure 7: Velocity field and streamlines for the test problem with non-linear pseudo-elastic mesh update. Left: time step 184, $t = 1.84$ s. Right: time step 368, $t = 3.68$ s.

sis for the same topology and initial conditions were done with larger ones. For small deformations, the inviscid natural sloshing frequencies are given by

$$\omega_i^2 = gk_i \tanh(k_i H_f) ; \quad (17)$$

for $i = 0, 1, \dots$, where $k_i = 2\pi/\lambda_i$ is the wave-number and λ_i is the wave-length of the i -mode (this expression is reduced to $\omega_i^2 \approx gk_i$ for the deep-water case). The fundamental sloshing-mode ($i = 0$) has the wave-length $\lambda_0 = 2L$ so $k_0 = \pi/L$ and $\omega_0 \approx 1.6974 \text{ s}^{-1}$. Then, the corresponding natural period is $T_0 = 2\pi/\omega_0 \approx 3.7016$ s, which was verified for this problem.

The free surface smoothing was applied over all models, because of the effects of high frequency waves over the interface. After several essays, best smoothing coefficients were determined in $\gamma = 2.0$ for the linear elastic mesh update case, $\gamma = 2.5$ for the stiffened elastic one and in $\gamma = 1.5$ for the optimization method. For the latter, only the quality

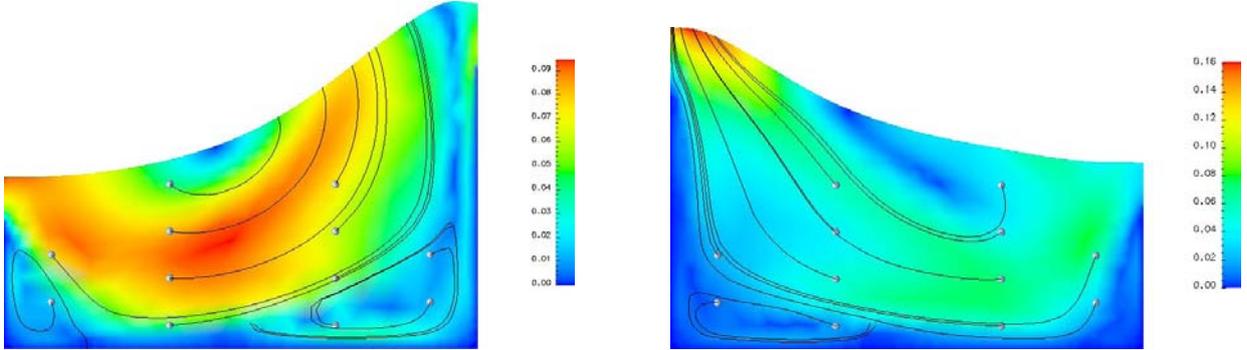


Figure 8: Velocity field and streamlines for the test problem with the mesh updated through the optimization problem. Left: time step 184, $t = 1.84$ s. Right: time step 368, $t = 3.68$ s.

term in Eq. 14 was considered, i.e., $C_v = 0$. Figs. 6 to 7 show some results obtained with the tested approaches.

It is remarked that the conservation of mass was controlled in all cases, with a mass increment of order 0.20% or smaller for $\Delta t = 0.01$ s, and grows to 0.50% when time-step was duplicated.

The three move-mesh strategies described have been also implemented for the three-dimensional case.

7 CONCLUSIONS

For a mesh-update in a finite element context of unsteady free-surface flows that do not break, three approaches were considered: (i) a linear pseudo-elastic one (or classic), through minimizing the elastic potential energy, (ii) a non-linear one, through an ad-hoc stiffness exponent, and (iii) minimization of mesh distortion functional. The first two cases were solved using standard finite element schemes while the third one involves an optimization problem. The considered mesh-update approaches were all appropriate for the numerical example, since the mass conservation was satisfied and the free-surface shape was predicted in a plausible way, but an issue that is not yet addressed is which of the considered approaches is more convenient for better prediction of the unsteady free-surface shape. This should be related to the origin of differences among the results obtained with each approach.

The overall approach allows to considering more complicated geometries whose analytical or semi-analytical solutions for the sloshing eigen-modes cannot be easily found, and thus ascertain what sort of containers reduces sloshing. Future modeling efforts would also be focused on a study of other items such as free surface stabilization and grid independence.

Acknowledgments

The authors thank E. López for his help and participation in fruitful discussions

on the extension of the mesh-move strategy applying an optimization problem. 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-FONCyT-PICT-14573 **Lambda**, and CAI+D-UNL-2000-43.

REFERENCES

- [1] Battaglia L., D'Elía J., Storti M., and Nigro N. Parallel implementations of free surface flows. In *Mecánica Computacional*, vol. XXIII, pages 3119–3132, (November 08-11 2004).
- [2] López E.J., Toth J.A., and Nigro N. Técnicas para definir la cinemática de mallas adaptables a dominios con fronteras móviles. In *Mecánica Computacional*, vol. XXIII, pages 3251–3271, (November 08-11 2004).
- [3] Güler I., Behr M., and Tezduyar T. Parallel finite element computation of free-surface flows. *Computational Mechanics*, **23**(2), 117–123 (1999).
- [4] Behr M. and Abraham F. Free surface flow simulations in the presence of inclined walls. *Computer Methods in Applied Mechanics and Engineering*, **191**(47-48), 5467–5483 (2002).
- [5] PETSc-FEM: A general purpose, parallel, multi-physics FEM program. GNU General Public License (GPL), <http://venus.ceride.gov.ar/petscfem>.
- [6] Sonzogni V.E., Yommi A., Nigro N., and Storti M. Cfd finite element parallel computations on a beowulf cluster. In *ECCOMAS 2000*, (11-14 September 2000).
- [7] Message passing interface (MPI). <http://www.mpi-forum.org/docs/docs.html>.
- [8] Balay S., Gropp W., McInnes L.C., and Smith B. Petsc 2.0 users manual. Technical Report UC-405, Argonne Nat. Lab., (1997).
- [9] Xu Z. and Accorsi M. Finite element mesh update methods for fluid-structure interaction simulation. *Finite Element in Analysis and Design*, **40**, 1259–1269 (2004).
- [10] Tezduyar T., Sathe S., Keedy R., and Stein K. Space-time techniques for finite element computation of flows with moving boundaries and interfaces. In *III Congreso Internacional sobre Métodos Numéricos en Ingeniería y Ciencias Aplicadas*, (2004).
- [11] Stein K., Tezduyar t., T., and Benney R. Automatic mesh update with the solid-extension mesh moving technique. *Comput. Meth. Appl. Mech. Engrg.*, **192**, 2019–2032 (2004).
- [12] Rabier S. and Medale M. Computation of free surface flows with a projection FEM in a moving mesh framework. *Computer methods in applied mechanics*, **192**, 4703–4721 (2003).