

International Journal of Computational Methods, 2004  
© World Scientific Publishing Company

## THE PARTICLE FINITE ELEMENT METHOD. AN OVERVIEW

E. OÑATE<sup>†</sup>, S.R. IDELSOHN<sup>†‡</sup>

<sup>†</sup> *International Center for Numerical  
Methods in Engineering (CIMNE)  
Universidad Politécnica de Catalunya  
Campus Norte UPC, 08034 Barcelona, Spain  
onate@cimne.upc.es  
Web page: <http://www.cimne.upc.es>*

F. DEL PIN<sup>‡</sup>, R. AUBRY<sup>†</sup>

<sup>‡</sup> *CIMEC  
Universidad Nacional del Litoral  
Güemes 3450, 3000 Santa Fe, Argentina  
sergio@ceride.gov.ar*

Received (14 January 2004)

Revised (revised date)

We present a general formulation for analysis of fluid-structure interaction problems using the particle finite element method (PFEM). The key feature of the PFEM is the use of a Lagrangian description to model the motion of nodes (particles) in both the fluid and the structure domains. Nodes are thus viewed as particles which can freely move and even separate from the main analysis domain representing, for instance, the effect of water drops. A mesh connects the nodes defining the discretized domain where the governing equations, expressed in an integral form, are solved as in the standard FEM. The necessary stabilization for dealing with the incompressibility condition in the fluid is introduced via the finite calculus (FIC) method. A fractional step scheme for the transient coupled fluid-structure solution is described. Examples of application of the PFEM method to solve a number of fluid-structure interaction problems involving large motions of the free surface and splashing of waves are presented.

*Keywords:* Particle finite element method; finite element method; fluid-structure interaction; finite calculus.

### 1. Introduction

There is an increasing interest in the development of robust and efficient numerical methods for analysis of engineering problems involving the interaction of fluids and structures accounting for large motions of the fluid free surface and the existence of fully or partially submerged bodies. Examples of this kind are common in ship hydrodynamics, off-shore structures, spillways in dams, free surface channel flows, liquid containers, stirring reactors, mould filling processes, etc.

The movement of solids in fluids is usually analyzed with the finite element method (FEM) [Zienkiewicz and Taylor (2000)] using the so called arbitrary Lagrangian-Eulerian (ALE) formulation [Donea and Huerta (2003)]. In the ALE approach the movement of the fluid particles is decoupled from that of the mesh nodes. Hence the relative velocity between mesh nodes and particles is used as the convective velocity in the momentum equations.

The ALE formulation has been used in conjunction with stabilized finite element method to derive a number of numerical procedures for FSI analysis. For instance the deforming-spatial-domain/stabilized space-time (DSD/SST) [Tezduyar *et al.* (1992a,b)] formulation has been used for computation of fluid-structure interaction and free-surface flow problems. The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) [Tezduyar (2001)] was proposed for computation of problems that involve both fluid-structure interactions and free surfaces. The MITICT can in general be used for classes of problems that involve both interfaces that can be tracked with a moving-mesh method and interfaces that are too complex or unsteady to be tracked and therefore require an interface-capturing technique.

Typical difficulties of fluid-structure interaction (FSI) analysis using the FEM with both the Eulerian and ALE formulation include the treatment of the convective terms and the incompressibility constraint in the fluid equations, the modelling and follow-up of the free surface in the fluid, the transfer of information between the fluid and solid domains via the contact interfaces, the modelling of wave splashing, the possibility to deal with large rigid body motions of the structure within the fluid domain, the efficient updating of the finite element meshes for both the structure and the fluid, etc.

Most of these problems disappear if a *Lagrangian description* is used to formulate the governing equations of both the solid and the fluid domain. In the Lagrangian formulation the motion of the individual particles are followed and, consequently, nodes in a finite element mesh can be viewed as moving “particles”. Hence, the motion of the mesh discretizing the total domain (including both the fluid and solid parts) is followed during the transient solution.

In this paper we present an overview of a particular class of Lagrangian formulation developed by the authors for solving problems involving the interaction between fluids and solids in a unified manner. The method, called the *particle finite element method* (PFEM), treats the mesh nodes in the fluid and solid domains as particles which can freely move and even separate from the main fluid domain representing, for instance, the effect of water drops. A finite element mesh connects the nodes defining the discretized domain where the governing equations are solved in the standard FEM fashion. The PFEM is the natural evolution of recent work of the authors for the solution of FSI problems using Lagrangian finite element and meshless methods [Aubry *et al.* (2004); Idelsohn *et al.* (2003a; 2003b; 2004); Oñate *et al.* (2003; 2004)].

An obvious advantage of the Lagrangian formulation is that the convective terms disappear from the fluid equations. The difficulty is however transferred to

the problem of adequately (and efficiently) moving the mesh nodes. Indeed for large mesh motions remeshing may be a frequent necessity along the time solution. We use an innovative mesh regeneration procedure blending elements of different shapes using an extended Delaunay tessellation [Idelsohn *et al.* (2003a; 2003c)]. Furthermore, this special polyhedral finite element needs special shape functions. In this paper, meshless finite element (MFEM) shape functions has been used [Idelsohn *et al.* (2003a)].

The need to properly treat the incompressible condition in the fluid still remains in the Lagrangian formulation. The use of standard finite element interpolations may lead to the volumetric locking defect unless some precautions are taken. A number of stabilization finite element procedures aiming to alleviate the locking problem in incompressible fluids have been proposed and some of them are listed in [Chorin (1967); Codina (2002); Codina *et al.* (1998); Codina and Blasco (2000); Codina and Zienkiewicz (2002); Cruchaga and Oñate (1997; 1999); Donea and Huerta (2003); Franca and Frey (1992); Hansbo and Szepessy (1990); Hughes *et al.* (1986; 1989; 1994); Oñate (1998); Sheng *et al.* (1996); Tezduyar *et al.* (1992); Zienkiewicz and Taylor (2000); Storti *et al.* (2004)]. A general aim is to use low order elements with equal order interpolations for the velocity and pressure variables. In our work the stabilization via a finite calculus (FIC) procedure has been chosen [Oñate (2000)]. Recent applications of the FIC method for incompressible flow analysis using linear triangles and tetrahedra are reported in [García and Oñate (2003); Oñate (2002); Oñate *et al.* (2000; 2004); Oñate and García (2001); Oñate and Idelsohn (1998)].

The Lagrangian formulation has many advantages for tracking the motion of fluid particles in flows accounting for large displacements of the fluid surface as in the case of breaking waves and splashing of liquids (Figure 1). We note that the information in the PFEM is typically nodal-based, i.e. the element mesh is mainly used to obtain the values of the state variables (i.e. velocities, pressure, etc.) at the nodes. A difficulty arises in the identification of the boundary of the domain from a given collection of nodes. Indeed the “boundary” can include the free surface in the fluid and the individual particles moving outside the fluid domain. In our work the Alpha Shape technique [Edelsbrunner and Mücke (1999)] is used to identify the boundary nodes.

The layout of the paper is the following. In the next section the basic ideas of the PFEM are outlined. Next the basic equation for an incompressible flow using an updated Lagrangian description and the FIC formulation are presented. Then a fractional step scheme for the transient solution via standard finite element procedures is described. Details of the treatment of the coupled FSI problem are given. The procedures for mesh generation and for identification of the free surface nodes are briefly outlined. Finally, the efficiency of the *particle finite element method* (PFEM) is shown in its application to a number of FSI problems involving large flow motions, surface waves, moving bodies. etc.

4 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

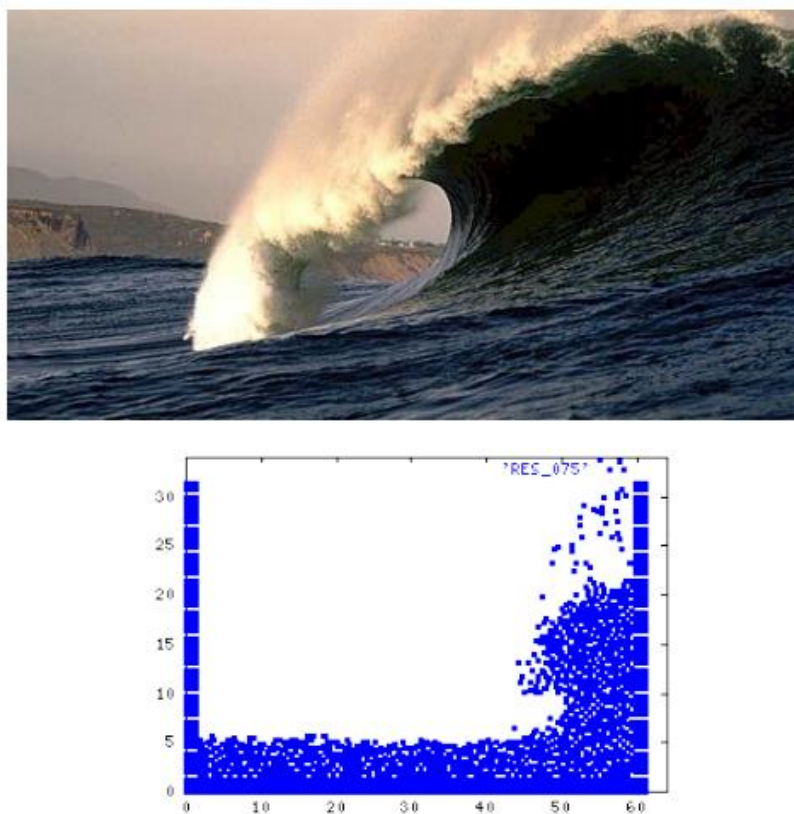


Fig. 1. (a) Large breaking wave. (b) PFEM results for a large wave hitting a vertical wall in 2D.

## 2. Rationale of the Particle Finite Element Method

Let us consider a domain containing both fluid and solid subdomains. The moving fluid particles interact with the solid boundaries thereby inducing the deformation of the solid which in turn affects the flow motion and, therefore, the problem is fully coupled.

FSI problems have been traditionally solved using an arbitrary Eulerian-Lagrangian description (ALE) for the flow equation whereas the structure is modelled with a full Lagrangian formulation. Many examples of applications of this type of approach are found in the literature. A good review can be found in [Donea and Huerta (2003); Zienkiewicz and Taylor (2000)].

In the PFEM approach presented here, both the fluid and the solid domains are modelled using an *updated Lagrangian formulation*. The finite element method (FEM) is used to solve the continuum equations in both domains. Hence a mesh discretizing these domains must be generated in order to solve the governing equations for both the fluid and solid problems in the standard FEM fashion. We note

once more that the nodes discretizing the fluid and solid domains can be viewed as material particles which motion is tracked during the transient solution.

The quality of the numerical solution will obviously depend of the discretization chosen as in the standard FEM. Adaptive mesh refinement techniques can be used to improve the solution in zones where large motions of the fluid or the structure occur.

The Lagrangian formulation allows to track the motion of each single fluid particle (a node). This is useful to model the separation of water particles from the main fluid domain and to follow their subsequent motion as individual particles with an initial velocity and subject to gravity forces.

In summary, a typical solution with the PFEM involves the following steps.

- (1) Discretize the fluid and solid domains with a finite element mesh. The mesh generation process can be based on a standard Delaunay discretization [George (1991)] of the analysis domain using an initial collection of points which then become the mesh nodes. Alternatively, the nodes can be created during the mesh generation process using a front generation method [Irons (1970); Thompson *et al.* (1999)].
- (2) Identify the external boundaries for both the fluid and solid domains. This is an essential step as some boundaries (such as the free surface in fluids) may be severely distorted during the solution process including separation and re-entring of nodes. The Alpha Shape method [Edelsbrunner and Mucke (2003)] is used for the boundary definition (see Section 7).
- (3) Solve the coupled Lagrangian equations of motion for the fluid and the solid domains. Compute the relevant state variables in both domains at each time step: velocities, pressure and viscous stresses in the fluid and displacements, stresses and strains in the solid.
- (4) Move the mesh nodes to a new position in terms of the time increment size. This step is typically a consequence of the solution process of step 3.
- (5) Generate a new mesh if needed. The mesh regeneration process can take place after a prescribed number of time steps or when the actual mesh has suffered severe distortions due to the Lagrangian motion. In our work we use an innovative mesh generation scheme based on the extended Delaunay tessellation (Section 7) [Idelsohn *et al.* (2003a; 2003b; 2004)].
- (6) Go back to step 2 and repeat the solution process for the next time step.

Details of the stabilized Lagrangian FEM for the solution of the fluid equations using a FIC formulation are presented in the next section. The fractional scheme chosen for the transient coupled FSI solution using the FEM and details of the boundary recognition method and the mesh regeneration process are given in subsequent sections. Finally some examples of application of the PFEM are given.

In order to complete this introduction, Figure 2 shows a typical example of a PFEM solution in 2D. The pictures correspond to the analysis of the problem of breakage of a water column described in Section 10.1. Figure 2a shows the initial grid

6 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

of four node rectangles discretizing the fluid domain and the solid walls. Boundary nodes identified with the Alpha-Shape method have been marked with a circle.

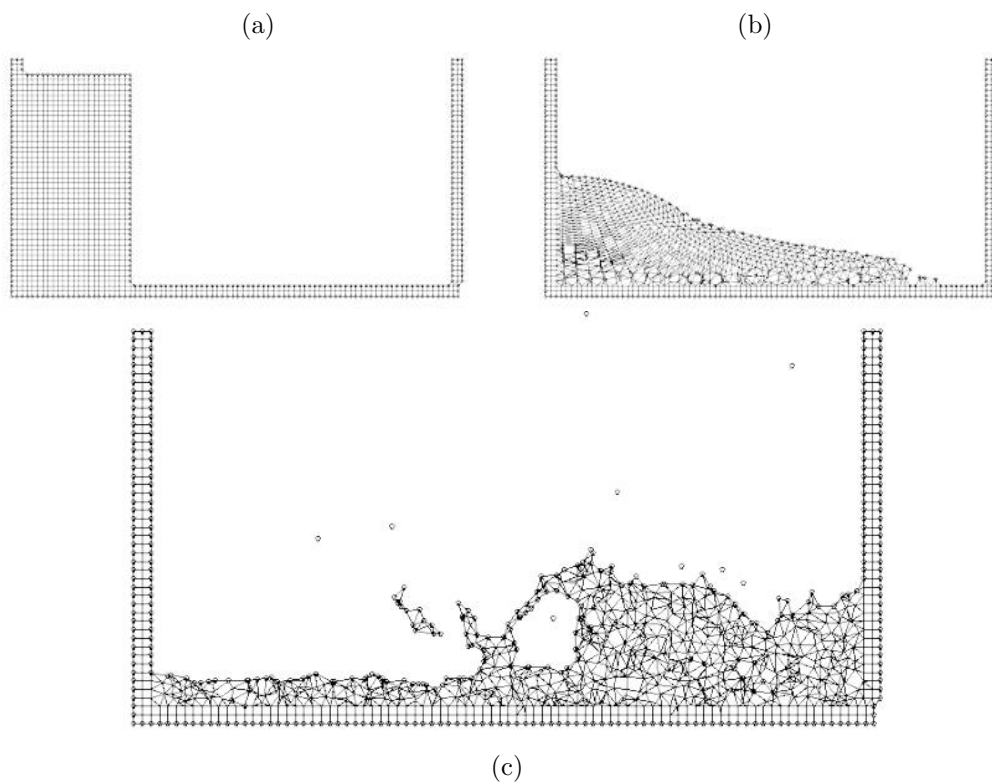


Fig. 2. Breakage of a water column. (a) Discretization of the fluid domain and the solid walls. Boundary nodes are marked with circles. (b) and (c) Mesh in the fluid and solid domains at two different times.

### 3. Lagrangian Equations for an Incompressible Fluid. FIC Formulation

The standard infinitesimal equations for a viscous incompressible fluid can be written in an updated Lagrangian frame as [Oñate (1998); Zienkiewicz and Taylor (2000)]

#### *Momentum*

$$r_{m_i} = 0 \quad \text{in } \Omega \quad (1)$$

#### *Mass balance*

$$r_d = 0 \quad \text{in } \Omega \quad (2)$$

where

$$r_{m_i} = \rho \frac{\partial v_i}{\partial t} + \frac{\partial p}{\partial x_i} - \frac{\partial s_{ij}}{\partial x_j} - b_i \quad (3)$$

$$r_d = \frac{\partial v_i}{\partial x_i} \quad i, j = 1, n_d \quad (4)$$

Above  $n_d$  is the number of space dimensions,  $v_i$  is the velocity along the  $i$ th global axis,  $\rho$  is the (constant) density of the fluid,  $p$  is the absolute pressure (defined positive in compression),  $b_i$  are the body forces and  $s_{ij}$  are the viscous deviatoric stresses related to the viscosity  $\mu$  by the standard expression

$$s_{ij} = 2\mu \left( \dot{\varepsilon}_{ij} - \delta_{ij} \frac{1}{3} \frac{\partial v_k}{\partial x_k} \right) \quad (5)$$

where  $\delta_{ij}$  is the Kronecker delta and the strain rates  $\dot{\varepsilon}_{ij}$  are

$$\dot{\varepsilon}_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (6)$$

In our work we will solve a *modified set of governing* equations derived using a finite calculus (FIC) formulation. The FIC governing equations are [Oñate (1998; 2000); Oñate *et al.* (2001)]

### **Momentum**

$$r_{m_i} - \frac{1}{2} h_j \frac{\partial r_{m_i}}{\partial x_j} = 0 \quad (7)$$

### **Mass balance**

$$r_d - \frac{1}{2} h_j \frac{\partial r_d}{\partial x_j} = 0 \quad (8)$$

The problem definition is completed with the following boundary conditions

$$n_j \sigma_{ij} - t_i + \frac{1}{2} h_j n_j r_{m_i} = 0 \quad \text{on } \Gamma_t \quad (9)$$

$$u_j - u_j^p = 0 \quad \text{on } \Gamma_u \quad (10)$$

and the initial condition is  $u_j = u_j^0$  for  $t = t_0$ . The standard sum convention for repeated indexes is assumed unless otherwise specified.

In Eqs.(7) and (8)  $t_i$  and  $u_j^p$  are surface tractions and prescribed displacements on the boundaries  $\Gamma_t$  and  $\Gamma_u$ , respectively,  $n_j$  are the components of the unit normal vector to the boundary and  $\sigma_{ij}$  are the total stresses given by  $\sigma_{ij} = s_{ij} - \delta_{ij}p$ .

The  $h_i$ 's in above equations are *characteristic lengths* of the domain where balance of momentum and mass is enforced. In Eq.(9) these lengths define the domain where equilibrium of boundary tractions is established. Details of the derivation of Eqs.(7)–(10) can be found in [Oñate (1998; 2000); Oñate *et al.* (2001)].

Eqs.(7)–(10) are the starting point for deriving stabilized finite element methods for solving the incompressible Navier-Stokes equations in a Lagrangian frame of reference using equal order interpolation for the velocity and pressure variables [Idelsohn *et al.* (2002; 2003a; 2003b; 2004); Oñate *et al.* (2003); Aubry *et al.* (2004)]. Application of the FIC formulation to finite element and meshless analysis of fluid flow problems can be found in [García and Oñate (2003); Oñate (2000; 2002); Oñate *et al.* (2000; 2004); Oñate and García (2001); Oñate and Idelsohn (1988)].

### 3.1. Transformation of the mass balance equation. Integral governing equations

The underlined term in Eq.(8) can be expressed in terms of the momentum equations. The new expression for the mass balance equation is (see Appendix)

$$r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} = 0 \quad (11)$$

with

$$\tau_i = \frac{3h_i^2}{8\mu} \quad (12)$$

The  $\tau_i$ 's in Eq.(11), when scaled by the density, are termed *intrinsic time parameters*. Similar values for  $\tau_i$  (usually  $\tau_i = \tau$  is taken) are used in other works from ad-hoc extensions of the 1D advective-diffusive problem [Codina *et al.* (1998); Codina and Blasco (2000); Codina (2002); Codina and Zienkiewicz (2002); Cruchaga and Oñate (1997; 1999); Donea and Huerta (2003); Franca and Frey (1992); Hansbo and Szepessy (1990); Hughes *et al.* (1986; 1989; 1994); Oñate (1998; 2000); Sheng *et al.* (1996); Storti *et al.* (2004); Tezduyar *et al.* (1992); Zienkiewicz and Taylor (2000)].

At this stage it is not longer necessary to retain the stabilization terms in the momentum equations. These terms are critical in Eulerian formulations to stabilize the numerical solution for high values of the convective terms. In the Lagrangian formulation the convective terms disappear from the momentum equations and the FIC terms in these equations are just useful to derive the form of the mass balance equation given by Eq.(11) and can be disregarded there onwards. Consistently, the stabilization terms are also neglected in the Neuman boundary conditions (eqs.(9)).

The weighted residual expression of the final form of the momentum and mass balance equations can be written as

$$\int_{\Omega} \delta u_i r_{m_i} d\Omega + \int_{\Gamma_i} \delta u_i (n_j \sigma_{ij} - t_i) d\Gamma = 0 \quad (13)$$

$$\int_{\Omega} q \left[ r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} \right] d\Omega = 0 \quad (14)$$



The  $r_{m_i}$  term in Eq.(14) and the deviatoric stresses and the pressure terms within  $r_{m_i}$  in Eq.(13) are integrated by parts to give

$$\int_{\Omega} \left[ \delta v_i \rho \frac{\partial v_i}{\partial t} + \delta \dot{\varepsilon}_{ij} (s_{ij} - \delta_{ij} p) \right] d\Omega - \int_{\Omega} \delta v_i b_i d\Omega - \int_{\Gamma_t} \delta v_i t_i d\Gamma = 0 \quad (15)$$

$$\int_{\Omega} q \frac{\partial v_i}{\partial x_i} d\Omega + \int_{\Omega} \left[ \sum_{i=1}^{n_d} \tau_i \frac{\partial q}{\partial x_i} r_{m_i} \right] d\Omega = 0 \quad (16)$$

In Eq.(15)  $\delta \dot{\varepsilon}_{ij}$  are virtual strain rates. Note that the boundary term resulting from the integration by parts of  $r_{m_i}$  in Eq.(16) has been neglected as the influence of this term in the numerical solution has been found to be negligible.

### 3.2. Pressure gradient projection

The computation of the residual terms in Eq.(16) can be simplified if we introduce now the pressure gradient projections  $\pi_i$ , defined as

$$\pi_i = r_{m_i} - \frac{\partial p}{\partial x_i} \quad (17)$$

We express now  $r_{m_i}$  in Eq.(17) in terms of the  $\pi_i$  which then become additional variables. The system of integral equations is therefore augmented in the necessary number of equations by imposing that the residual  $r_{m_i}$  vanishes within the analysis domain (in an average sense). This gives the final system of governing equation as:

$$\int_{\Omega} \left[ \delta v_i \rho \frac{\partial v_i}{\partial t} + \delta \dot{\varepsilon}_{ij} (s_{ij} - \delta_{ij} p) \right] d\Omega - \int_{\Omega} \delta v_i b_i d\Omega - \int_{\Gamma_t} \delta v_i t_i d\Gamma = 0 \quad (18)$$

$$\int_{\Omega} q \frac{\partial v_i}{\partial x_i} d\Omega + \int_{\Omega} \sum_{i=1}^{n_d} \tau_i \frac{\partial q}{\partial x_i} \left( \frac{\partial p}{\partial x_i} + \pi_i \right) d\Omega = 0 \quad (19)$$

$$\int_{\Omega} \delta \pi_i \tau_i \left( \frac{\partial p}{\partial x_i} + \pi_i \right) d\Omega = 0 \quad \text{no sum in } i \quad (20)$$

with  $i, j, k = 1, n_d$ . In Eqs.(20)  $\delta \pi_i$  are appropriate weighting functions and the  $\tau_i$  weights are introduced for symmetry reasons.

## 4. Finite Element Discretization

### 4.1. Derivation of the discretized equations

We choose  $C^0$  continuous interpolations of the velocities, the pressure and the pressure gradient projections  $\pi_i$  over each element with  $n$  nodes. The interpolations are written as

$$v_i = \sum_{j=1}^n N_j \bar{v}_i^j, \quad p = \sum_{j=1}^n N_j \bar{p}^j, \quad \pi_i = \sum_{j=1}^n N_j \bar{\pi}_i^j \quad (21)$$

10 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

where  $(\bar{\cdot})^j$  denotes nodal variables and  $N_j$  are the shape functions [Zienkiewicz and Taylor (2000)]. More details of the mesh discretization process and the choice of shape functions are given in Section 6.

Substituting the approximations (21) into Eqs.(19–20) and choosing a Galerking form with  $\delta v_i = q = \delta \pi_i = N_i$  leads to the following system of discretized equations

$$\mathbf{M}\dot{\bar{\mathbf{v}}} + \bar{\mathbf{g}} - \mathbf{f} = 0 \quad (22a)$$

$$\mathbf{G}^T \bar{\mathbf{v}} + \mathbf{L}\bar{\mathbf{p}} + \mathbf{Q}\bar{\boldsymbol{\pi}} = \mathbf{0} \quad (22b)$$

$$\mathbf{Q}^T \bar{\mathbf{p}} + \hat{\mathbf{M}}\bar{\boldsymbol{\pi}} = \mathbf{0} \quad (22c)$$

where

$$\bar{\mathbf{g}} = \int_{\Omega} \mathbf{B}^T [\mathbf{s} - \mathbf{m}p] d\Omega \quad (23)$$

is the internal nodal force vector derived from the momentum equations,  $\mathbf{s}$  is the deviatoric stress vector,  $\mathbf{B}$  is the strain rate matrix and  $\mathbf{m} = [1, 1, 0]^T$  for 2D problems.

This vector and the rest of the matrices and vectors in Eqs.(22) are assembled from the element contributions given by (for 2D problems)

$$M_{ij} = \int_{\Omega^e} \rho N_i N_j d\Omega \quad , \quad \bar{\mathbf{g}}_i = \int_{\Omega} \mathbf{B}_i^T [\mathbf{s} - \mathbf{m}p] d\Omega \quad , \quad \mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x_1} & 0 \\ 0 & \frac{\partial N_i}{\partial x_2} \\ \frac{\partial N_i}{\partial x_2} & \frac{\partial N_i}{\partial x_1} \end{bmatrix}$$

$$\mathbf{Q} = [\mathbf{Q}^1, \mathbf{Q}^2] \quad , \quad Q_{ij}^k = \int_{\Omega^e} \tau_k \frac{\partial N_i}{\partial x_k} N_j d\Omega \quad (24)$$

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{M}}^1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{M}}^2 \end{bmatrix} \quad , \quad \hat{M}_{ij} = \int_{\Omega^e} \tau_k N_i N_j d\Omega$$

$$\mathbf{f}_i = \int_{\Omega^e} N_i \mathbf{b} d\Omega + \int_{\Gamma^e} N_i \mathbf{t} d\Gamma \quad , \quad \mathbf{b} = [b_1, b_2]^T \quad , \quad \mathbf{t} = [t_1, t_2]^T$$

with  $i, j = 1, n$  and  $k, l = 1, 2$ .

As usual the deviatoric stresses  $s_{ij}$  are related to the strain rates  $\dot{\epsilon}_{ij}$  by Eq.(5)

It can be shown that the system of Eqs.(22) leads to a stabilized numerical solution. For details see [Oñate *et al.* (2003)].

### **Remark 1**

Eq.(22a) can be written in a more explicit form in terms of the velocity and pressure variables as

$$\mathbf{M}\dot{\bar{\mathbf{v}}} + \mathbf{K}\bar{\mathbf{v}} - \mathbf{G}\bar{\mathbf{p}} - \mathbf{f} = 0 \quad (25)$$

where

$$\mathbf{G}_{ij} = \int_{\Omega^e} \mathbf{B}_i^T \mathbf{m} N_j d\Omega \quad \text{and} \quad \mathbf{K}_{ij} = \int_{\Omega^e} \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j d\Omega \quad (26)$$

where  $\mathbf{D}$  is the constitutive matrix. For 2D problems

$$\mathbf{D} = \mu \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (27)$$

### 5. Fractional Step Method for Fluid-Structure Interaction Analysis

A simple and effective iterative algorithm can be obtained by splitting the pressure from the momentum equations as follows

$$\bar{\mathbf{v}}^* = \bar{\mathbf{v}}^n - \Delta t \mathbf{M}^{-1} [\mathbf{g}^{n+\theta_1,j} - \mathbf{f}^{n+1}] \quad (28a)$$

$$\bar{\mathbf{v}}^{n+1,j} = \bar{\mathbf{v}}^* + \Delta t \mathbf{M}^{-1} \int_{\Omega^{n+\theta_1,j}} \mathbf{B}^T \mathbf{m} \delta p = \bar{\mathbf{v}}^* + \Delta t \mathbf{M}^{-1} \mathbf{G} \delta \bar{\mathbf{p}} \quad (28b)$$

In Eq.(28a)

$$\mathbf{g}^{n+\theta_1,j} = \int_{\Omega^{n+\theta_1,j}} \mathbf{B}^T [\mathbf{s}^{n+\theta_1,j} - \alpha \mathbf{m}^T p^n] d\Omega$$

and  $\alpha$  is a variable taking values equal to zero or one. For  $\alpha = 0$ ,  $\delta p \equiv p^{n+1,j}$  and for  $\alpha = 1$ ,  $\delta p = \Delta p$ . Note that in both cases the sum of Eqs.(28a) and (28b) gives the time discretization of the momentum equations with the pressures computed at  $t^{n+1}$ .

In above equations and in the following superindex  $j$  denotes the iteration number within each time step.

The value of  $\bar{\mathbf{v}}^{n+1,j}$  from Eq.(28b) is substituted now into Eq.(22b) to give

$$\mathbf{G}^T \bar{\mathbf{v}}^* + \Delta t \mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} \delta \bar{\mathbf{p}} + \mathbf{L} \bar{\mathbf{p}}^{n+1,j} + \mathbf{Q} \bar{\boldsymbol{\pi}}^{n+\theta_2,j} = 0 \quad (29a)$$

The product  $\mathbf{G}^T \mathbf{M}^{-1} \mathbf{G}$  can be approximated by a laplacian matrix, i.e.

$$\mathbf{G}^T \mathbf{M}^{-1} \mathbf{G} = \hat{\mathbf{L}} \quad \text{with} \quad \hat{\mathbf{L}} \simeq \int_{\Omega^e} \frac{1}{\rho} \boldsymbol{\nabla}^T N_i \boldsymbol{\nabla} N_j d\Omega \quad (29b)$$

In above equations  $\theta_1$  and  $\theta_2$  are algorithmic parameters ranging between zero and one. A discussion of the choice of  $\theta_1$  and  $\theta_2$  is given below.

A semi-implicit algorithm can be derived as follows. For each iteration:

**Step 1** Compute  $\bar{\mathbf{v}}^*$  from Eq.(28a) with  $\mathbf{M} = \mathbf{M}_d$  where subscript  $d$  denotes hereonwards a diagonal matrix.

**Step 2** Compute  $\delta \bar{\mathbf{p}}$  and  $\bar{\mathbf{p}}^{n+1}$  from Eq.(29a) as

$$\delta \bar{\mathbf{p}} = -(\mathbf{L} + \Delta t \hat{\mathbf{L}})^{-1} [\mathbf{G}^T \bar{\mathbf{v}}^* + \mathbf{Q} \bar{\boldsymbol{\pi}}^{n+\theta_2,j} + \alpha \mathbf{L} \bar{\mathbf{p}}^n] \quad (30a)$$

12 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

$$\mathbf{p}^{n+1,j} = \mathbf{p}^n + \delta\mathbf{p} \quad (30b)$$

**Step 3** Compute  $\bar{\mathbf{v}}^{n+1,j}$  from Eq.(28b) with  $\mathbf{M} = \mathbf{M}_d$

**Step 4** Compute  $\bar{\boldsymbol{\pi}}^{n+1,j}$  from Eq.(22c) as

$$\bar{\boldsymbol{\pi}}^{n+1,j} = -\hat{\mathbf{M}}_d^{-1} \mathbf{Q}^T \bar{\mathbf{p}}^{n+1,j} \quad (31)$$

**Step 5** Solve for the movement of the structure due to the fluid flow forces.

This implies solving the dynamic equations of motion for the structure written as

$$\mathbf{M}_s \ddot{\mathbf{d}} + \mathbf{K}_s \mathbf{d} = \mathbf{f}_{\text{ext}} \quad (32)$$

where  $\mathbf{d}$  and  $\ddot{\mathbf{d}}$  are respectively the displacement and acceleration vectors of the nodes discretizing the structure,  $\mathbf{M}_s$  and  $\mathbf{K}_s$  are the mass and stiffness matrices of the structure and  $\mathbf{f}_{\text{ext}}$  is the vector of external nodal forces accounting for the fluid flow forces induced by the pressure and the viscous stresses. Clearly the main driving forces for the motion of the structure is the fluid pressure which acts as normal a surface traction on the structure. Indeed Eq.(32) can be augmented with an appropriate damping term. The form of all the relevant matrices and vectors can be found in standard books on FEM for structural analysis [Zienkiewicz and Taylor (2000)].

Solution of Eq.(32) in time can be performed using implicit or fully explicit time integration algorithms. In both cases the values of the nodal displacements, velocities and accelerations of the structure at  $t^{n+1}$  are found for the  $j$ th iteration.

### **Step 6**

Update the mesh nodes in a Lagrangian manner as

$$\mathbf{x}_i^{n+1,j} = \mathbf{x}_i^n + \bar{\mathbf{v}}_i^{n+1,j} \Delta t \quad (33)$$

### **Step 7**

Generate a new mesh. This can be effectively performed using the procedure described in Section 6.

### **Step 8**

Check the convergence of the velocity and pressure fields in the fluid and the displacements strains and stresses in the structure. If convergence is achieved move to the next time step, otherwise return to step 1 for the next iteration with  $j + 1 \rightarrow j$ .

Despite the motion of the nodes within the iterative process, in general there is not need to regenerate the mesh at each iteration. A new mesh is typically generated after a prescribed number of converged time steps, or when the nodal displacements induce significant geometrical distortions in some elements. *In the*

examples presented in the paper the mesh in the fluid domain has been regenerated at each time step.

The boundary conditions are applied as follows. No condition is applied in the computation of the fractional velocities  $\mathbf{v}^*$  in Eq.(28a). The prescribed velocities at the boundary are applied when solving for  $\bar{\mathbf{v}}^{n+1,j}$  in step 3. The prescribed pressures at the boundary are imposed by making zero the pressure increments at the relevant boundary nodes and making  $\bar{\mathbf{p}}^n$  equal to the prescribed pressure values.

Details of the treatment of the contact conditions at the solid-fluid interface are given in Section 8 [Idelsohn *et al.* (2004)].

Note that solution of steps 1, 3 and 4 does not require to solve a system of equations as a diagonal form is chosen for  $\mathbf{M}$  and  $\hat{\mathbf{M}}$ . The whole solution process within a time step can be linearized by choosing  $\theta_1 = \theta_2 = 0$  and now the iteration loop is not longer necessary. The implicit solution for  $\theta_1 = \theta_2 = 1$  is however very effective as larger time steps can be used. This requires some iterations within steps 1–8 until converged values for the fluid and solid variables and the new position of the mesh nodes at time  $n + 1$  are found.

In the examples presented in the paper the time increment size has been chosen as

$$\Delta t = \min(\Delta t_i) \quad \text{with} \quad \Delta t_i = \frac{|\mathbf{v}|}{h_i^{\min}} \quad (34)$$

where  $h_i^{\min}$  is the distance between node  $i$  and the closest node in the mesh.

### **Remark 2**

Although not explicitly mentioned for  $\theta_1 = 1$  all matrices and vectors in Eqs.(28)–(32) are computed at the final configuration  $\Omega^{n+1,j}$ . This means that the integration domain changes for each iteration and, hence, all the terms involving space derivatives must be updated at each iteration. This problem disappears if  $\Omega^n$  is taken as the reference configuration ( $\theta_1 = 0$ ) as this remains fixed during the iterations. The penalty to pay in this case, however, is the evaluation of the Jacobian matrix at each iterations [Aubry *et al.* (2004)].

## **6. Treatment of Contact Between Fluid and Solid Interfaces**

The condition of prescribed velocities or pressures at the solid boundaries in the PFEM are applied in strong form to the boundary nodes. These nodes might belong to fixed external boundaries or to moving boundaries linked to the interacting solids. In some problems it is useful to define a layer of nodes adjacent to the external boundary in the fluid where the condition of prescribed velocity is imposed. These nodes typically remain fixed during the solution process. Contact between water particles and the solid boundaries is accounted for by the incompressibility condition which *naturally prevents the water nodes to penetrate into the solid boundaries*. This

simple way to treat the water-wall contact is another attractive feature of the PFEM formulation.

## 7. Generation of a New Mesh

One of the key points for the success of the Lagrangian flow formulation here described is the fast regeneration of a mesh at every time step on the basis of the position of the nodes in the space domain. In our work the mesh is generated using the so called extended Delaunay tessellation (EDT) presented in [Idelsohn *et al.* (2003a; 2003c; 2004)]. The EDT allows to generate non standard meshes combining elements of arbitrary polyhedral shapes (triangles, quadrilaterals and other polygons in 2D and tetrahedra, hexahedra and arbitrary polyhedra in 3D) in a computing time of order  $n$ , being  $n$  the total number of nodes in the mesh (Figure 3). The  $C^0$  continuous shape functions of the elements can be simply obtained using the so called meshless finite element interpolation (MFEM). Details of the mesh generation procedure and the derivation of the MFEM shape functions can be found in [Idelsohn *et al.* (2003a; 2003c; 2004)].

Once the new mesh has been generated at each time step the numerical solution is found using the fractional step algorithm described in the previous section.

The combination of elements with different geometrical shapes in the same mesh is one of the innovative aspects of the Lagrangian formulation presented here.

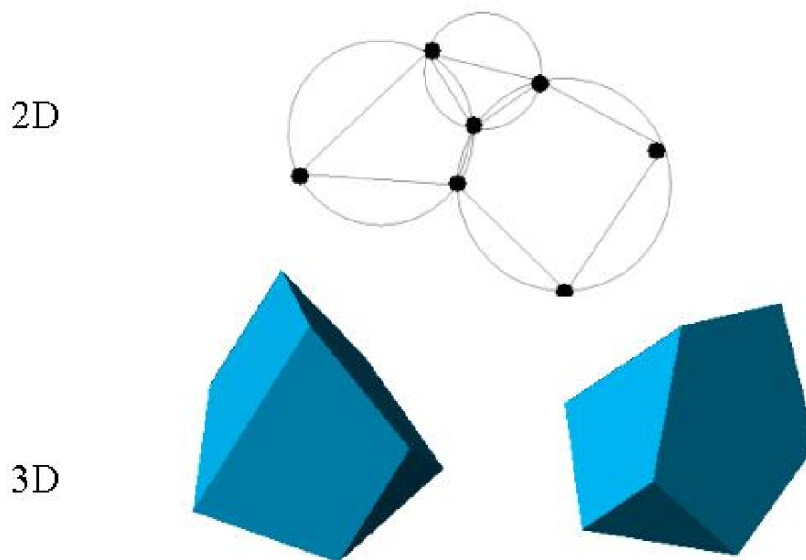


Fig. 3. Generation of non standard meshes combining different polygons (in 2D) and polyhedra (in 3D) using the extended Delaunay technique.

## 8. Identification of Boundary Surfaces

One of the main tasks in the PFEM is the correct definition of the boundary domain. Sometimes, boundary nodes are explicitly defined as such differently from internal nodes. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundary nodes.

The use of the extended Delaunay partition makes it easier to recognize boundary nodes.

Considering that the nodes follow a variable  $h(x)$  distribution, where  $h(x)$  is the minimum distance between two nodes, the following criterion has been used. *All nodes on an empty sphere with a radius greater than  $\alpha h$ , are considered as boundary nodes.* In practice,  $\alpha$  is a parameter close to, but greater than one. Note that this criterion is coincident with the Alpha Shape concept [Edelsbrunner and Mücke (1999)].

Once a decision has been made concerning which nodes are on the boundaries, the boundary surface must be defined. It is well known that in 3-D problems the surface fitting a number of nodes is not unique. For instance, four boundary nodes on the same sphere may define two different boundary surfaces, a concave one and a convex one.

In this work, the boundary surface is defined by all the polyhedral surfaces (or polygons in 2D) having all their nodes on the boundary and belonging to just one polyhedron.

Figure 4 shows example of the boundary recognition using the Alpha Shape technique.

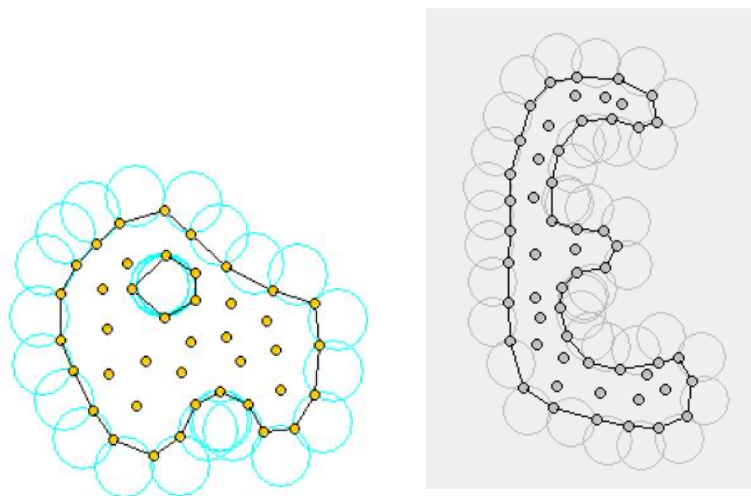


Fig. 4. Examples of boundary recognition with the Alpha Shape method. Empty circles with radius greater than  $\alpha h(x)$  define the boundary particles.

The correct boundary surface is important to define the normal external to the surface. Furthermore, in weak forms (Galerkin) such as those used here a correct evaluation of the volume domain is also important. In the criterion proposed above, the error in the boundary surface definition is proportional to  $h$  which is an acceptable error. The only way to obtain a more accurate boundary surface definition is by reducing the distance between the nodes.

The method described also allows to identify isolated fluid particles outside the main fluid domain. These particles are treated as part of the external boundary where the pressure is fixed to the atmospheric value.

Figure 5 shows a schematic example of the process to identify individual particles (or a group of particles) starting from a given collection of nodes. A practical application of the method for identifying free surface particles is shown in Figure 6. The example corresponds to the analysis of the motion of a fluid within an oscillating ellipsoidal container. Note that the method captures the individual water drops departing from the free surface during the fluid motion.

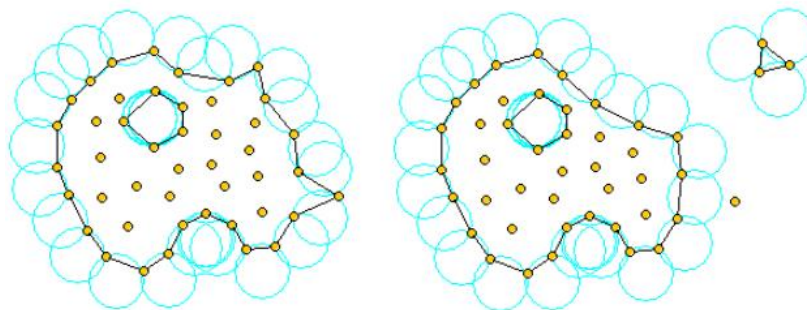


Fig. 5. Identification of individual particles (or a group of particles) starting from a given collection of nodes.

## 9. Modelling the Structure as a Viscous Fluid

A simple and yet effective way to analyze the rigid motion of solid bodies in fluids with the Lagrangian flow description is to model the solid as a fluid with a viscosity much higher than that of the surrounding fluid. The fractional step scheme of Section 5 can be readily applied skipping now step 5 and solving now for the simultaneous motion of both fluid domains (the actual fluid and the fictitious fluid modelling the quasi-rigid body). Examples of this type are presented in Sections 10.3 and 10.4.

Indeed this approach can be further extended to account for the elastic deformation of the solid treated now as a visco-elastic fluid. This will however introduce some complexity in the formulation and the full coupled FSI scheme described in Section 5 is preferable.



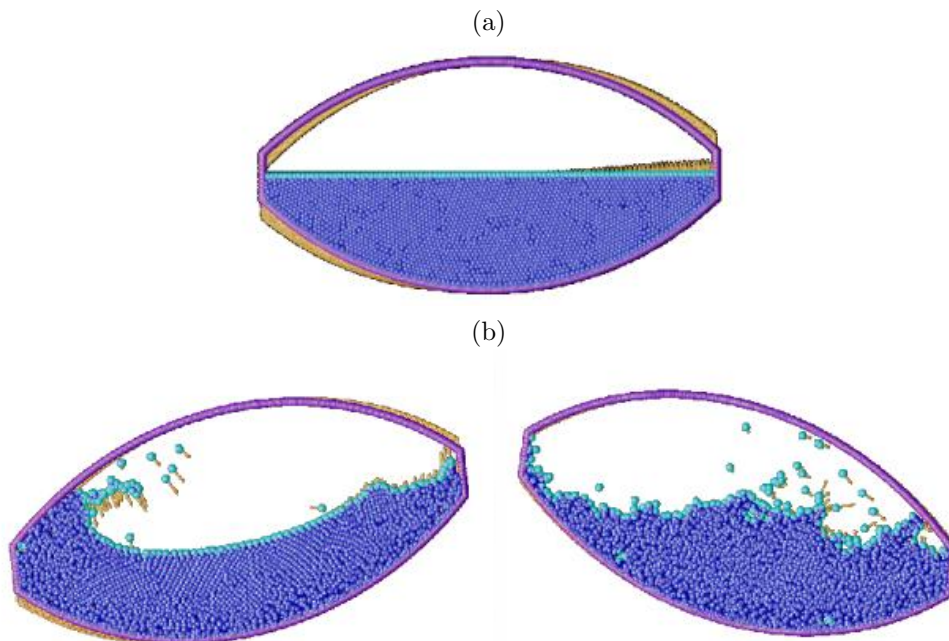


Fig. 6. Motion of a liquid within an oscillating container. (a) Original distribution of particles (nodes) prior to the oscillation. (b) Position of the liquid particles at two different times. The boundary particles representing the free surface, the fluid drops and the container wall are plotted with a lighter colour. Arrows indicate velocity vectors for each particle.

## 10. Examples

The examples chosen show the applicability of the PFEM to solve problems involving large fluid motions and FSI situations. The fractional step algorithm of Section 5 with  $\theta_2 = 1$  and  $\alpha = 1$  has been used in all cases.

In examples 10.1–10.10 a value of  $\theta_1 = 1$  has been chosen. This basically means that the final configuration  $\Omega^{n+1,j}$  has been taken as the reference configuration at each iteration. In example 10.11  $\theta_1 = 0$  has been selected and, hence, the initial configuration  $\Omega^n$  has been taken as a fixed reference configuration for all the iterations within a time step.

### 10.1. Collapse of a water column

The first problem solved to show the potential of the PFEM is the study of the collapse of a water column. This problem was solved by Koshizu and Oka (1996) both experimentally and numerically. It has become a classical example to validate the Lagrangian formulation for fluid flows. The water is initially kept within a rectangular container including a removable vertical board. A double layer of nodes in the solid walls is used in order to prevent water nodes not to exit the

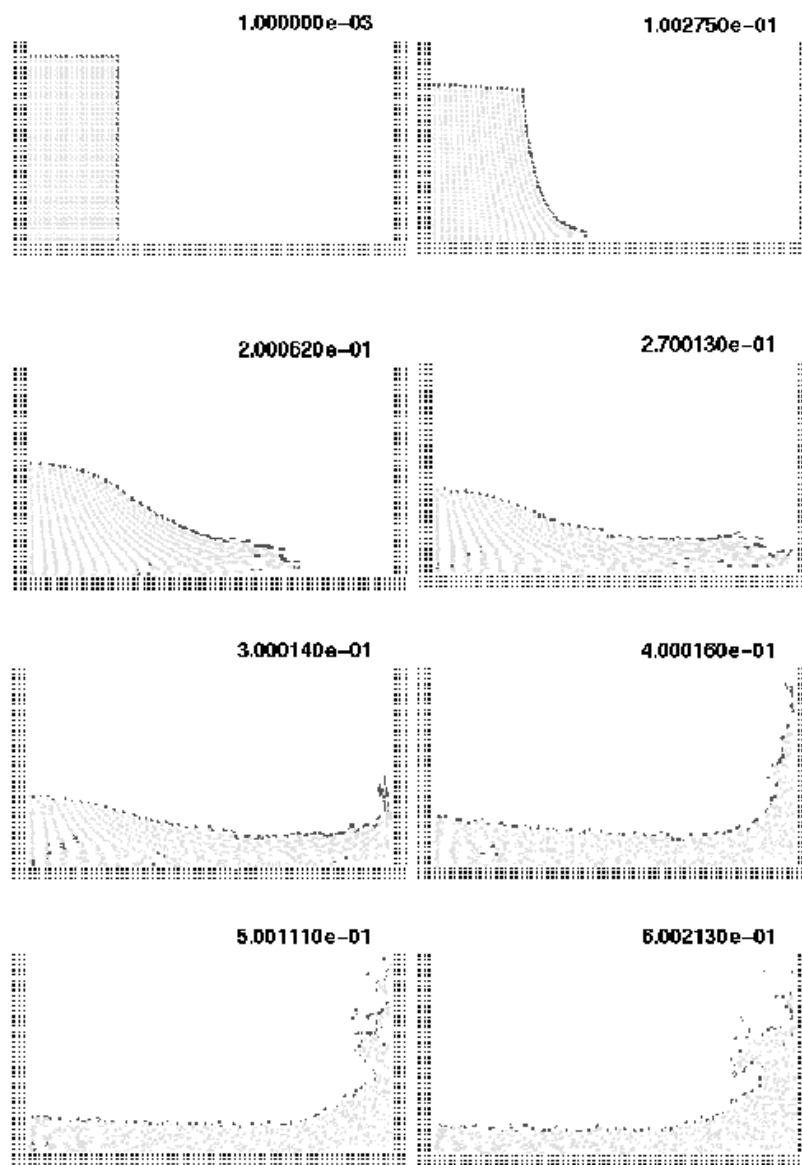


Fig. 7. Water column collapse at different time steps.

analysis domain. The boundary conditions impose zero velocity at the wall nodes and zero (atmospheric) pressure at the free surface. Figures 7b and 7c show the mesh discretizing the water domain and the solid walls at two different times of the analysis. Note that the method allows to follow the large motion of the water particles including the separation of some water drops. The collapse starts at time  $t = 0$ , when the board is removed. Viscosity and surface tension are neglected in

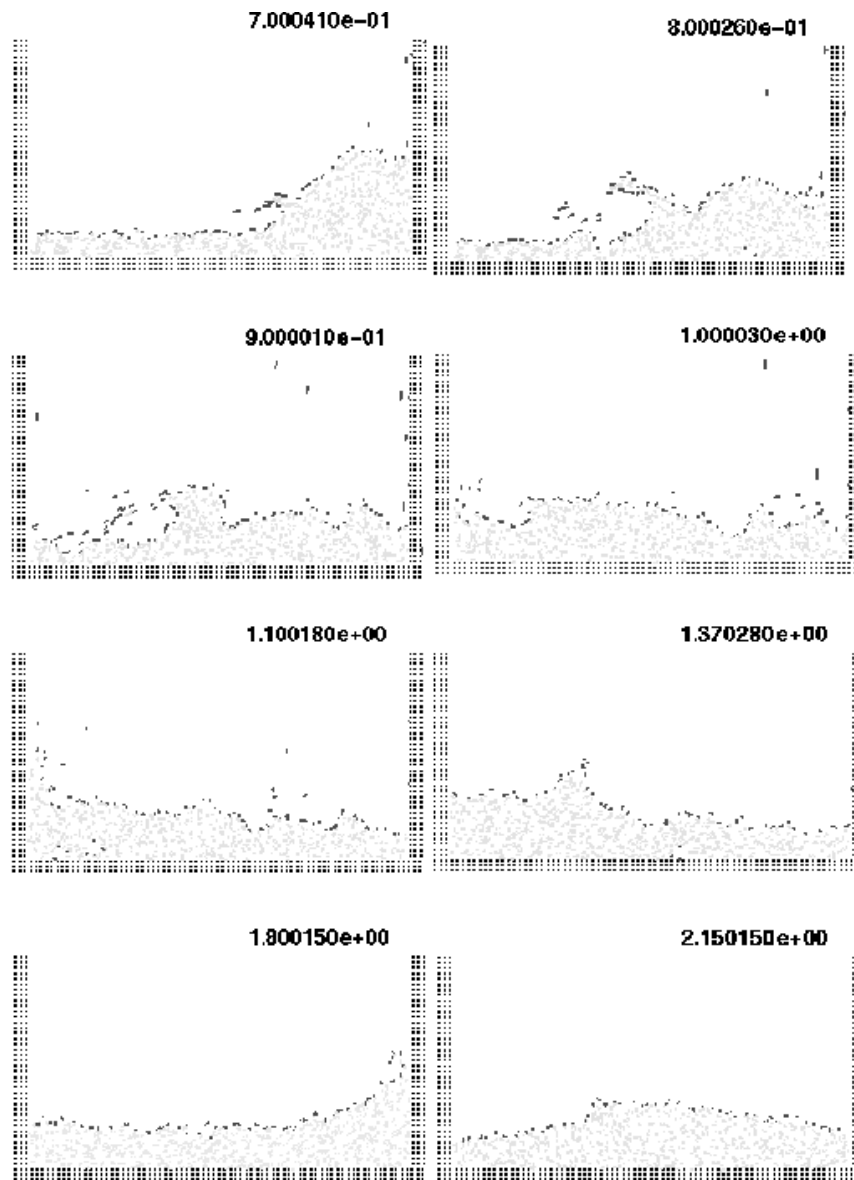


Fig. 7. cont.

the analysis. Figure 7 shows the point positions at different time steps. The dark points represent the free-surface detected with the algorithm described in Section 8. The internal points are shown in a gray colour and the fixed points in black. The meshes generated at different times during the fluid motion are shown in Figure 7.

The water is running on the bottom wall until, at 0.3 sec it impinges on the right vertical wall. Breaking waves appear at 0.6 sec. At about 1 sec. the water

20 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

reaches the left wall. Agreement with the experimental results of [Koshizuka and Oka (1996)] both in the shape of the free surface as well as in its time evolution are excellent.

The 3D solution of the same problem is shown in Figure 8. More information on the PFEM solution of this problem can be found in Idelsohn *et al.* (2004).

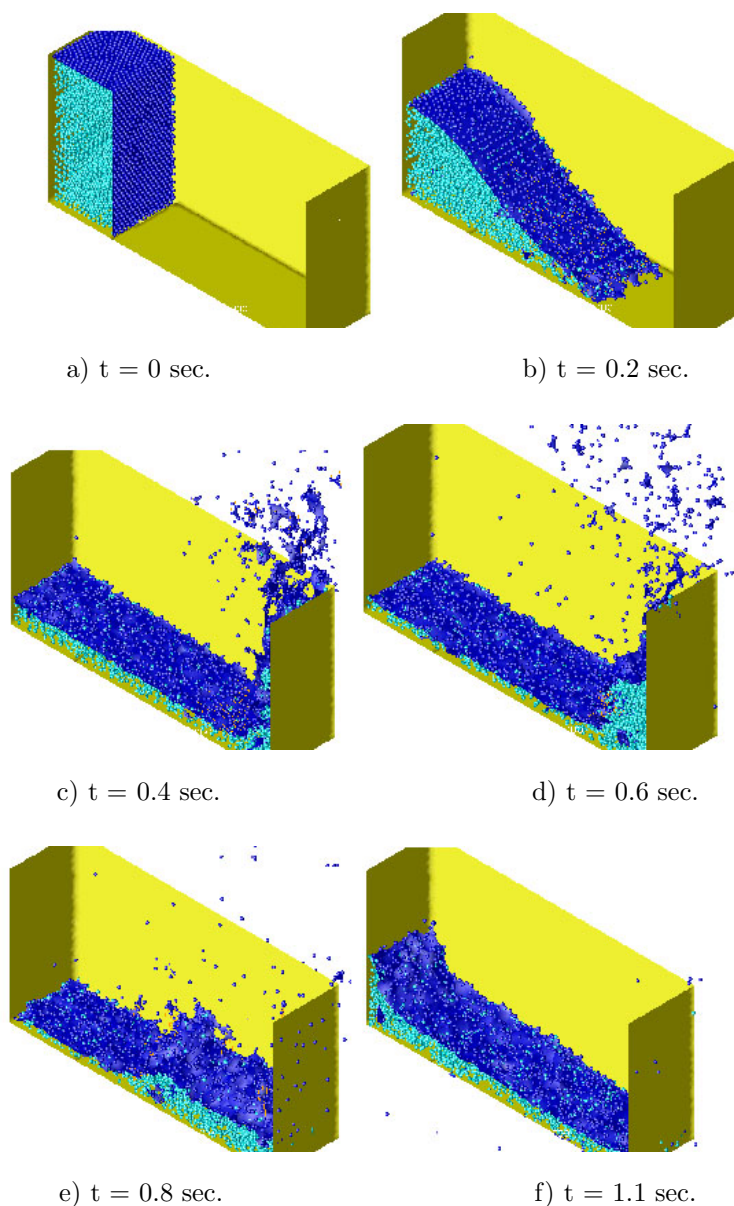


Fig. 8. Water column collapse in a 3D domain.

### 10.2. Sloshing problems

The simple problem of the free oscillation of an incompressible liquid in a container is considered next. Numerical solutions for this problem can be found in several references [Radovitzki and Ortiz (1998)]. This problem is interesting because there is an analytical solution for small amplitudes. Figure 9 shows a schematic view of the problem and the point distribution in the initial position. The dark points represent the fixed points on the walls where the velocity is fixed to zero.

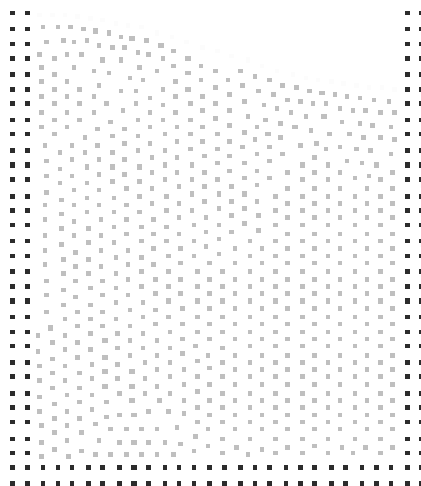


Fig. 9. Sloshing. Initial point distribution.

Figure 10 shows the time evolution of the amplitude compared with the analytical results for the near inviscid case. Little numerical viscosity is observed on the phase wave and amplitude in spite of the relative poor point distribution.

The analytical solution is only acceptable for small wave amplitudes. For larger amplitudes, additional waves are overlapping and, finally, the wave breaks and also some particles separate from the fluid domain due to their large velocity. Figure 11 shows the numerical results obtained with the PFEM for larger sloshing amplitudes. Breaking waves as well as separation effects can be seen on the free-surface. This particular and very complicated effect is well represented by the PFEM.

In order to test the potentiality of the PFEM in a 3D domain, the same sloshing problem was solved in 3D. Figure 12 show the different point position at two time steps. Each point position was represented by a sphere and only a half of the fixed recipient is represented on the figure. This sphere representation is only used in order to improve the visualization of the numerical results.

22 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

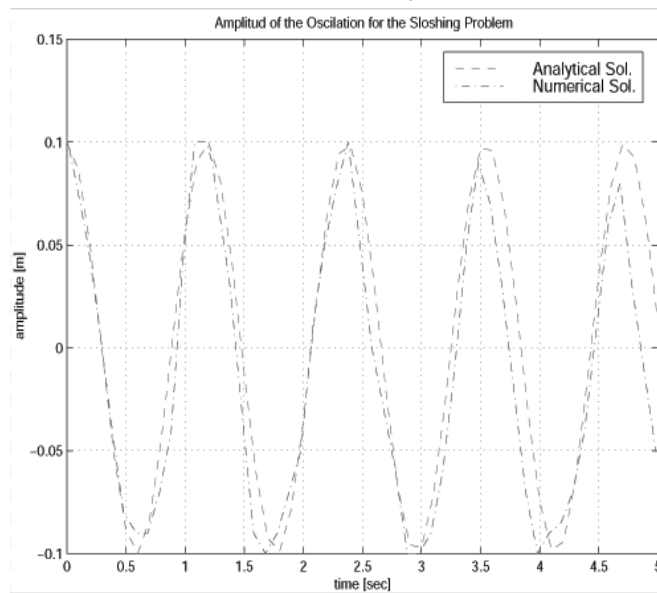


Fig. 10. Sloshing: Comparison of the numerical and analytical solutions.

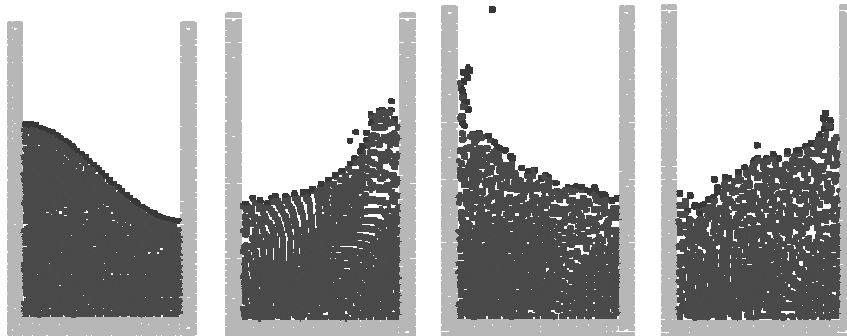


Fig. 11. PFEM results for a large amplitude sloshing problems.

### 10.3. *Wave breaking on a beach*

A simulation of the propagation of a water wave and its breaking due to shoaling over a plane slope is presented next. This example was numerically studied in [Radovitzki and Ortiz (1998)] with a Lagrangian formulation using directly the standard Finite Element Method with remeshing. There is also an analytical solution for a simplified approximation that is used for comparisons [Laitone (1960)]. The geometry of the problem as well as a discussion of the analytical solution may be found in [Laitone (1960)]. Figure 13 shows the initial point distribution and Figure 11 a comparison with the analytical free-surface at different time steps.

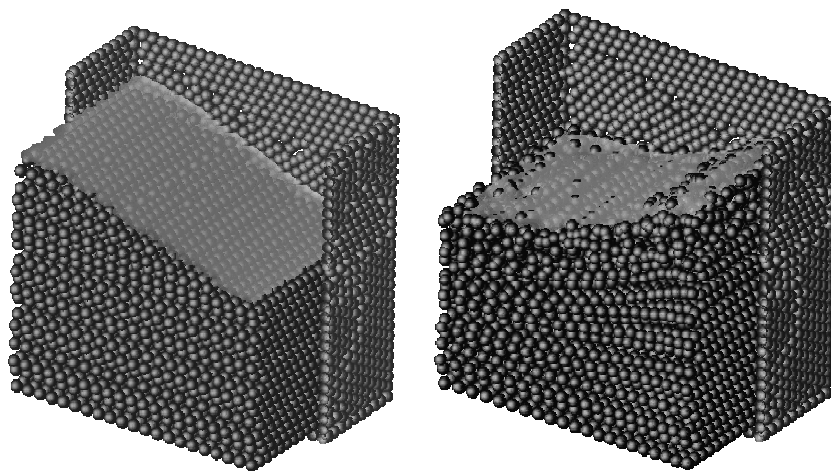


Fig. 12. 3D sloshing problem.

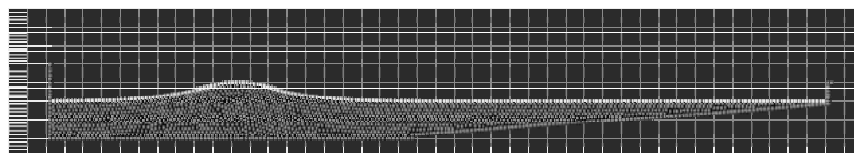


Fig. 13. Wave breaking on a beach. Initial geometry and point positions.

Initially (Figures 14a. and 14b.) the wave travels over a constant depth bottom towards the slope with no ostensible change of shape. Strongly non-linear effects appear when the wave hits the slope (Fig. 14c.). The crest of the wave accelerates until it reaches the shore (Fig. 14d.). At this time the comparisons with the analytical solution are in agreement only in the wave position. The shape of the wave obtained with the numerical solution is totally different. The reason is that the analytical solution gives symmetrical shape waves, which are not physical, before the breaking process. Subsequently, a water jet is formed at the crest plunge making the breaking wave (Figs. 14e. and 14f.) and coming in contact with the nearly still surface of the water ahead. In Ref. [Radovitzki and Ortiz (1998)] the computation is stopped before this contact point. Using the methodology proposed in this paper, the analysis may be continued until the end. In Figs. 14g. and 14h. the wave finally hits a lateral wall (introduced in the model to stop the lateral effects) producing drop separations, and then coming back toward to the left as a new wave.

The ability of the PFEM to accurately simulate the various stages of the wave breaking is noteworthy.

24 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

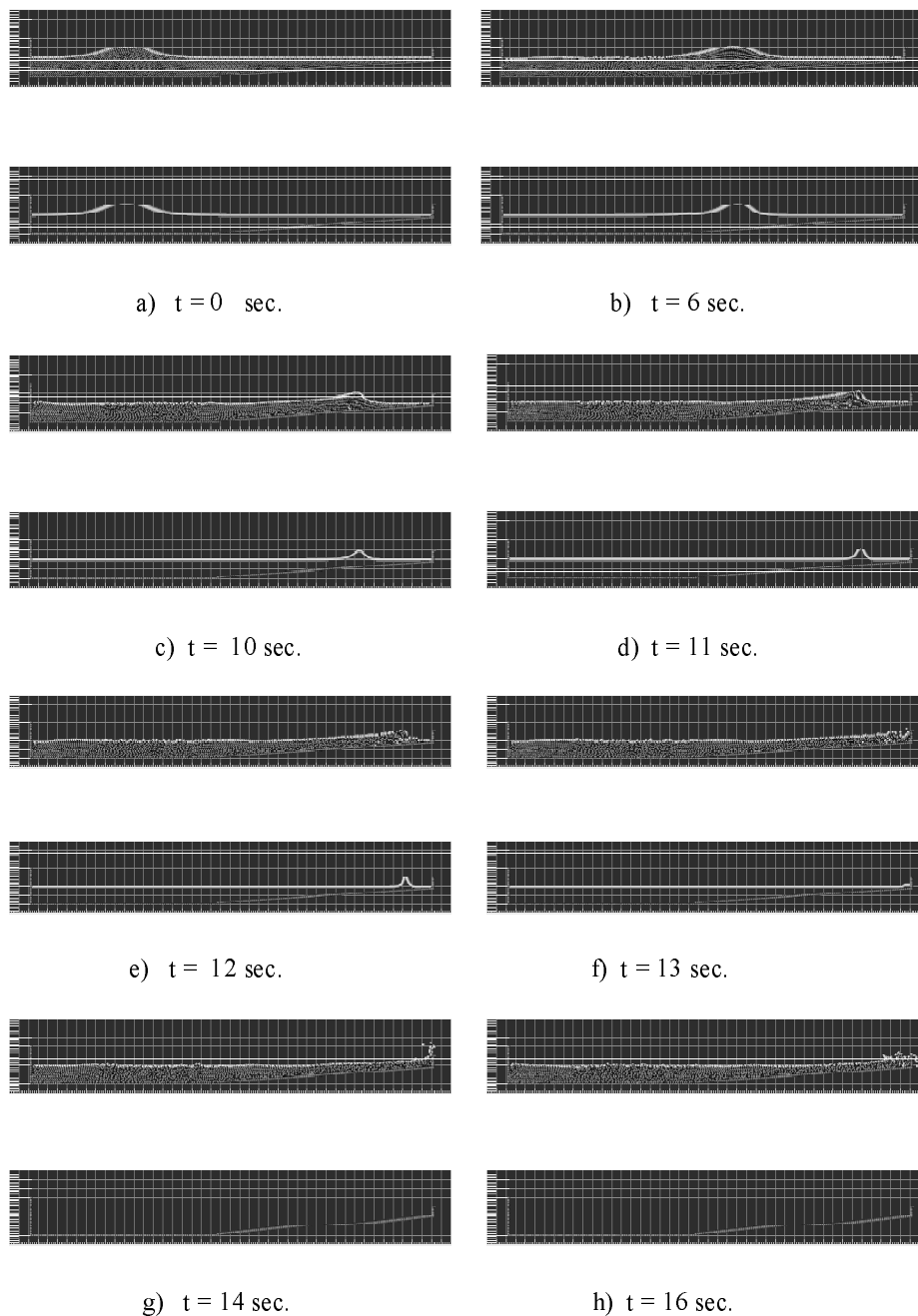


Fig. 14. Wave breaking on a beach. Comparison with analytical results at different time steps. Top: PFEM solution. Bottom: Analytical solution.



In order to show the power of the PFEM, the same problem was solved in a 3D domain. Now, the initial position of the wave was given an oblique angle with the beach line. In this way, 3D effects show more clearly. When the wave hits the slope, the crest of the wave accelerates differently accordingly with the depth, inducing the wave to correct its oblique position and break parallel to the beach. The results may be seen in Figure 15 for different time steps.

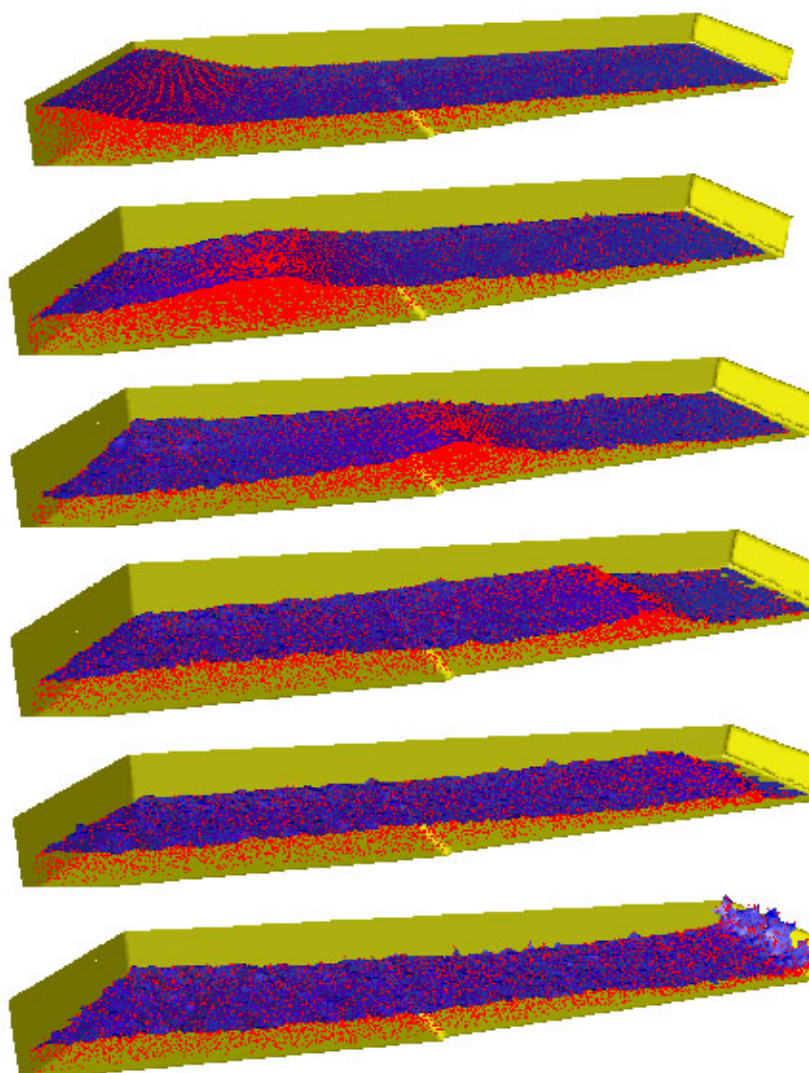


Fig. 15. Breaking wave on a beach. Oblique wave on a 3D domain.

#### **10.4. *Fixed ship hit by wave***

This example is a very schematic representation of a ship when is hit by a big wave produced by the collapse of a water recipient (Figure 16). The ship can not move and initially the free-surface near the ship is horizontal. Fixed nodes represent the ship as well as the domain walls. The example tests the suitability of the PFEM to solve water-wall contact situations even in the presence of curved walls. Note the breaking and splashing of the waves under the ship prow and the rebound of the incoming wave. It is also interesting to see the different water-wall contact situations at the internal and external ship surfaces and the moving free-surface at the back of the ship.

#### **10.5. *Horizontal motion of a rigid ship in a reservoir***

In the next example (Figure 17) the same ship of the previous example moves now horizontally at a fixed velocity in a water reservoir. The free-surface, which was initially horizontal, takes the correct position at the ship prow and stern. Again, the complex water-wall contact problem is naturally solved in the curved prow region.

#### **10.6. *Semi-submerged rotating water mill***

The example shown in Figure 18 is the analysis of a rotating water mill semi-submerged in water. The blades of the mill are treated as a rigid body with an imposed rotating velocity, while the water is initially in a stationary flat position. Fluid structure interactions with free-surfaces and water fragmentation are well reproduced in this example.

#### **10.7. *Floating wood piece***

The next example shows an initially stationary recipient with a floating piece of wood where a wave is produced on the left side. The wood has been simulated by a liquid of higher viscosity as described in Section 9. The wave intercepts the wood piece producing a breaking wave and displacing the floating wood as shown in Figure 19.

#### **10.8. *Ships hit by an incoming wave***

In the example of Figure 20 the motion of a fictitious rigid ship hit by an incoming wave is analyzed. The dynamic motion of the ship is induced by the resultant of the pressure and the viscous forces acting on the ship boundaries. The horizontal displacement of the gravity center of the ship was fixed to zero. In this way, the ship moves only vertically although it can freely rotate. The position of the ship boundary at each time step defines a moving boundary condition for the free surface particles in the fluid. Figure 20 shows instants of the motion of the ship and the

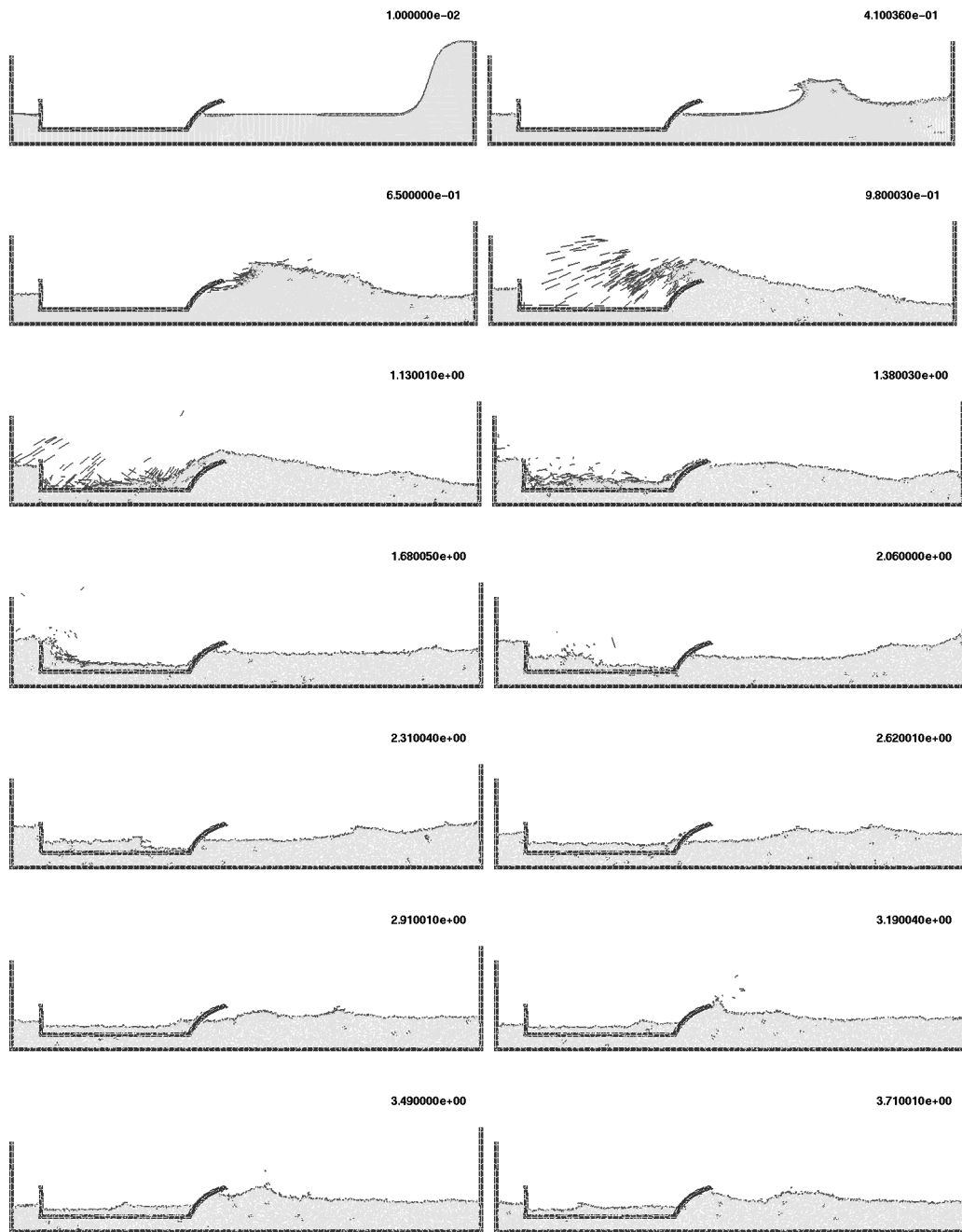


Fig. 16. Fixed ship hit by incoming wave

28 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

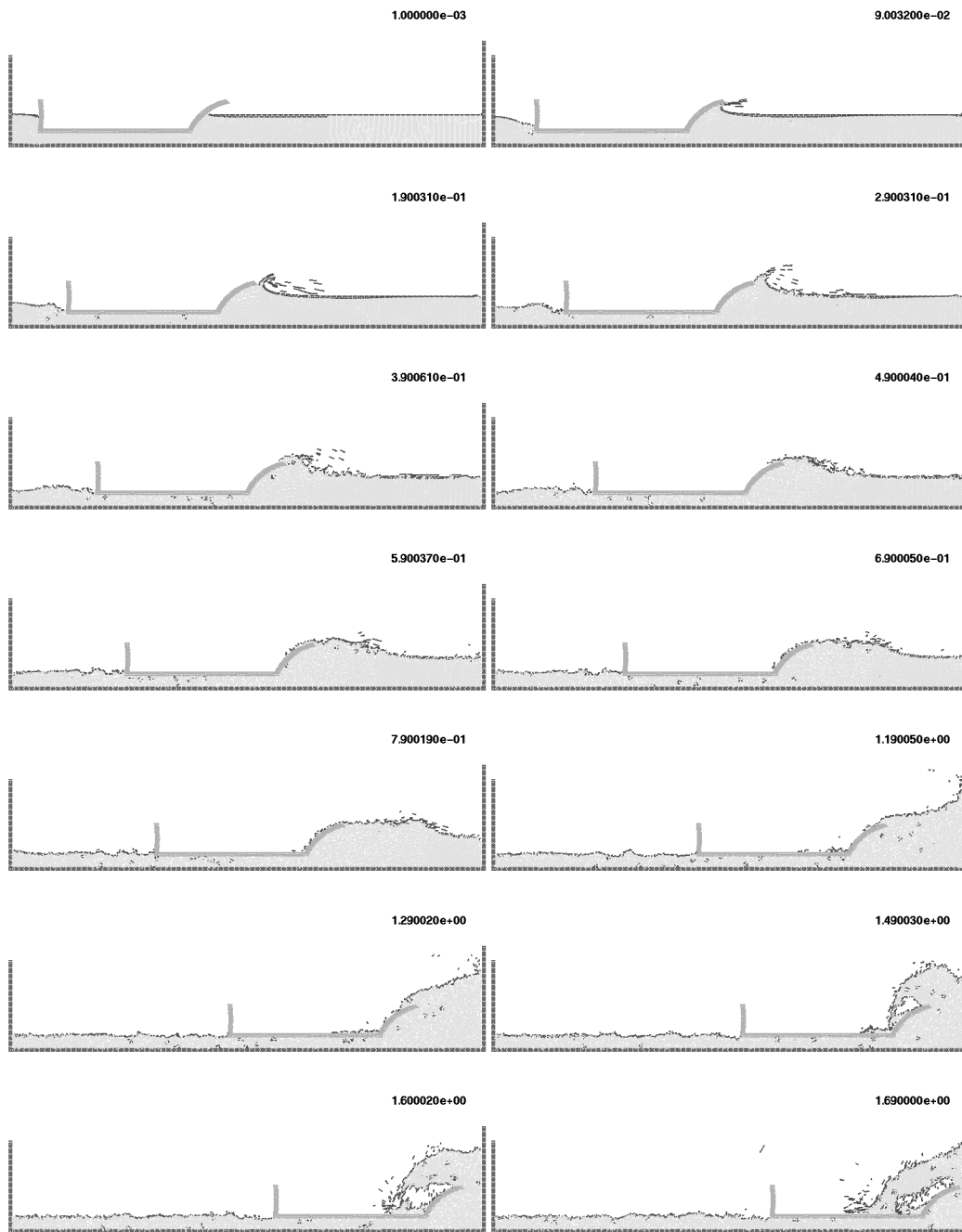


Fig. 17. Moving ship with fixed velocity

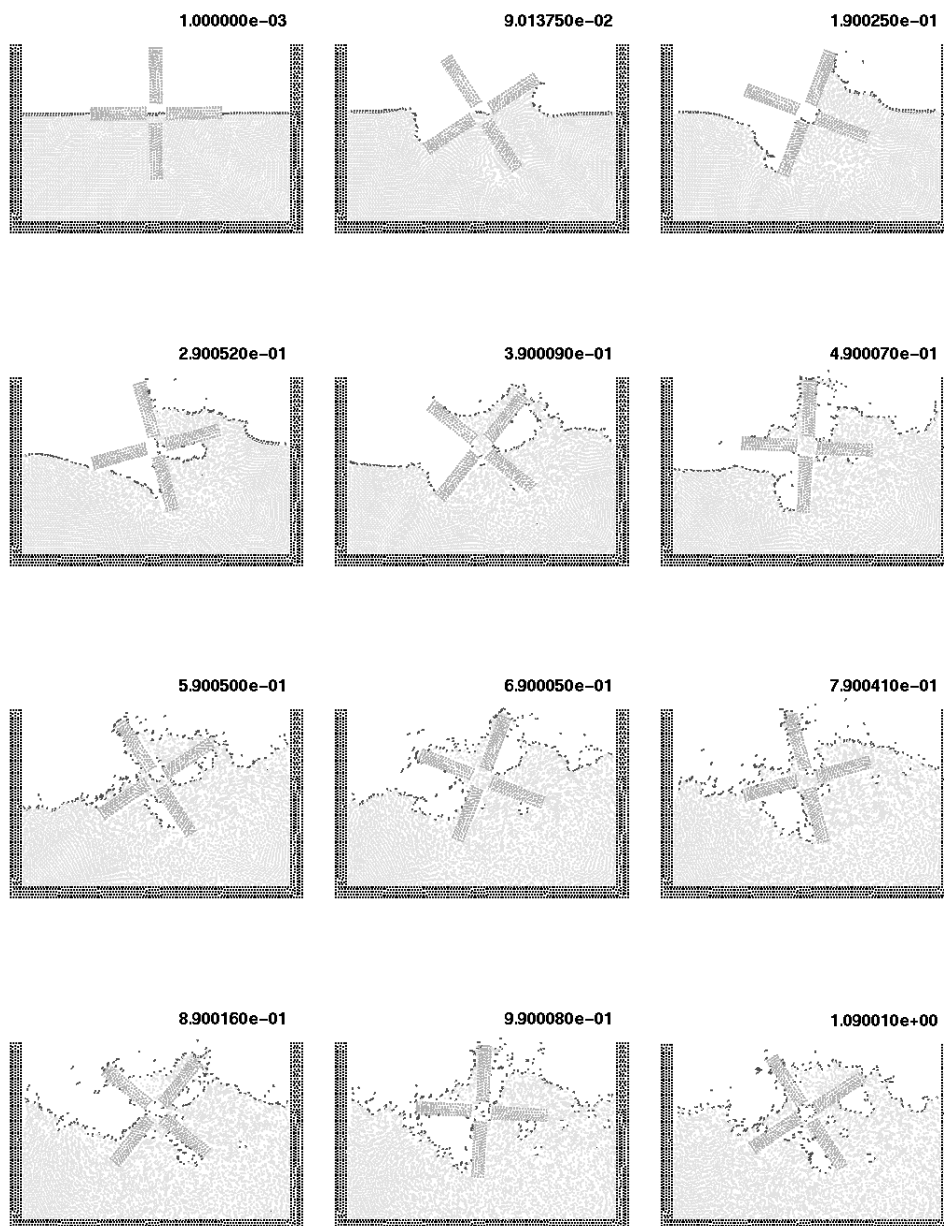


Fig. 18. Rotating water mill.

surrounding fluid. It is interesting to see the breaking of a wave on the ship prow as well as on the stern when the wave goes back. Note that some water drops slip over the ship deck.

Figure 21 shows the analysis of a similar problem using precisely the same ap-

30 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

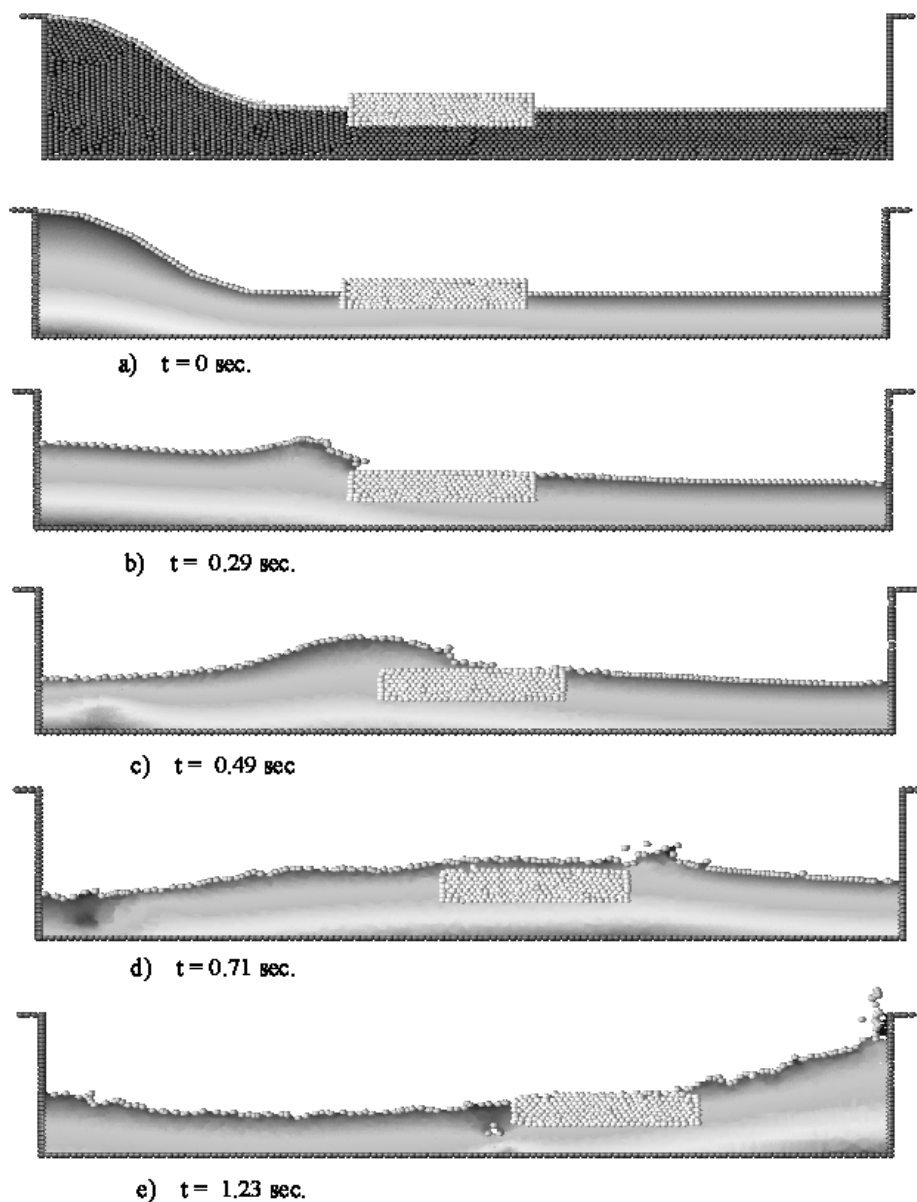


Fig. 19. Floating wood piece hit by a wave

proach. The section of the ship analyzed corresponds now to that of a real container ship. Differently to the previous case the rigid ship is free to move laterally due to the sea wave forces. The objective of the study was to asses the influence of the stabilizers in the ship roll. The figures show clearly how the PFEM predicts the ship and wave motions in a realistic manner.

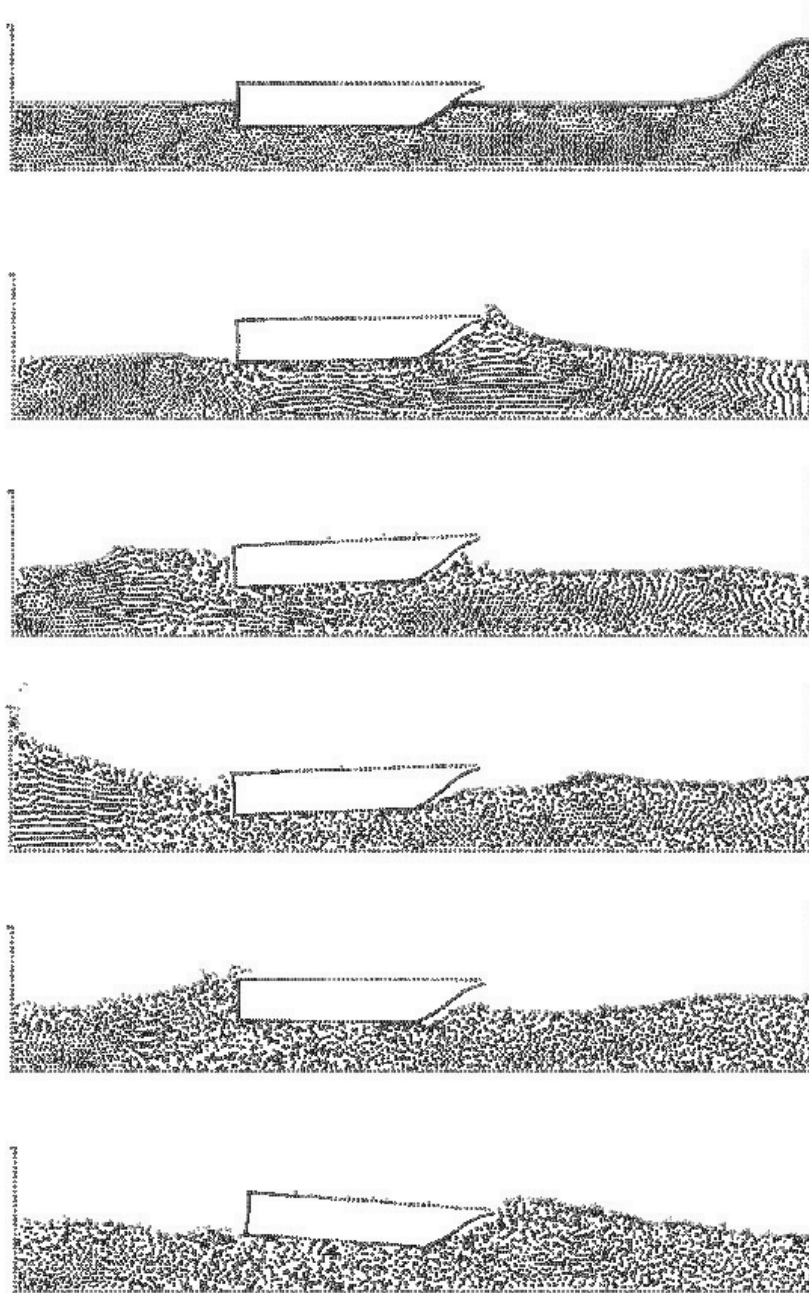


Fig. 20. Motion of a rigid ship hit by an incoming wave. The ship is modelled as a rigid solid restrained to move in the vertical direction.

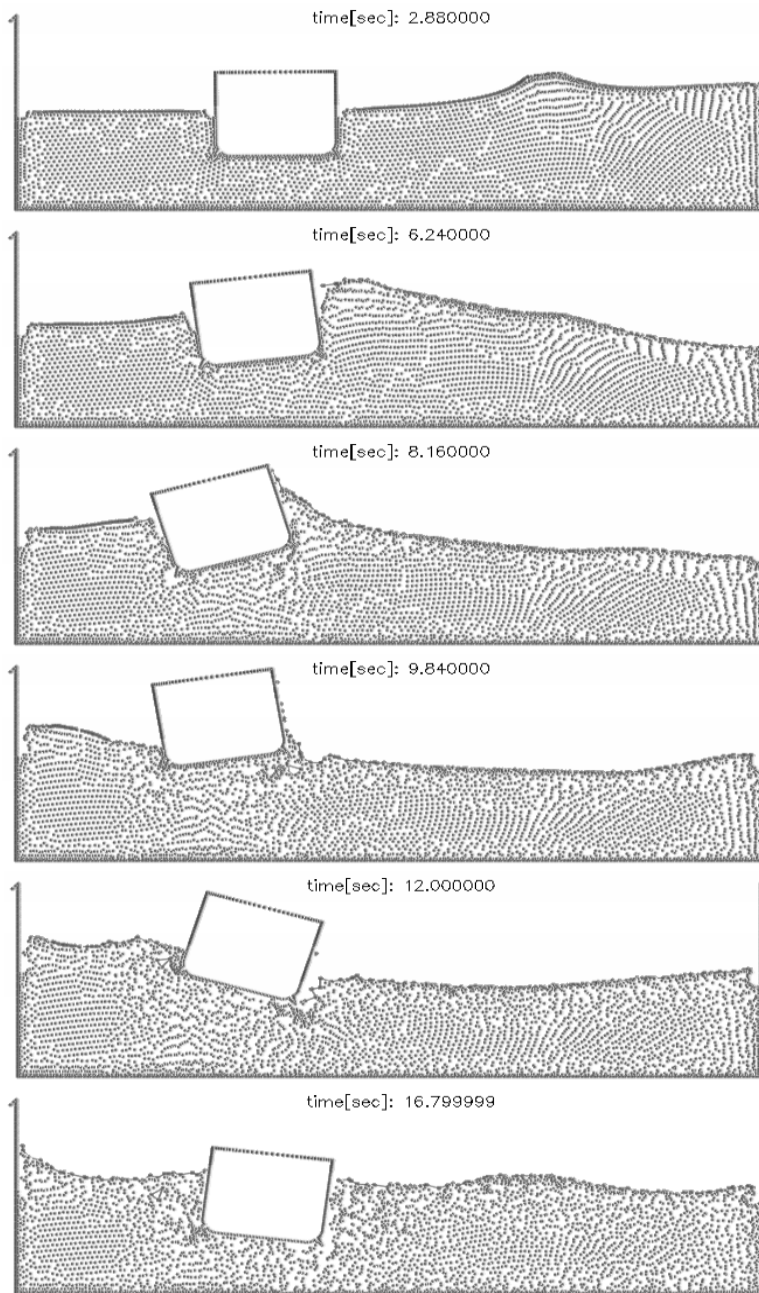


Fig. 21. Ship with stabilizers hit by a lateral wave



**10.9. Tank-ship hit by a lateral wave**

Figure 22 represents a transversal section of a tank-ship and a wave approaching to it. The tank-ship is modelled as a rigid body and it carries a liquid inside which can move freely within the ship domain.

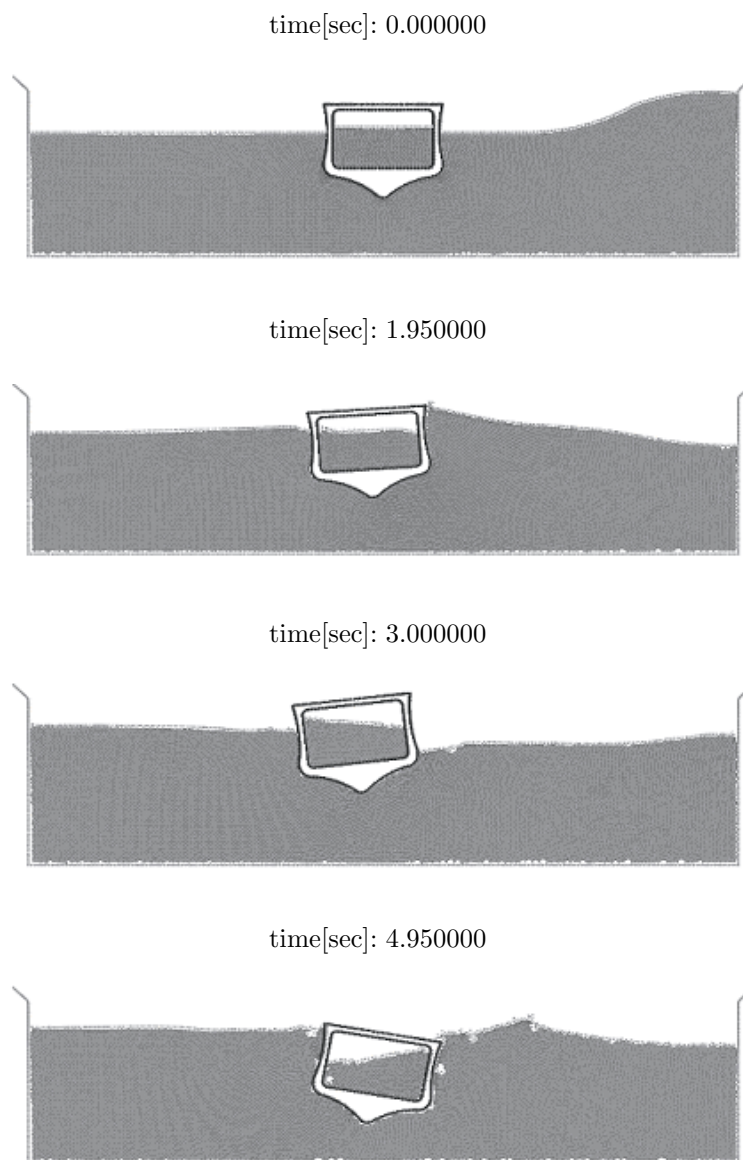


Fig. 22. Tank-ship carrying an internal liquid hit by wave. Ship and fluids motion at different time steps

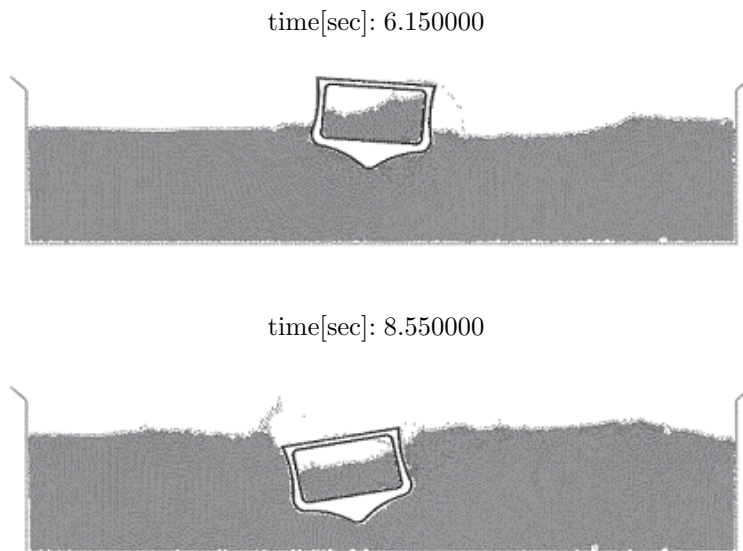
34 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

Fig. 22. Cont.

Initially ( $t = 0.0$ ) the ship is released from a fixed position and the equilibrium configuration found is consistent with Arquimides principle. During the following time steps the external wave hits the ship and both the internal and the external fluids interact with the ship boundaries. At times  $t = 6.15$  and  $8.55$  breaking waves and some water drops slipping along the ship deck can be observed. Figure 23 shows the pressure pattern at two time steps.

#### 10.10. *Rigid cube falling in a recipient with water*

In the next example a solid cube is initially free and falls down within a water recipient. In this example, the rigid solid is modeled first as a fictitious fluid with a higher viscosity, similarly as for the floating solid of Section 10.7. The results of this analysis are shown in Figure 24. Note that the method reproduces very well the interaction of the cube with the free surface as well as the overall sinking process. A small deformation of the cube is produced. This can be reduced by increasing further the fictitious viscosity of the cube particles.

The same problem is analyzed again considering now the cube as a rigid solid subjected to pressure and viscous forces acting in its boundaries. The resultant of the fluid forces and the weight of the cube are applied to the center of the cube. These forces govern the displacement of the cube which is computed by solving the dynamic equations of motion as described in the fractional step algorithm of Section 5, similarly as for the rigid ships of the three previous examples. Here again the moving cube contours define a boundary condition for the fluid particles at each time step.

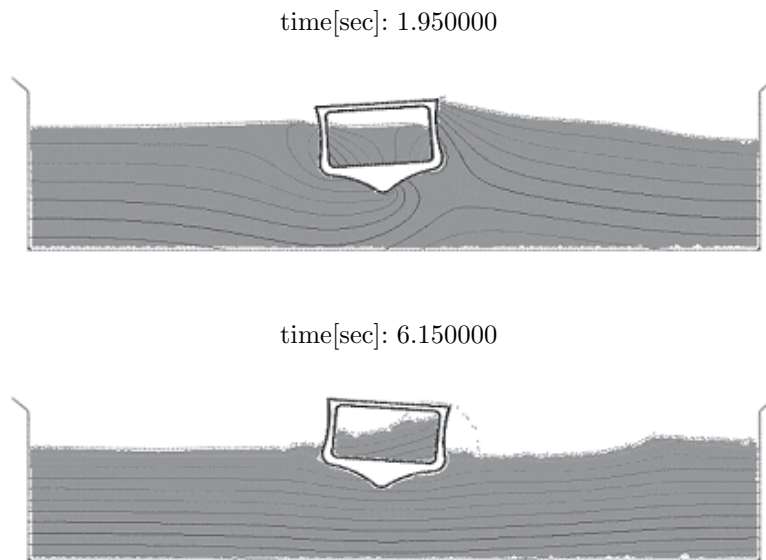


Fig. 23. Tank ship under lateral wave. Pressure distribution at two time steps.

Initially the solid falls down freely due to the gravity forces (Figure 25). Once in contact with the water surface, the Alpha-Shape method recognizes the different boundary contours which are shown with a thick line in the figure. The pressure and viscous forces are evaluated in all the domain and in particular on the cube contours. The fluid forces introduce a negative acceleration in the vertical motion until, once the cube is completely inside the water, the vertical velocity becomes zero. Then, Arquimides forces bring the cube up to the free-surface. It is interesting to observe that there is a rotation of the cube. The reason is that the center of the floating forces is higher in the rotated position than in the initial ones.

Figure 26 shows a repetition of the same problem showing now all the finite elements in the mesh discretizing the fluid. We recall that in all the problems here described the mesh in the fluid domain *is regenerated at each time step* combining linear triangles and quadrilateral elements as described in Section 7. Note that some fluid particles separate from the fluid domain. These particles are treated as free boundary points with zero pressure and hence fall down due to gravity.

It is interesting to see that the final position of the cube is different from that of Figure 25. This is due to the unstable character of the cube motion. A small difference in the numerical computations (for instance in the mesh generation process) shifts the movement of the cube towards the right or the left. Note that a final rotated equilibrium position is found in both cases.

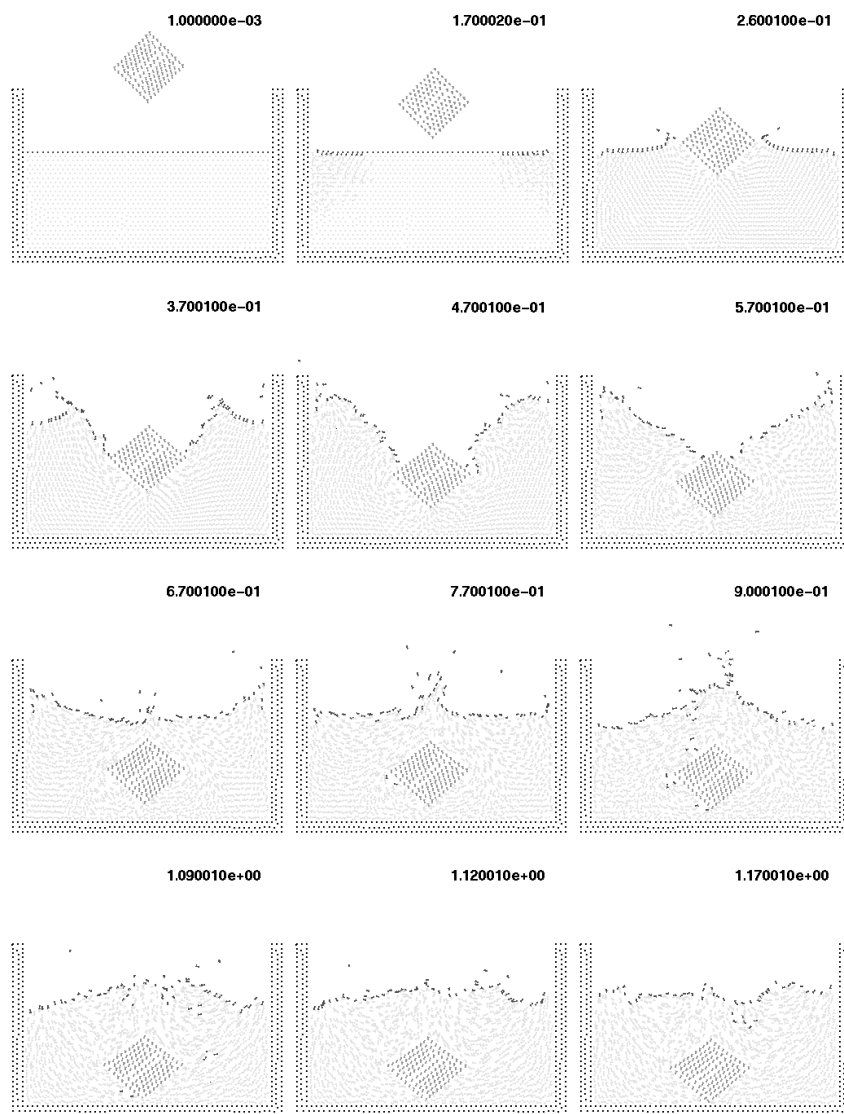
36 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

Fig. 24. Solid cube falling into a recipient with water. The cube is modelled as a very viscous fluid.

### 10.11. *The Rayleigh-Bénard Instability*

This example shows that the PFEM can also be successfully used to solve fluid flow problems traditionally analyzed with Eulerian formulations. The problem solved is that of a heated thin cavity containing a fluid. The flow pattern yields the so called Rayleigh-Bénard hydrodynamical instability giving a roll pattern along the cavity. In this case the Lagrangian fluid flow equations are solved together with the

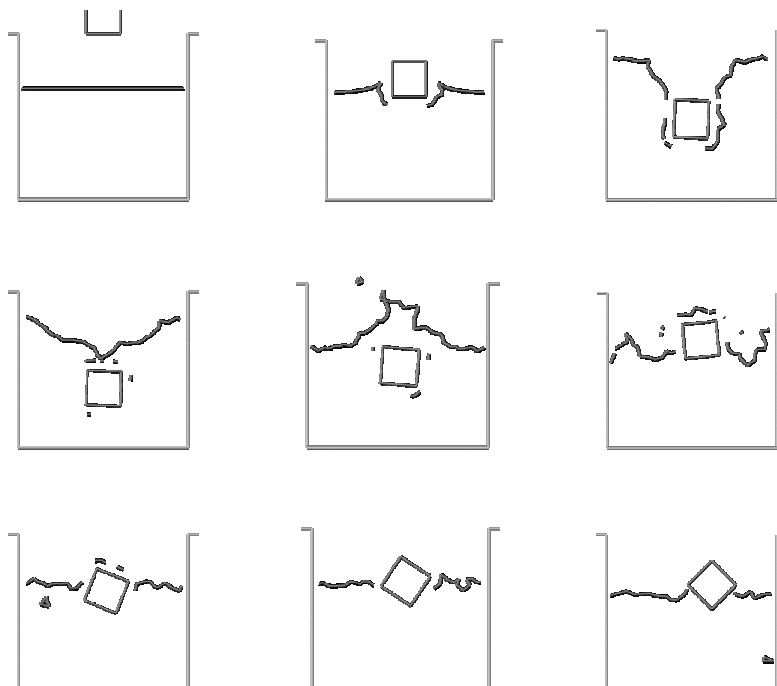


Fig. 25. Cube falling into a recipient with water. The cube is modelled as a rigid solid. Motion of the cube and free surface positions at different time steps.

heat transfer equation also written in a Lagrangian manner. As mentioned at the introduction of Section 10 a value of  $\theta_1 = 0$  has been taken in this example. Details of the solution scheme using a Boussinesq approximations for the coupling between the heat transfer equation and the flow equations are given in Aubry *et al.* (2004).

The bottom and upper part are isothermal with a temperature of  $21^\circ\text{C}$  for the bottom and  $19^\circ\text{C}$  for the top. The initial and reference temperature in the fluid is  $20^\circ\text{C}$  and the side parts are adiabatic. The Rayleigh number is  $10^5$  and the Prandtl number is  $10^{-1}$ . The mesh has 35500 nodes and 69700 elements at the beginning of the analysis. The numerical computations start with the fluid at rest as the initial conditions. For rigid-rigid boundary conditions, the critical value of the Rayleigh number is 1708 so that the flow is here supercritical. However, a quasi-steady state is reached, with periodic oscillations of the temperature and the cells. Figure 27 shows results of the temperature and velocity field showing the development of rolls. Numerical results have been plotted using the GiD pre/postprocessing system developed at CIMNE [Gid (2004)]. More details on the application of the PFEM to this problem can be found in Aubry *et al.* (2004).

38 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

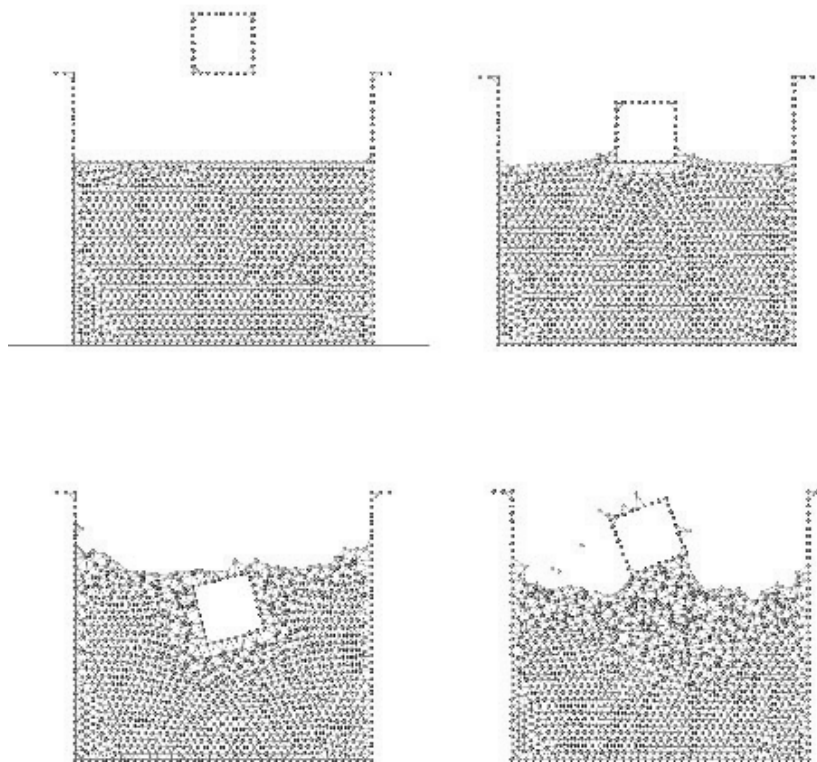


Fig. 26. Cube falling in a water recipient. The cube is modeled as a rigid solid. The finite element meshes generated at the selected instants are shown.

## 11. CONCLUSIONS

The particle finite element method (PFEM) seems ideal to treat problems involving fluids with free surface and submerged or floating structures within a unified Lagrangian finite element framework. Problems such as the analysis of fluid-structure interactions, large motion of fluid or solid particles, surface waves, water splashing, separation of water drops, etc. can be easily solved with the PFEM. The success of the method lays in the accurate and efficient solution of the equations of an incompressible fluid and of solid dynamics using a stabilized finite element method via a fractional step scheme allowing the use of low order elements with equal order interpolation for all the variables. Other essential solution ingredients are the efficient regeneration of the finite element mesh using an extended Delaunay tessellation, the meshless finite element interpolation (MFEM) and the identification of the boundary nodes using an Alpha Shape type technique. The examples presented have shown the potential of the PFEM for solving a wide class of practical FSI problems.

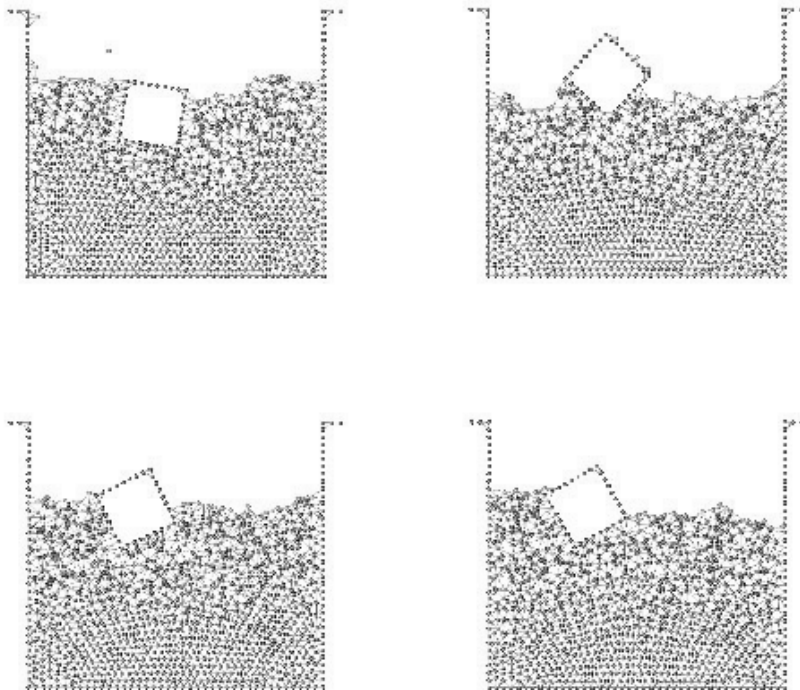


Fig. 26. Cont.

### Acknowledgements

Thanks are given to Mr. Nestor Calvo for this help in the development of the mesh generation process using the extended Delaunay technique.

### Appendix

From Eq.(7) it can be obtained (taking into account Eqs.(5))

$$\frac{\partial r_d}{\partial x_i} = -\frac{1}{a_i} \left[ \bar{r}_{m_i} - \frac{h_j}{2} \frac{\partial r_{m_i}}{\partial x_j} \right] - \frac{\rho u_i h_k}{2a_i} \frac{\partial r_d}{\partial x_k} \quad i, j = 1, n_d \quad , \quad k \neq i \quad (A.1)$$

where

$$a_i = \frac{2\mu}{3} \quad (A.2a)$$

and

$$\bar{r}_{m_i} = \rho \frac{\partial u_i}{\partial t} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j} (2\mu \dot{\epsilon}_{ij}) - b_i \quad (A.2b)$$

Substituting Eq.(A.1) into Eq.(8) and retaining the terms involving the derivatives of  $r_{m_i}$  with respect to  $x_i$  only, leads to the following expression for the stabilized mass balance equation

40 *E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry*

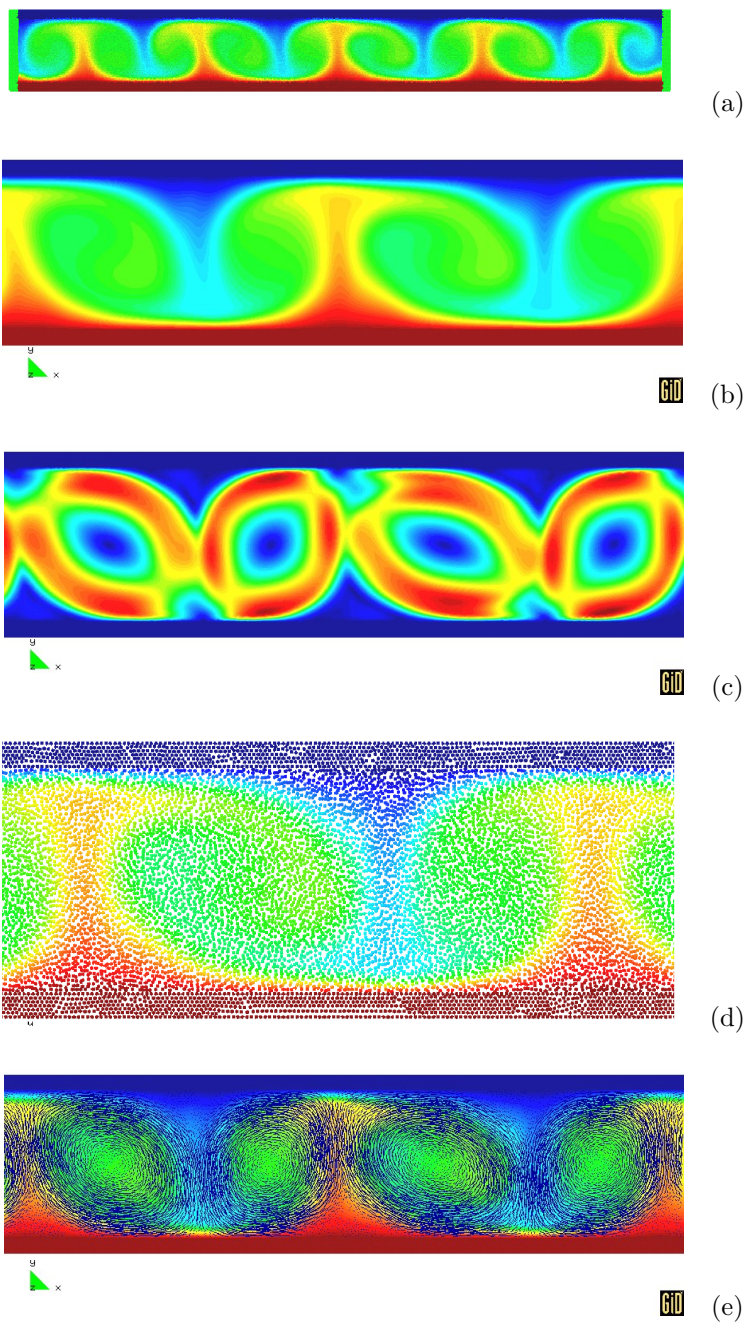


Fig. 27. Rayleigh-Bénard instability with  $Ra = 10^5$  and  $Pr = 10^{-1}$ . (a) Temperature field. (b) Detail of temperature field. (c) Velocity norm field. (d) Detail of velocity norm field plotted on each particle. (e) Velocity vectors on temperature field.



$$r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} = 0 \quad (A.3)$$

with

$$\tau_i = \frac{3h_i^2}{8\mu} \quad (A.4)$$

## References

- Aubry, R., Idelsohn, S.R. and Oñate, E. (2004). Particle finite element method in fluid mechanics including thermal convection-diffusion. *Computer & Structures*, submitted.
- Chorin, A.J. (1967). A numerical solution for solving incompressible viscous flow problems. *J. Comp. Phys.*, **2**: 12–26.
- Codina, R. (2002). Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput. Methods Appl. Mech. Engrg.*, **191**: 4295–4321.
- Codina, R., Vazquez, M. and Zienkiewicz, O.C. (1998). A general algorithm for compressible and incompressible flow - Part III. The semi-implicit form. *Int. J. Num. Meth. in Fluids*, **27**: 13–32.
- Codina, R. and Blasco, J. (2000). Stabilized finite element method for the transient Navier-Stokes equations based on a pressure gradient operator. *Comput. Methods in Appl. Mech. Engrg.*, **182**: 277–301.
- Codina, R. and Zienkiewicz, O.C. (2002). CBS versus GLS stabilization of the incompressible Navier-Stokes equations and the role of the time step as stabilization parameter. *Communications in Numerical Methods in Engineering*, **18** (2): 99–112.
- Cruchaga, M.A. and Oñate, E. (1997). A finite element formulation for incompressible flow problems using a generalized streamline operator. *Comput. Methods in Appl. Mech. Engrg.*, **143**: 49–67.
- Cruchaga, M.A. and Oñate, E. (1999). A generalized streamline finite element approach for the analysis of incompressible flow problems including moving surfaces. *Comput. Methods in Appl. Mech. Engrg.*, **173**: 241–255.
- Donea, J. and Huerta, A. (2003). *Finite element method for flow problems*. J. Wiley.
- Edelsbrunner, H. and Mücke, E.P. (1999). Three dimensional alpha shapes. *ACM Trans. Graphics*, **13**: 43–72.
- Franca, L.P. and Frey, S.L. (1992). Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Comput. Method Appl. Mech. Engrg.*, **99**: 209–233.
- García, J. and Oñate, E. (2003). An unstructured finite element solver for ship hydrodynamic problems. in *J. Appl. Mech.*, **70**: 18–26, January.

- George. (1991). *Automatic mesh generation. Application to the finite element method*, J. Wiley.
- GiD. (2004). The personal pre/postprocessor. CIMNE, Barcelona, [www.gidhome.com](http://www.gidhome.com).
- Hansbo, P. and Szepessy, A. (1990). A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, **84**: 175–192.
- Hughes, T.J.R., Franca, L.P. and Balestra, M. (1986). A new finite element formulation for computational fluid dynamics. V Circumventing the Babuska-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal order interpolations. *Comput. Methods Appl. Mech. Engrg.*, **59**: 85–89.
- Hughes, T.J.R., Franca, L.P. and Hulbert, G.M. (1989). A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Comput. Methods Appl. Mech. Engrg.*, **73**: 173–189.
- Hughes, T.J.R., Hauke, G. and Jansen, K. (1994). Stabilized finite element methods in fluids: Inspirations, origins, status and recent developments. in: *Recent Developments in Finite Element Analysis*. A Book Dedicated to Robert L. Taylor, T.J.R. Hughes, E. Oñate and O.C. Zienkiewicz (Eds.), CIMNE, Barcelona, Spain, pp. 272–292.
- Idelsohn, S.R., Oñate, E., Del Pin F. and Calvo, N. (2002). Lagrangian formulation: the only way to solve some free-surface fluid mechanics problems. *Fifth World Congress on Computational Mechanics*, Mang HA, Rammerstorfer FG and Eberhardsteiner J. (eds), July 7–12, Viena, Austria.
- Idelsohn, S.R., Oñate, E., Calvo, N. and del Pin, F. (2003a). The meshless finite element method. *Int. J. Num. Meth. Engng.*, **58,6**: 893–912.
- Idelsohn, S.R., Oñate, E. and Del Pin, F. (2003b). A lagrangian meshless finite element method applied to fluid-structure interaction problems. in *Computer and Structures*, **81**: 655–671.
- Idelsohn, S.R., Calvo, N. and Oñate, E. (2003c). Polyhedrization of an arbitrary point set. *Comput. Method Appl. Mech. Engng.*, **192** (22-24): 2649–2668.
- Idelsohn, S.R., Oñate, E. and Del Pin, F. (2004). The particle finite element method a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int. J. Num. Meth. Engng.*, submitted.
- Irons, B.M. (1970). A frontal solution program. *Int. J. Num. Meth. Engng.*, **2**: 5–32.
- Koshizuka, S. and Oka, Y. (1996). Moving particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Engng. Science*, **123**: 421–434.
- Laitone, E.V. (1960). The second approximation to cnoidal waves. *J. Fluids Mech.*, **9**: 430.
- Oñate, E. (1998). Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. *Comput. Meth. Appl. Mech. Engng.*, **151**: 233–267.

- Oñate, E. (2000). A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. *Comp. Meth. Appl. Mech. Engrg.*, **182** (1–2): 355–370.
- Oñate, E. (2002). Possibilities of finite calculus in computational mechanics. Submitted to *Int. J. Num. Meth. Engrg.*
- Oñate, E. and Idelsohn, S.R. (1998). A mesh free finite point method for advective-diffusive transport and fluid flow problems. *Computational Mechanics*, **21**: 283–292.
- Oñate, E., Sacco, C. and Idelsohn, S.R. (2000). A finite point method for incompressible flow problems. *Comput. and Visual. in Science*, **2**: 67–75.
- Oñate, E. and García, J. (2001). A finite element method for fluid-structure interaction with surface waves using a finite calculus formulation. *Comput. Meth. Appl. Mech. Engrg.*, **191**: 635–660.
- Oñate, E., Idelsohn, S.R. and Del Pin, F. (2003). Lagrangian formulation for incompressible fluids using finite calculus and the finite element method. in *Numerical Methods for Scientific Computing Variational Problems and Applications*, Y. Kuznetsov, P. Neittanmaki and O. Pironneau (Eds.), CIMNE, Barcelona.
- Oñate, E., García, J. and Idelsohn, S.R. (2004). Ship hydrodynamics. In *Encyclopedia of Computational Mechanics*, E. Stein, R. de Borst and T.J.R. Hughes (Eds), J. Wiley.
- Radovitzki, R. and Ortiz, M. (1998). Lagrangian finite element analysis of a Newtonian flows. *Int. J. Num. Meth. Engrg.*, **43**: 607–619.
- Sheng, C., Taylor, L.K. and Whitfield, D.L. (1996). Implicit lower-upper/approximate-factorization schemes for incompressible flows” *Journal of Computational Physics*, **128** (1), 32–42, 1996.
- Storti, M., Nigro, N. and Idelsohn, S.R. (1995). Steady state incompressible flows using explicit schemes with an optimal local preconditioning. *Computer Methods in Applied Mechanics and Engineering*, **124**: 231–252.
- Tezduyar, T.E., Mittal, S., Ray, S.E. and Shih, R. (1992). Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity–pressure elements. *Comput. Methods Appl. Mech. Engrg.*, **95**: 221–242.
- Thompson, J.F., Soni, B.K. and Weatherill, N.P. (Eds.) (1999). *Handbook of Grid Generation*, CRC Press.
- Zienkiewicz, O.C. and Taylor, R.L. (2000). *The finite element method*. 5th Edition, 3 Volumes, Butterworth–Heinemann.