Fluid-Structure Interaction Using the Particle Finite Element Method

S.R. Idelsohn[1,2], E. Oñate[2], F. Del Pin[1]   and   Nestor Calvo[1]

[1] International Center for Computational Methods in Engineering (CIMEC)
Universidad Nacional del Litoral and CONICET, Santa Fe, Argentina
e–mail: sergio@ceride.gov.ar
[2] International Center for Numerical Methods in Engineering (CIMNE)
Universidad Politécnica de Cataluña, Barcelona, Spain
e–mail: onate@cimne.upc.es

Abstract

In the present work a new approach to solve fluid-structure interaction problems is described. Both, the equations of motion for fluids and for solids have been approximated using a material (Lagrangian) formulation. To approximate the partial differential equations representing the fluid motion, the shape functions introduced by the Meshless Finite Element Method (MFEM) have been used. Thus, the continuum is discretized into particles that move under body forces (gravity) and surface forces (due to the interaction with neighboring particles). All the physical properties such as density, viscosity, conductivity, etc., as well as the variables that define the temporal state such as velocity and position and also other variables like temperature are assigned to the particles and are transported with the particle motion. The so called Particle Finite Element Method (PFEM) provides a very advantageous and efficient way for solving contact and free-surface problems, highly simplifying the treatment of fluid-structure interactions.

Key words: Fluid-Structure interaction, Particle methods, Lagrange formulations, Incompressible Fluid Flows, Meshless Methods, Finite Element Method.

1. Introduction

Many classifications have been proposed to enclose the numerical formulations that approximate the continuum equations that govern incompressible fluid flows. In particular the one describing the way that convection is treated divides the numerical formulations into two classes, namely *material (or lagrangian) formulations* and *spatial (or eulerian) formulations*. The first one describes convection by placing a set of axes over the material particles that move accordingly to the equations of motion. In the eulerian case the axes are set fixed in space and convection terms are included in the equations describing the transport of the fluid flow. The present work will describe a method that uses a material formulation. The equations of motion for both, the solid and fluid do not present convection terms, implying that the convection effect is directly obtained by moving the discrete domain.

Many authors have taken advantage of Lagrangian formulations to describe different types of problems. The *Smooth Particle Hydrodynamics* (SPH) method developed by Monaghan (1977) [mon81, mon97] should be mentioned as a pioneer method of this kind.

Many other methods have been derived from SPH. One that has shown remarkable results is the *Moving Particle Semi-Implicit* method (MPS) introduced by Koshizuka and Oka (1996) [kos96]. These methods use a kernel function to interpolate the unknowns. SPH uses a weak formulation while MPS uses a strong form of the governing equations.

Ramaswamy (1986) [ram87] proposed a Lagrangian finite element formulation for a 2-D incompressible fluid flow. In that paper the mesh was convected according to the equations of motion but without change of topology, making it rather limiting when the elements got highly distorted. The equations of motion were discretized in space by using the finite element method with linear shape functions.

Another possible classification for numerical formulations may be the one that separates the methods that make use of a *standard finite element mesh* (like those made of tetrahedra or hexahedra), and the methods that do not need a standard mesh, namely the *meshless methods*. The formulation described in this paper can be considered a particular class of meshless method. Again, SPH might be cited as one the first meshless methods.

Indeed, after Monaghans work and in particular in the past 20 years, many have been the attempts to develop a robust meshless method that could approximate PDE's in 2-D and 3-D with acceptable accuracy, convergence and speed. Among others, the methods based on *Moving Least Square* interpolations [nay92, bel94], *Partition of Unity* [dua95], and the ones based on the *natural neighbor* interpolation functions [bra95, suk98] may be listed.

In this work the interpolation function used by the Meshless Finite Element Method (MFEM) [ide03a] will be implemented. This function uses the Voronoï diagram of the cloud of points to construct the interpolant. The *extended Delaunay tessellation* (EDT) [ide03b] is applied to connect the neighboring particles. The EDT provides polyhedral elements that are sliver-free in 3-D, avoiding instabilities of the Delaunay tessellation due to distorted tetrahedra. The MFEM shape functions adapt automatically to the polyhedra and in the case that the polyhedron is a simple, the shape function behaves exactly as the linear finite element shape function.

Fluid-structure interaction (FSI) problems have been of special interest for designers and engineers in the past 20 years. This explains why more robust and stable formulations have been developed to assist the approximation of contact problems. Embedded methods have been developed by Löhner et al. [loh03] where a single mesh is used to partition the fluid as well as the structure. Also Arbitrary Lagrangian-Eulerian (ALE) formulations [Sou00] have given acceptable results when the displacements or the geometry deformations are not excessively large.

The approximation for the FSI problem depends basically on the coupling of the fluid and structure equations. Based on this coupling FSI problems may be divided into problems with weak interaction and problems with strong interaction. The later are found when elastic deformation of the solid takes place. The weak interpolation case happens when large rigid displacements are present. This situation is typical in ship hydrodynamics, when a rigid body moves according to the forces given by the pressure field obtained from the fluid dynamic problem. These forces applied to the rigid body will accelerate it, changing its velocity and therefore, its position.

FSI problems have been classically solved in a partitioned manner solving iteratively the discretized equations for the flow and the solid domain separately. The solution of both, fluid flow and solid, with the same material formulation, open the door to solve the global coupled problem in a monolithic fashion. Nevertheless, in this paper the rigid solid will still be solved separately from the fluid. A partitioned method [pip95, mok01] or iterative method [rug00, rug01, zha01 is chosen to solve the coupling between the fluid and solid. The advantage to use a material formulation for both, solid and fluid parts will be used here only to better reproduce breaking waves or separated drops in the fluid, which are phenomena impossible to reproduce using a spatial formulation.

The layout of the paper is the following: in the next section the basic lagrangian equations of motion for the fluid and solid domains are given. Next the discretization method chosen to solve the incompressible fluid flow equations and the solid dynamics in time equations are detailed. The algorithm for the recognition of the boundary nodes and the treatment of the free-surface in the fluid is explained. Finally, the efficiency of the Particle Finite Element Method for solving a variety of fluid-structure interaction problems involving large motion of the free-surface in the fluid is shown.

2.Equations of motion

2.1 Fluid dynamic problem: updating the fluid particle positions

The fluid particle positions will be updated via solving the lagrangian form of the Navier-Stokes equations.

Let X_i the initial position of a particle a time $t=t_0$ and let x_i the final position. Been $u_i(x_j,t)=u_i$ the velocity of the particle in the final position the following approximate relation can be written:

$$x_i = X_i + f(u_i,t,Du_i/Dt).$$ (1)

Conservation of momentum and mass for incompressible Newtonian fluids in the lagrangian frame of reference are represented by the Navier-Stokes equations and the continuity equation in the final x_i position, as follows:

Mass conservation:

$$\frac{D\rho}{Dt} + \rho\frac{\partial u_i}{\partial x_i} = 0.$$ (2)

Momentum conservation:

$$\rho \frac{Du_i}{Dt} = -\frac{\partial}{\partial x_i} p + \frac{\partial}{\partial x_j} \tau_{ij} + \rho f_i, \tag{3}$$

where ρ is the density, p the pressure, τ_{ij} the deviatoric stress tensor, f_i the source term (usually the gravity) and $\frac{D}{Dt}$ represents the total or material time derivative.

For Newtonian fluids the stress tensor τ_{ij} may be expressed as a function of the velocity field through the viscosity μ by

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_l}{\partial x_l} \delta_{ij} \right). \tag{4}$$

For near incompressible flows $\frac{\partial u_i}{\partial x_i} << \frac{\partial u_k}{\partial x_l}$ the term:

$$\frac{2\mu}{3} \frac{\partial u_i}{\partial x_i} \approx 0, \tag{5}$$

and it may be neglected from Eq. 4. Then:

$$\tau_{ij} \approx \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \tag{6}$$

In the same way, the term $\frac{\partial}{\partial x_j} \tau_{ij}$ in the momentum equations may be simplified for near incompressible flows as:

$$\frac{\partial}{\partial x_j} \tau_{ij} = \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) = \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_j}{\partial x_i} \right) =$$
$$\mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \mu \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_j} \right) \approx \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right). \tag{7}$$

Using eq. (7), the momentum equations can be finally written as:

$$\rho \frac{Du_i}{Dt} = -\frac{\partial}{\partial x_i} p + \frac{\partial}{\partial x_j} \tau_{ij} + \rho f_i \approx -\frac{\partial}{\partial x_i} p + \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \rho f_i \tag{8}$$

Note: eq.(3) or the equivalent for incompressible fluid flow eq.(8) are non-linear. In Eulerian formulations the non-linearity is explicitly present in the convective terms. In this lagrangian formulation, the non-linearity is due to the fact that eqs. (3) and (8) are written in the final positions of the particles, which are unknown. There are others way to write

lagrangian formulations, for instance staying in the initial position [aub04]. In all cases, the equations are non-linear.

Boundary conditions

On the boundaries, the standard boundary conditions for the Navier-Stokes equations are:

$$\tau_{ij}v_j - pv_i = \overline{\sigma}_{ni} \qquad \text{on } \Gamma_{\sigma}$$

$$u_i v_i = \overline{u}_n \qquad \text{on } \Gamma_n$$

$$u_i \zeta_i = \overline{u}_t \qquad \text{on } \Gamma_t$$

where v_i and ζ_i are the components of the normal and tangent vectors to the boundary.

2.2 Solid dynamics problem: updating the rigid body position

In this paper, the structure will be considered as a rigid solid. Then, the equations of motion for a rigid body are:

$$m\frac{DU_i}{Dt} = F_i \tag{9}$$

where F_i are the resultant of the external forces (surface forces, gravity force, etc.), whose line of action passes through the mass center of the body, U_i is the velocity of the mass center and m the total mass of the solid.

The actual motion of the rigid body consists in the superposition of the translation produced by the resultant force F_i and the rotation produced by the couple T_i satisfying:

$$\frac{DM_i}{Dt} = T_i, \tag{10}$$

where M_i is the angular momentum about the mass center. It must be noted that in (10) the time derivative is expressed as the rate of change with respect to any non-rotating system of axis. It may be also expressed as the derivative with respect to the body fixed axes by:

$$\frac{DM_i}{Dt} \equiv \frac{\overline{DM}_i}{Dt} + \varepsilon_{ijk}e_i\Omega_j M_k = T_i, \tag{11}$$

where Ω denotes the angular velocity of the body, e are orthogonal unit basis vectors, ε the permutation symbol and $\overline{D}/Dt$ is the derivative with respect to the body fixed axes.

Let now the body fixed axes be the principal axes of inertia of the body, with its origin at the center of mass, then:

$$M_i = I_i\Omega_i, \text{ (without summation in the index } i) \tag{12}$$

where I_i are the principal moments of inertia and then:

$$\frac{\overline{DM}_i}{Dt} = I_i\frac{\overline{D\Omega}_i}{Dt}. \tag{13}$$

Finally, the equations of motion of the body might be summarized as:

$$m \frac{DU_i}{Dt} = F_i, \qquad (14)$$

$$I_i \frac{\overline{D\Omega_i}}{Dt} + \varepsilon_{ijk} e_i \Omega_j (I_k \Omega_k) = T_i, \qquad (15)$$

Calling a_i and α_i the linear and the angular acceleration of the mass center of the body:

$$m a_i = F_i, \qquad (16)$$

$$I_i \alpha_i + \varepsilon_{ijk} e_i \Omega_j (I_k \Omega_k) = T_i, \qquad (17)$$

This is a non-linear system of partial differential equations that has to be linearized for its numerical approximation.

The final rigid body velocity of an arbitrary point is a combination of both, the linear velocity of the center of mass U_i and the angular velocity Ω_i according to:

$$u_i = U_i + \varepsilon_{ijk} e_i \Omega_j r_k. \qquad (18)$$

where r_i is the distance from the origin of the body axes to an arbitrary point attached to the body. The velocity u_i will be used later as a boundary condition for the fluid dynamics problem.

A very large number of problems involve plane motion. In this case, equation (15) reduces to:

$$I \frac{\overline{D\Omega}}{Dt} \equiv I \alpha = T, \qquad (19)$$

where Ω, I, α and T are the planar angular velocity, the moment of inertia, the planar angular acceleration and the external couple respectively.

3. The discrete fluid dynamics problem

The Navier-Stokes equations present three main difficulties:

- The equations are time dependent and thus a temporal integration needs to be carried out.
- A spatial dependency is also present and thus the space will be discretized.
- Finally, Eq. 3 presents a non-linearity, which must be solved iteratively.

Each of the above items will be explained and a solution algorithm will be introduced to obtain a final accurate and robust numerical scheme.

3.1 Implicit-explicit time integration

Let t^n and t^{n+1} be the initial and final time step. Let $\Delta t = t^{n+1} - t^n$ be the time increment. Eq. (8) is integrated implicitly in time as:

$$\rho \frac{Du_i}{Dt} \approx \rho \frac{u_i(x_i, t^{n+1}) - u_i(X_i, t^n)}{\Delta t} = \rho \frac{u_i^{n+1} - u_i^n}{\Delta t} = \left[-\frac{\partial}{\partial x_i} p + \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} \right) + \rho f_i \right]^{n+\theta}, \quad (20)$$

where $[\phi(x,t)]^{n+\theta}$ means $\theta\phi(x,t^{n+1}) + (1-\theta)\phi(x,t^n) = \theta\phi^{n+1} + (1-\theta)\hat{\phi}^n$

and $\hat{\phi}^n = \phi(x,t^n)$ represents the value of the function at time t^n but at the final position x. For simplicity ϕ^n will be used instead of $\hat{\phi}^n$.

Only the case of $\theta = 1$ (fully implicit scheme) will be considered next. Other values, as for instance $\theta = 1/2$, may be considered without major changes.
The time integrated equations become:

$$\rho\frac{u_i^{n+1} - u_i^n}{\Delta t} = \left[-\frac{\partial}{\partial x_i}p\right]^{n+1} + \left[\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) + \rho f_i\right]^{n+1}.$$ (21)

The mass conservation is also integrated implicitly by:

$$\frac{D\rho}{Dt} \approx \frac{\rho^{n+1} - \rho^n}{\Delta t} = -\rho^{n+1}\frac{\partial u_i^{n+1}}{\partial x_i}.$$ (22)

The time integration of Eq. (20) presents some difficulties: it is a fully coupled equation involving four degrees of freedom by node. When the fluid is incompressible or nearly incompressible advantages can be taken from the fact that in Eq. (20) the three components of the velocity are only coupled via the pressure. The fractional-step method proposed in [cod01] will be used for the time solution. This basically consists in splitting each time step in two pseudo-time steps. In the first step the implicit part of the pressure is avoided in order to have a decoupled equation in each of the velocity components. The implicit part of the pressure is added at a second step. The fractional-step algorithm for eqs. (21) and (22) is the following:

Split of the momentum equations

$$\frac{Du_i}{Dt} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_i^{n+1} - u_i^* + u_i^* - u_i^n}{\Delta t} = -\frac{1}{\rho}\frac{\partial}{\partial x_i}p^{n+1} + \frac{1}{\rho}\frac{\partial \tau_{ij}^{n+\theta}}{\partial x_j} + f_i,$$ (23)

where u_i^* are fictitious variables termed fractional velocities defined by the split:

$$u_i^* = u_i^n + f_i\,\Delta t - \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}\gamma p^n + \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}\tau_{ij}^{n+\theta},$$ (24)

$$u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}(p^{n+1} - \gamma p^n),$$ (25)

where $p^n = p(x,t^n)$ is the value of the pressure at time t^n evaluated at the final position and f_i is considered constant in time.

In Eqs. (24) and (25) γ is a parameter giving the amount of pressure splitting, varying between 0 and 1. A larger value of γ means a small pressure split. In this paper γ will be fixed to 0 in order to have the larger pressure split and hence, a better pressure stabilization. Other values as, for instance $\gamma = 1$, may be used to derive high order schemes in time [cod01].

Taking into account Eq. (7), the last term in Eq. (24) may be written as:

$$\frac{\partial}{\partial x_j}\tau_{ij}^{n+\theta} = \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{n+\theta}}{\partial x_j}\right) = \mu(1-\theta)\frac{\partial}{\partial x_j}\left(\frac{\partial \hat{u}_i^{n}}{\partial x_j}\right) + \mu\theta\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{n+1}}{\partial x_j}\right). \tag{26}$$

The following approximations have been introduced [cod01]:

$$\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{n+\theta}}{\partial x_j}\right) \approx \mu(1-\theta)\frac{\partial}{\partial x_j}\left(\frac{\partial \hat{u}_i^{n}}{\partial x_j}\right) + \mu\theta\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{*}}{\partial x_j}\right). \tag{27}$$

This allows to write Eq. (24) as:

$$u_i^{*} = u_i^{n} + f_i\,\Delta t - \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}\gamma\hat{p}^{n} + \frac{\Delta t}{\rho}\mu(1-\theta)\frac{\partial}{\partial x_j}\left(\frac{\partial \hat{u}_i^{n}}{\partial x_j}\right) + \frac{\Delta t}{\rho}\mu\theta\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{*}}{\partial x_j}\right). \tag{28}$$

For $\gamma = 1$ and $\theta = 1$ the equations for the fractional velocities becomes:

$$u_i^{*} - \frac{\Delta t}{\rho}\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{*}}{\partial x_j}\right) = u_i^{n} + f_i\,\Delta t. \tag{29}$$

Split of the mass conservation equations

$$\frac{D\rho}{Dt} \approx \frac{\rho^{n+1}-\rho^{n}}{\Delta t} = \frac{\rho^{n+1}-\rho^{*}+\rho^{*}-\rho^{n}}{\Delta t} = -\rho\frac{\partial(u_i^{n+1}-u_i^{*}+u_i^{*})}{\partial x_i}, \tag{30}$$

where ρ^{*} is a fictitious variable defined by the split

$$\frac{\rho^{*}-\rho^{n}}{\Delta t} = -\rho\frac{\partial u_i^{*}}{\partial x_i}, \tag{31}$$

$$\frac{\rho^{n+1}-\rho^{*}}{\Delta t} = -\rho\frac{\partial(u_i^{n+1}-u_i^{*})}{\partial x_i}. \tag{32}$$

Coupled equations

From eqs. (25), (31) and (32) the coupled mass-momentum equation becomes:

$$\frac{\rho^{n+1}-\rho^{*}}{\Delta t^{2}} = \frac{\partial^{2}}{\partial x_i^{2}}(p^{n+1}). \tag{33}$$

Taking into account Eq. (31) the above expression can be written as:

$$\frac{\rho^{n+1}-\rho^{n}}{\Delta t^{2}} + \frac{\rho}{\Delta t}\frac{\partial u_i^{*}}{\partial x_i} = \frac{\partial^{2}}{\partial x_i^{2}}(p^{n+1}). \tag{34}$$

It is important to note that in eq. (34) the incompressibility condition has not be introduced yet. The simplest way to introduce the incompressibility condition in a lagrangian formulation is to write:

$$\rho^{n+1} = \rho^{n} = \rho^{0} = \rho. \tag{35}$$

Then, the first term of Eq. (34) disappears, giving:

$$\frac{\rho}{\Delta t}\frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2}(p^{n+1}) \quad . \tag{36}$$

The three steps of the fractional method used here can be summarized by:

$$u_i^* - \frac{\Delta t}{\rho}\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^*}{\partial x_j}\right) = u_i^{\,n} + f_i\,\Delta t \qquad \Rightarrow u_i^*, \tag{37}$$

$$\frac{\rho}{\Delta t}\frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2}(p^{n+1}) \qquad \Rightarrow p^{n+1}, \tag{38}$$

$$u_i^{\,n+1} = u_i^* - \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}(p^{n+1}) \qquad \Rightarrow u_i^{\,n+1}. \tag{39}$$

3.2 The spatial discretization provided by the MFEM

One of the key to solve a fluid mechanics problem using a lagrangian formulation is to generate efficiently the shape functions to approximate the spatial unknown. In the Finite Element context, this means to generate permanently, at each time step, a new mesh. In this work the interpolation function used by the Meshless Finite Element Method (MFEM) [ide03a] will be implemented. This function uses the Voronoï diagram of the cloud of points to construct the interpolant. The extended Delaunay tessellation (EDT) [ide03b] is applied to connect the neighboring particles. The EDT provides polyhedral elements that are sliver-free in 3-D, avoiding instabilities of the Delaunay tessellation due to distorted tetrahedral. EDT provide a way to generate meshes at each time step very efficiently in a computing time which is largely smaller than the computing time needed to solve the linearized system of equation. EDT together with the MFEM are the main key to make the PFEM presented in this paper a useful tool.

The unknown functions are approximated using an equal order interpolation for all variables in the final configuration:

$$u_i = \sum_l N_l(X,t)U_{il}, \tag{40}$$

$$p = \sum_l N_l(X,t)P_l. \tag{41}$$

In matrix form:

$$u_i = \mathbf{N}^T(X,t)\mathbf{U}_i \tag{42}$$

$$p = \mathbf{N}^T(X,t)\mathbf{P}, \tag{43}$$

or in compact form:

$$u_i = \mathbf{N}_i^T\mathbf{U} = \begin{bmatrix} \mathbf{N}^T & & \\ & \mathbf{N}^T & \\ & & \mathbf{N}^T \end{bmatrix}\mathbf{U}, \tag{44}$$

where $\mathbf{N}^T$ are the MFEM shape functions and $\mathbf{U}, \mathbf{P}$ the nodal values of the three components of the unknown velocity and the pressure respectively.

It must be noted that the shape functions $\mathbf{N}(X,t)$ are functions of the particle coordinates only. Then, the shape functions may change in time following the particle positions.

During the time step, a mesh update may introduce a change in the shape function definition, which must be taken into account. During the time integration there are two times involved: t^n and t^{n+1}. The following notation will be used to distinguish between $\mathbf{N}(X,t^n)$ and $\mathbf{N}(X,t^{n+1})$:

$$\mathbf{N}(X,t^n) = \mathbf{N}^n \quad \text{and} \quad \mathbf{N}(X,t^{n+1}) = \mathbf{N}^{n+1}. \tag{45}$$

In this work, the following hypothesis will be introduced: There is not mesh update during each time step. This means that if a mesh update is introduced at the beginning of a time step, the same mesh (but deformed) will be kept until the end of the time step.

Mathematically this means:

$$\mathbf{N}(X,t^n) = \mathbf{N}(X,t^{n+1}). \tag{46}$$

Unfortunately, this hypothesis is not always true and this introduces small errors in the computation, which are neglected in this paper.

Using the Galerkin weighted residual method to solve the splitted equations, the following integrals are obtained:

$$\int_V \mathbf{N}_i u_i^* dV \frac{\rho}{\Delta t} - \int_V \mathbf{N}_i u_i^n dV \frac{\rho}{\Delta t} - \int_V \mathbf{N}_i f_i \, \rho dV + \int_V \mathbf{N}_i \frac{\partial}{\partial x_i} \gamma p^n dV -$$

$$\int_V \mathbf{N}_i \mu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{n+\theta}}{\partial x_j}\right) dV - \int_{\Gamma_\sigma} \mathbf{N}_i \left[\bar{\sigma}_{ni} - (\tau_{ij}^{n+\theta} v_j - \gamma p^n v_i)\right] d\Gamma = 0, \tag{47}$$

$$\int_V \mathbf{N}\left[\frac{\rho}{\Delta t}\left(\frac{\partial u_i^*}{\partial x_i}\right) - \frac{\partial^2}{\partial x_i^2}(p^{n+1} - \gamma p^n)\right] dV + \frac{\rho}{\Delta t}\int_{\Gamma_u} \mathbf{N} \, (\bar{u}_i^{n+1} v_i - u_i^{n+1} v_i) d\Gamma = 0, \tag{48}$$

$$\int_V \mathbf{N}_i\left[(u_i^{n+1} - u_i^*)\frac{\rho}{\Delta t} + \frac{\partial}{\partial x_i}(p^{n+1} - \gamma p^n)\right] dV - \int_{\Gamma_\sigma} \mathbf{N}_i(p^{n+1} - \gamma\hat{p}^n) v_i d\Gamma = 0, \tag{49}$$

where the boundary conditions have been also spliced and V is the volume at time t^{n+1}. Integrating by parts some of the terms, the above equations become:

$$\int_V \mathbf{N}_i(u_i^* - f_i \, \Delta t)\frac{\rho}{\Delta t} dV - \int_V \mathbf{N}_i u_i^n \frac{\rho}{\Delta t} dV + \int_V \mathbf{N}_i \frac{\partial}{\partial x_i}\gamma p^n dV +$$

$$+ \mu\int_V \frac{\partial \mathbf{N}_i}{\partial x_j}\frac{\partial u_i^{n+\theta}}{\partial x_j} dV - \int_{\Gamma_\sigma} \mathbf{N}_i(\bar{\sigma}_{ni} + \gamma p^n v_i) d\Gamma = 0, \tag{50}$$

$$-\frac{\rho}{\Delta t}\int_V \frac{\partial \mathbf{N}}{\partial x_i} u_i^* \, dV - \int_V \frac{\partial \mathbf{N}}{\partial x_i} \frac{\partial (p^{n+1} - \gamma p^n)}{\partial x_i})dV + \frac{\rho}{\Delta t}\int_{\Gamma_u} \mathbf{N} \, \bar{u}_n^{\,n+1} d\Gamma = 0, \tag{51}$$

$$\int_V \mathbf{N}_i \{(u_i^{\,n+1} - u_i^*)\frac{\rho}{\Delta t} + \frac{\partial}{\partial x_i}(p^{n+1} - \gamma p^n)\} dV - \int_{\Gamma_\sigma} \mathbf{N}_i (p^{n+1} - \gamma p^n) d\Gamma = 0. \tag{52}$$

The essential and natural boundary conditions of Eq. (51) are:
$$p = 0 \qquad\qquad on\ \Gamma_\sigma$$
$$\bar{u}_i^{\,n+1} \cdot v = u_i^{\,n+1}\big|_s \cdot v \qquad on\ \Gamma_u$$

where $u_i^{\,n+1}\big|_s$ is the rigid body velocity obtained from Eq. (19).

Discrete equations
Using the approximations given be Eqs. (44), (45) and (46) the discrete equations become:
$$\int_V \mathbf{N}_i \mathbf{N}_i^T dV \, \mathbf{U}_i^* = \int_V \mathbf{N}_i \mathbf{N}_i^T dV \ \mathbf{U}_i^n + \Delta t \int_V \mathbf{N}_i f_i \, dV -$$

$$\frac{\gamma \, \Delta t}{\rho}\int_V \mathbf{N}_i \frac{\partial \mathbf{N}^T}{\partial x_i} \, dV \, \mathbf{P}^n - \frac{\Delta t \mu}{\rho}\int_V \frac{\partial \mathbf{N}_i}{\partial x_j}\frac{\partial \mathbf{N}_i^T}{\partial x_j} \, dV \, \mathbf{U}_i^{n+\theta} + \frac{\Delta t}{\rho}\int_{\Gamma_\sigma} \mathbf{N}_i (\bar{\sigma}_{ni} + \gamma \, p^n) d\Gamma. \tag{53}$$

In compact form:
$$\mathbf{M}\mathbf{U}^* = \mathbf{M}\mathbf{U}^n + \Delta t \, \mathbf{F} - \frac{\gamma \, \Delta t}{\rho}\mathbf{B}^T \, \mathbf{P}^n - \frac{\Delta t \mu}{\rho}\mathbf{K}\mathbf{U}^{n+\theta}. \tag{54}$$

Making use of the approximation described before for $\mathbf{U}^{n+\vartheta}$:
$$\left(\mathbf{M} + \frac{\Delta t \mu \theta}{\rho}\mathbf{K}\right)\mathbf{U}^* = \mathbf{M}\mathbf{U}^n + \Delta t \, \mathbf{F} - \frac{\gamma \, \Delta t}{\rho}\mathbf{B}^T \, \mathbf{P}^n - \frac{\Delta t \mu (1-\theta)}{\rho}\mathbf{K}\mathbf{U}^n, \tag{55}$$

For $\theta = 1$ and $\gamma = 0$:
$$\boxed{\left(\mathbf{M} + \frac{\Delta t \mu}{\rho}\mathbf{K}\right)\mathbf{U}^* = \mathbf{M}\mathbf{U}^n + \Delta t \, \mathbf{F}.} \tag{56}$$

In the same way:
$$-\frac{\rho}{\Delta t}\int_V \left(\frac{\partial \mathbf{N}}{\partial x_i}\mathbf{N}_i^T\right) dV \, \mathbf{U}^* + \frac{\rho}{\Delta t}\int_{\Gamma_u} \mathbf{N} \, \bar{u}_n^{\,n+1} d\Gamma = -\int_V \left(\frac{\partial \mathbf{N}}{\partial x_i}\frac{\partial \mathbf{N}^T}{\partial x_i}\right) dV \, (\mathbf{P}^{n+1} - \gamma \mathbf{P}^n). \tag{57}$$

In compact form:
$$\mathbf{S} \, \mathbf{P}^{n+1} = \frac{\rho}{\Delta t}(\mathbf{B} \ \mathbf{U}^* - \hat{\mathbf{U}}) + \mathbf{S} \, \gamma \, \mathbf{P}^n, \tag{58}$$

For $\theta = 1$ and $\gamma = 0$:

$$\mathbf{S}\,\mathbf{P}^{n+1} = \frac{\rho}{\Delta t}(\mathbf{B}\quad\mathbf{U}^* - \widehat{\mathbf{U}}). \qquad (59)$$

Finally:

$$\int_V \mathbf{N}_i \mathbf{N}_i^T dV \; \mathbf{U}^{n+1} =$$

$$\int_V \mathbf{N}_i \mathbf{N}_i^T dV \; \mathbf{U}^* - \frac{\Delta t}{\rho}\int_V \mathbf{N}_i \frac{\partial \mathbf{N}^T}{\partial x_i} dV \; (\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) + \int_{\Gamma_\sigma} \mathbf{N}_i \mathbf{N}^T d\Gamma \; (\mathbf{P}^{n+1} - \gamma \mathbf{P}^n). \qquad (60)$$

In compact form:

$$\mathbf{M}\mathbf{U}^{n+1} = \mathbf{M}\mathbf{U}^* - \frac{\Delta t}{\rho}\mathbf{B}^T(\mathbf{P}^{n+1} - \gamma \mathbf{P}^n) \qquad (61)$$

For $\theta = 1$ and $\gamma = 0$:

$$\mathbf{M}\mathbf{U}^{n+1} = \mathbf{M}\mathbf{U}^* - \frac{\Delta t}{\rho}\mathbf{B}^T\mathbf{P}^{n+1}. \qquad (62)$$

Where the matrices are:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_p & 0 & 0 \\ 0 & \mathbf{M}_p & 0 \\ 0 & 0 & \mathbf{M}_p \end{bmatrix}, \qquad (63)$$

$$\mathbf{M}_p = \int_V \mathbf{N}\mathbf{N}^T dV, \qquad (64)$$

$$\mathbf{B} = \left[\int_V \frac{\partial \mathbf{N}}{\partial x}\mathbf{N}^T dV \; ; \int_V \frac{\partial \mathbf{N}}{\partial y}\mathbf{N}^T dV \; ; \int_V \frac{\partial \mathbf{N}}{\partial z}\mathbf{N}^T dV \right], \qquad (65)$$

$$\mathbf{S} = \int_V \frac{\partial \mathbf{N}}{\partial x}\frac{\partial \mathbf{N}^T}{\partial x} + \frac{\partial \mathbf{N}}{\partial y}\frac{\partial \mathbf{N}^T}{\partial y} + \frac{\partial \mathbf{N}}{\partial z}\frac{\partial \mathbf{N}^T}{\partial z} dV, \qquad (66)$$

$$\widehat{\mathbf{U}} = \int_{\Gamma_u} \mathbf{N}\;\bar{u}_n^{n+1} d\Gamma, \qquad (67)$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} & 0 & 0 \\ 0 & \mathbf{S} & 0 \\ 0 & 0 & \mathbf{S} \end{bmatrix}, \qquad (68)$$

$$\mathbf{F}^T = \left[\int_V \mathbf{N}^T f_x \; dV; \quad \int_V \mathbf{N}^T f_y \; dV; \quad \int_V \mathbf{N}^T f_z \; dV \right] +$$

$$\frac{1}{\rho}\left[\int_{\Gamma_\sigma} \mathbf{N}^T \bar{\sigma}_{nx} \; d\Gamma; \quad \int_{\Gamma_\sigma} N^T \bar{\sigma}_{ny} \; d\Gamma; \quad \int_{\Gamma_\sigma} \mathbf{N}^T \bar{\sigma}_{nz} \; d\Gamma \right]. \qquad (69)$$

Stabilization of the incompressibility condition

In the Eulerian form of the momentum equations, the discrete form must be stabilized in order to avoid numerical wiggles in the velocity and pressure results. This is not the case in the lagrangian formulation where no stabilization terms must be added to eqs. (62). Nevertheless, the incompressibility condition must be stabilized in equal-order approximations to avoid pressure oscillations in some particular cases.

For instance for small pressure split ($\gamma \neq 0$) or for small time step increments (Courant number much less than one) it is well known that the fractional step does not stabilize the pressure waves [cod01]. In those particular cases, a stabilization term must be introduced in Eqs. (62) in order to eliminate pressure oscillations.

A simple and effective procedure to derive a stabilized formulation for incompressible flows is based in the so-called Finite Calculus (FIC) formulations [ona98, ona00, ona02]. In all the examples presented in this paper the FIC formulation was used to stabilize the pressure oscillations.

3.3 Non-linearity of the lagrangian formulation

Many algorithms are available to linearize the equations of motion. The Newton-Raphson scheme is probably the most popular because of its robustness and fast convergence. It has been applied with success in this type of formulation in [rad98]. Nevertheless we consider that it might not be the most appropriate option for the type of equations we are intending to solve as it requires large memory storage. Instead, the successive iteration algorithm has been chosen for the present analysis. In this case, only the variables that induces the non-linearity need to be stored in successive iterations. Let us now describe the process that may take place until convergence:

I)	Approximate u^{n+1} (For the first iteration $u^{n+1} = 0$. For the subsequent iterations the value of u^{n+1} corresponding to the last iteration is taken).
II)	Move the particles to the x^{n+1} position and perform an EDT polyhedrization.
III)	Evaluate the fractional velocity u^* from (56). It must be noted that the matrices M and K are separated in 3 blocks. Then, these equations may be solved separately for U_x^*, U_y^* and U_z^*. For $\theta \neq 0$ (implicit scheme) involves the solution of 3 symmetric linear systems of equations. For $\theta = 0$ (explicit scheme) the M matrix may be lumped and inverted directly.
IV)	Evaluate the pressure p^{n+1} by solving the Laplacian equation (58).
V)	Evaluate u^{n+1} using (61).

4. Time integration of the solid dynamics problem

Eqs. (14) and (15) that govern the movement of rigid bodies are integrated in time by the explicit Newmark algorithm. It consists in evaluating the velocity by linearizing the acceleration between two time steps:

$$U_i^{n+1} = U_i^n + \Delta t\left((1-\gamma)a_i^n + \gamma a_i^{n+1}\right).$$
$$\Omega_i^{n+1} = \Omega_i^n + \Delta t\left((1-\gamma)\alpha_i^n + \gamma\alpha_i^{n+1}\right) \tag{70}$$

The point position for the explicit version of the Newmark algorithm is evaluated by:
$$x_i^{n+1} = x_i^n + \Delta t\, U_i^n + \Delta t^2 a_i^n /2 \tag{71}$$
To integrate the angular acceleration in a 3D system by Newmark algorithm two steps are needed, namely:

predictor step:
$$\Omega_i^* = \Omega_i^n + \Delta t\left((1-\gamma)\alpha_i^n\right) \tag{72}$$
then, the accelerations are predicted by using Eq. (17):
$$\alpha_i^{n+1} = T_i^{n+1}/I_i - \varepsilon_{ijk}e_i\Omega_j^*(I_k\Omega_k^*)/I_i \tag{73}$$
corrector step
$$\Omega_i^{n+1} = \Omega_i^* + \Delta t\,\gamma\,\alpha_i^{n+1} \tag{74}$$

The linear velocities are integrated directly using Eqs.(16) and (70):
$$a_i^{n+1} = F_i^{n+1}/m$$
$$U_i^{n+1} = U_i^n + \Delta t\left((1-\gamma)a_i^n + \gamma a_i^{n+1}\right). \tag{75}$$

For the present analysis $\gamma = 1/2$ will be considered.

Both velocities, U_i^{n+1} and Ω_i^{n+1} are used in (18) to evaluate the new velocity of all the points of the body.

In the explicit version of the Newmark algorithm, the new position of the rigid body is evaluated by :
$$x_i^{n+1} = x_i^n + \Delta t\, U_i^n + \Delta t^2 a_i^n /2$$
$$\phi_i^{n+1} = \Delta t\,\Omega_i^n + \Delta t^2\,\alpha_i^n /2 \tag{76}$$
where x_i is the new position of the center of mass and ϕ_i are the angular rotations of the body.

It must be noted that for planar motion, the predictor step is unnecessary and α_i^{n+1} may be evaluated directly using (19).

4.1 The Coupled Problem
On the coupling boundary the fluid velocity and the solid velocity should converge to the same value. This could be expressed as:
$$u_f|_{\Gamma 1} = u_s|_{\Gamma 2} \tag{77}$$

Thus, two subsystems need to be solved, namely the fluid system:

$$N\left(u_f^{n+1}, p^{n+1}, x_f^{n+1}, u_s^{n+1}, x_s^{n+1}\right) = 0$$

and the solid system:

$$S\left(u_s^{n+1}, x_s^{n+1}, u_f^{n+1}, p^{n+1}, x_f^{n+1}\right) = 0.$$

In the equations above only the variables to be solved at time step $n+1$ are shown.

An iterative procedure has to be implemented to couple both systems. A fixed point algorithm may be implemented and thus the system could be written as:

$$\left(u_{f,k+1}^{n+1}, p_{k+1}^{n+1}, x_{f,k+1}^{n+1}\right) = F\left(u_{f,k}^{n+1}, p_k^{n+1}, x_{f,k}^{n+1}, u_s^{n+1}, x_s^{n+1}\right)$$

$$\left(u_{s,k+1}^{n+1}, x_{s,k+1}^{n+1}\right) = G\left(u_{s,k}^{n+1}, x_{s,k}^{n+1}, u_f^{n+1}, p^{n+1}, x_f^{n+1}\right)$$
(78)

The first equation in Eqs. (78) denotes the fluid subsystem and the second equation the solid subsystem. The subscript k is the iteration counter.

In the present analysis a Gauss-Seidel process has been chosen [cod96]. In this way, the iterative procedure means to solve first one of the subsystems, for instance the fluid system. Next the solid system is solved using the information from the fluid computation. Eqs. (78) should be modified and the final expression used for the computation is as follow:

$$\left(u_{f,k+1}^{n+1}, p_{k+1}^{n+1}, x_{f,k+1}^{n+1}\right) = F\left(u_{f,k}^{n+1}, p_k^{n+1}, x_{f,k}^{n+1}, u_{s,k}^{n+1}, x_{s,k}^{n+1}\right)$$

$$\left(u_{s,k+1}^{n+1}, x_{s,k+1}^{n+1}\right) = G\left(u_{s,k}^{n+1}, x_{s,k}^{n+1}, u_{f,k+1}^{n+1}, p_{k+1}^{n+1}, x_{s,k+1}^{n+1}\right)$$

Convergence occurs when the difference between velocities of successive iteration steps is less than the acceptable error.

5. Free-surface and boundary recognition

The solution of partial differential equations (PDE) requires to prescribe boundary conditions as a necessary step to a well-posed problem. When the PDEs are approximated in space and the domain is partitioned into discrete elements (finite elements, particles, balls, nodes, etc.) the boundary elements should be provided at the initial time step, such that, at run time the algorithm knows where to impose or fix the variables of the analysis (pressure, velocity and their derivatives for instance). This would be the case of a static domain, where the geometry does not change in time and the boundaries remain constant.

In this work the interest is focused on problems where the solution domain is highly distorted, and boundary elements can change between time steps. In this case an efficient boundary recognition algorithm is mandatory in order to impose boundary conditions over the right elements, thus avoiding possible error accumulation over the time.

When applying the MFEM [Ide03a] to the discrete space problem, the EDT [Ide03b] is computed to connect the particles that discretize the domain, thus, all the empty Voronoï spheres are found and stored. These spheres will be used to compute the boundary using the alpha-shape technique [ede94].

The particles will follow a given $h(x)$ distribution according to the maximum error allowed for the discrete space problem, where $h(x)$ is the expected distance among neighboring particles. Then, having all the empty Voronoï spheres and $h(x)$ the boundary particles are

regarded as: *all the particles which are on an empty sphere with a radius r bigger than αh.*

In this criterion, α is a parameter close to one, typically $\alpha = 1.2$ and h is the mean value taken from the defining particles of the sphere under inspection.

Once a decision has been made concerning which of the 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, one concave and the other convex.

In this work, the boundary surface is defined by all the polyhedron faces having all their nodes on the boundary and belonging to just one polyhedron. See [ide03b].

The correct boundary surface may be important to define the correct normal external to the surface. Furthermore; in weak forms (Galerkin) it is also important a correct evaluation of the volume domain. It must however be noted that in the criterion proposed above, the error in the boundary surface definition is of order h. This is the standard error of the boundary surface definition in a meshless method for a given node distribution.

Another important feature of the alpha-shape technique related to the contact problem is shown in Fig.5.1. The image shows two different time steps of a solid cube falling into water after the alpha-shape algorithm for the boundary recognition has been applied. The particles, as well as the connections provided by the EDT are depicted. At the first time step all the radii of the empty circles constructed with the nodes of the cube and the nodes regarded as belonging to the free-surface are larger than $\alpha h(x)$ and thus the elements that they define are eliminated from the tessellation. The second picture shows a more evolved state, with the cube reaching the water surface. Thus, at this state the circle radii are less than $\alpha h(x)$ and so the connections between the cube and the fluid take part of the computation. In this way free surface and contacts are solved at once.
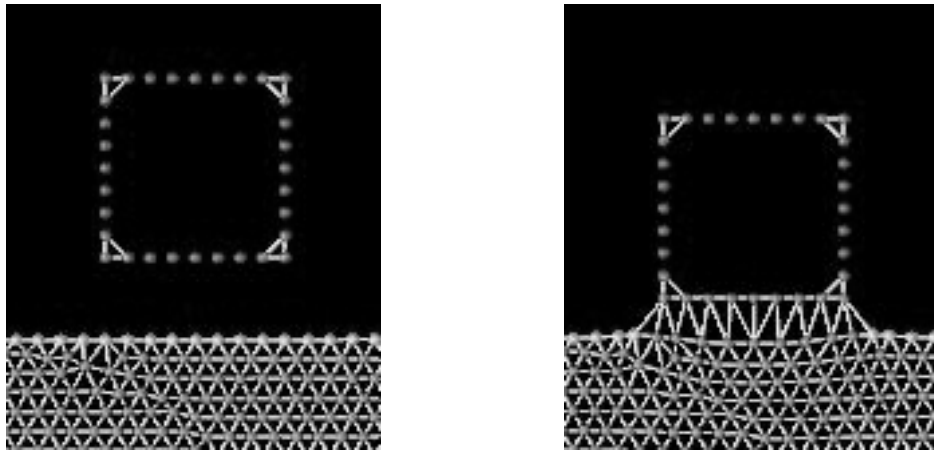

Fig. 5.1. *The alpha-shape technique used for contact recognition.*

6. Joining and breaking particles

The idea of h variable mesh is rather different in particle methods than in classical eulerian formulations. In particle methods, each particle is followed in time and the same particle can cross domains in which the solution need small h in order to represent high gradients or can cross a region with large h where the solution is smooth. The concept of variable h is introduce in particle methods by joining two particles when they are too close to each other or breaking a particle in two when all the neighboring particles are too far and the solution needs a higher gradient.

In the example presented in this paper the following criterion has been used:
1) During the EDT algorithm to build the polyhedral mesh a particle is not added if there is a previous point at a distance $d < \lambda\, h(x)$, being $\lambda = 0.5$ a constant parameter.
2) on the contrary, when there is an empty sphere whose radius $r > \Lambda\, h(x)$ a point is added to its center and assigned with values interpolated from the sphere defining particles. The parameter is currently taken as $\Lambda = 1.1*\mathrm{sqrt(dim)}/2$ in order to accept, with a 10% of tolerance, near-square (dim = 2, $\Lambda \approx .78$) or near-cube (dim = 3, $\Lambda \approx .95$) local arrays as connected inner points. This parameter must be less than the alpha-shape parameter $\alpha\, h(x)$ in order to avoid interference with the boundary recognition algorithm.

7. Validation Examples.

PFEM was developed as a general-purpose method for solving different kind of problems on which large free surface or interface boundaries changes are involved. The method is well suited to solve a large variety of mechanical problems including mixing fluid and solid materials, wave motion problems, mould filling, coupled thermal-mechanical problems and fluid-solid interaction as well.
In this section some problems are included in order to show the validation of the present approach towards experimental and numerical tests. In the following section, some more specific examples on fluid structure interaction will be shown.

7.1 Sloshing

The simple problem of the free oscillation of an incompressible liquid in a container is considered first. Numerical solutions for this problem can be found in several references [rad98]. This problem is interesting because there is an analytical solution for small amplitudes. For larger amplitudes the wave breaks and also some particles can be separated from the fluid domain due to their large velocity. PFEM can solve very large amplitudes, even in a 3D domain [ide04]. However, in this section only the small amplitude, two-dimensional example is shown to validate the method.

Figure 7.1 shows the variation in time 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.
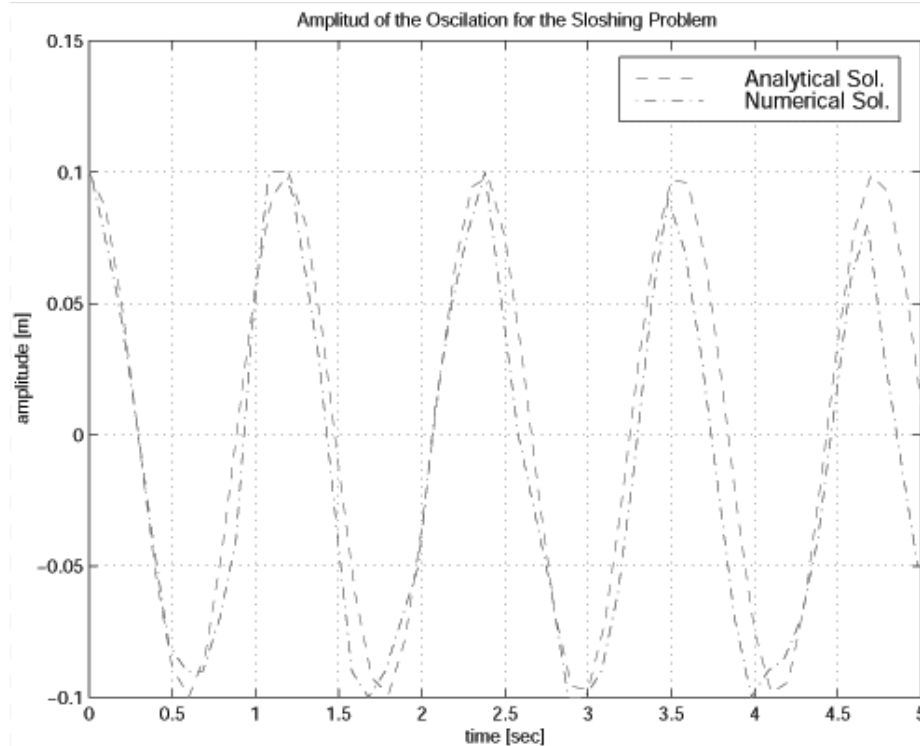
Fig. 7.1. *Sloshing: Comparison of the numerical and analytical solution.*

7.2 Wave on a channel: comparison with experimental results.

This example was performed in order to compare and validate the method with experimental results. A wave is running from the left to the right arriving to a shallow domain were the wave breaks. The example is represented on Fig 7.2 were the calculated particle positions are shown at different time steps. The wave was produce by a particular movement of the left wall. This example was reproduced experimentally in the CIEM (Maritime Experimental and Research Channel) of the Escuela Técnica de Ingenieros de Canales Caminos y Puertos in the University of Catalunya. The channel is 100 meters length, 3 meters wide and 5 meters high.

A pressure sensor was placed on the right wall at 0.2 meters from the bottom.
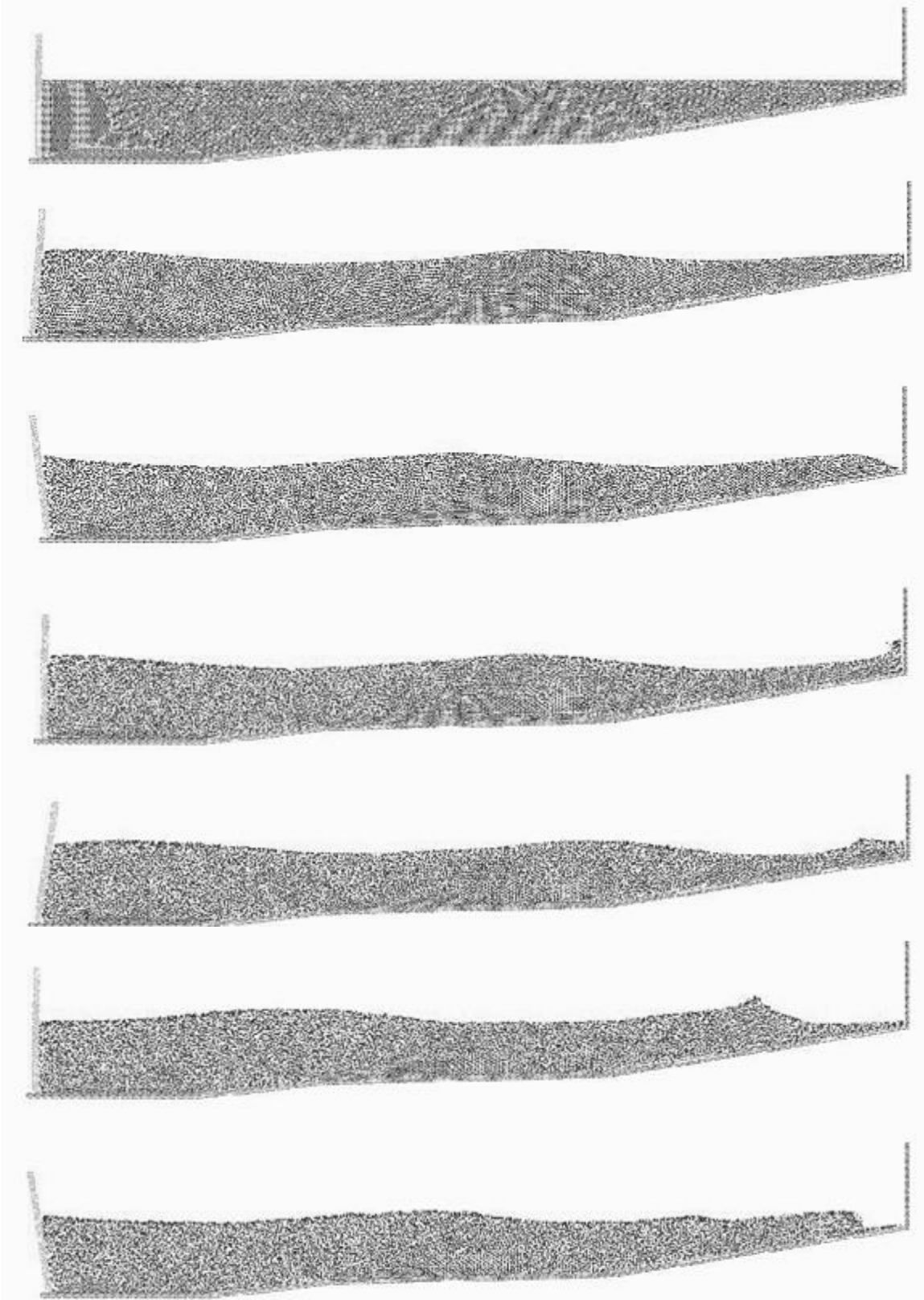
Fig. 7.2. *Wave on a channel: particle distribution for different time steps.*

Experimental and numerical pressure results are compared at different time step in Fig. 7.3. Both results, experimental and numerical, were smoothed in order to ignore the artificial oscillations from high order waves present in the problem. The comparison of the results in the pressure values shows a reasonably agreement
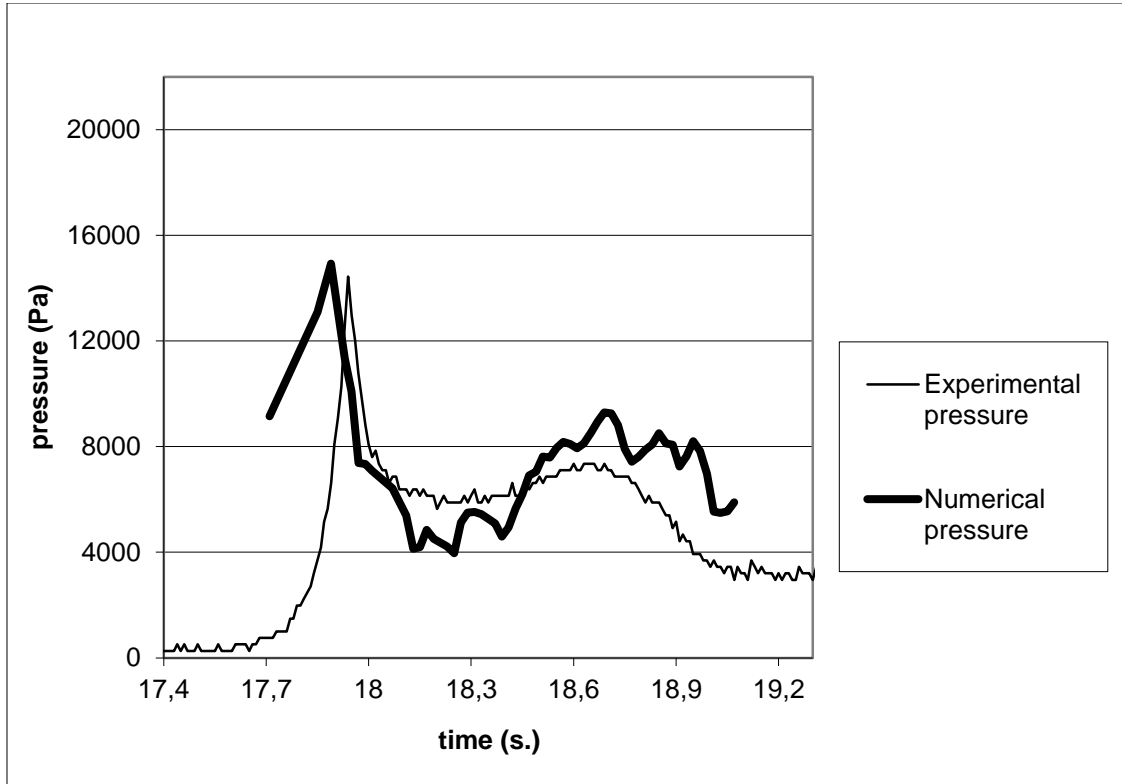

Fig. 7.3. *Wave on a channel: experimental and numerical comparison of the pressure.*

7.3 Dam collapse
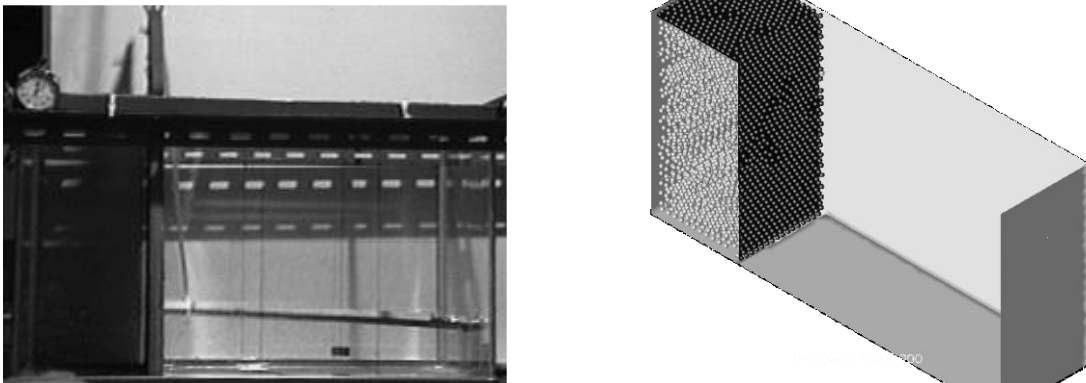
Fig. 7.4. *Dam Collapse. Initial position. Left: experimental [Kos96]. Right: 3D simulation.*

The dam collapse problem represented in Figure 7.4 was solved by Koshizu and Oka [Kos96] both experimentally and numerically in a 2D domain. It became a classical example to test the validation of the Lagrangian formulation in fluid flows.

The water is initially located on the left supported by a removable board. The collapse starts at time $t = 0$, when the removable board is slid-up. Viscosity and surface tension are neglected. The water is running on the bottom wall until, near 0.3 sec, it impinges on the right vertical wall. Breaking waves appear at 0.6 sec. Around t=1 sec. the main water wave reaches the left wall again.

In [Ide04] the results obtained using the method proposed in 2D and 3D domains are presented and compared with experimental results. Agreement with the experimental results of [kos96] both in the shape of the free surface and in the time development are excellent.

In this example the power of the method to represent breaking waves and flow separation for a very complicated and random problem is verified and compared to experimental results.

This example is further exploited here to compare results obtained with three different node densities at some time steps in order to check the convergence of the method.
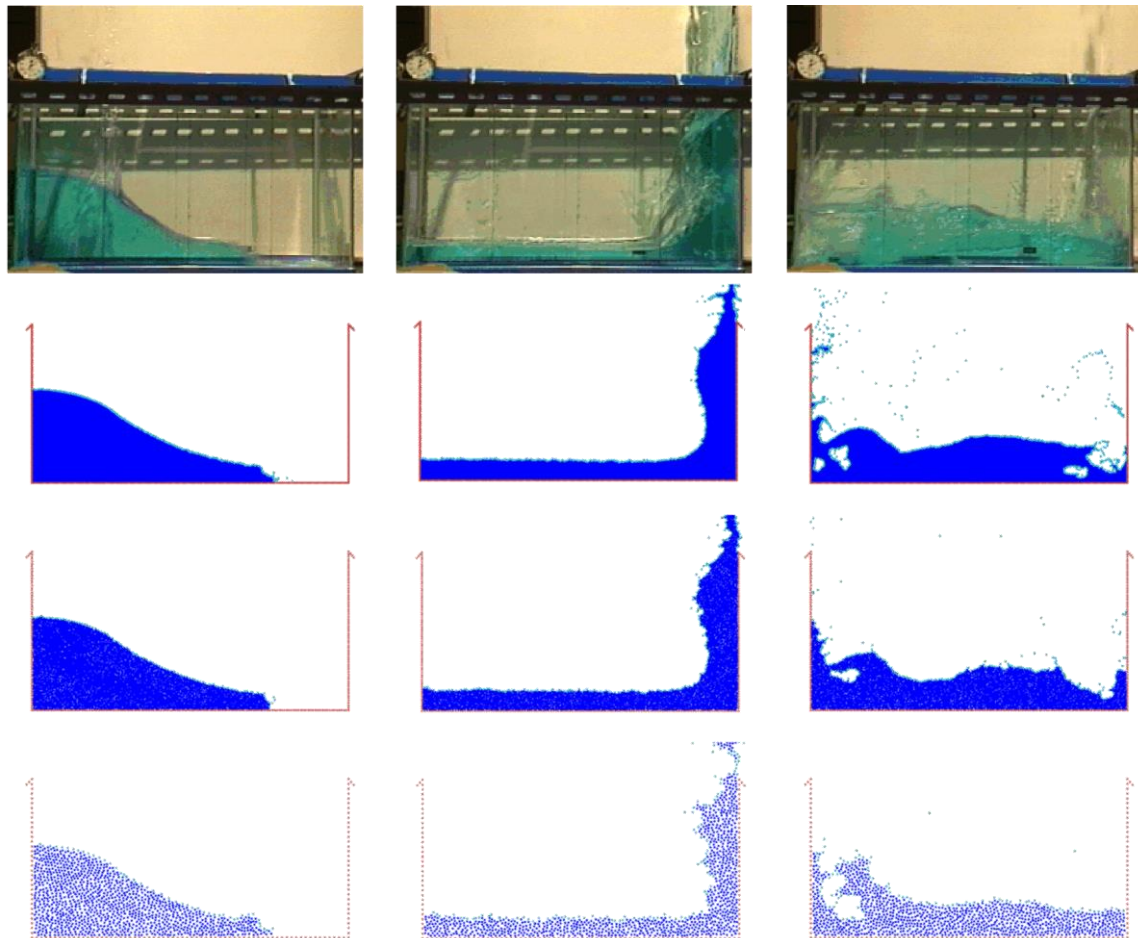

Fig. 7.5. *Dam Collapse. Comparison between the experiment and numerical results obtained in different time steps, with different refinement levels.*

Figure 7.5 shows the domain profile at different time steps and with different refinement levels. At the top there are some photos taken on the experimental setup corresponding to

.2, .6 and 1 second from the starting point. From top to bottom grids with 2.2, 4.4 and 8.8mm in typical distance between neighboring particles are shown.

The excellent agreement already obtained with low refinement is even better with higher level of refinement.

8 Further Examples on fluid-structure interaction

8.1 Ship profile hit by a wave

In the example of Figure 8.1 the motion of a fictitious rigid ship hit by an incoming wave is analyzed. This is the first example in which the rigid body is moved by the fluid forces in a coupling problem as was explained in the previous chapters.

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 mass centre 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 is evaluated using eq. (76) and the velocity of the body by using eq. (18). This defines a moving boundary condition for the free surface particles in the fluid as introduced in eq. (59).

Figure 7.7 shows instants of the motion of the ship and the surrounding fluid. It is interesting to see the breaking of a wave on the ship prow at t= 0.91 sec. as well as on the stern at t= 2.05 sec. when the wave goes back. Note that some water drops slip over the ship deck at t= 1.3 sec. and 2.95 sec.

8.2 Oil ship tank under a lateral wave.

The present example depicts the flexibility of the algorithm introduced in this paper to solve some complicated configurations as the one shown in Fig. 8.2. The traversal cut of an oil ship tank is hit by a wave. The structure of the ship does not only interact with the external water but it also moves due to the fluid forces induced by the fluid in the tank.

Fig. 8.2 shows the temporal development of the problem. The blue lines over each particle represent the magnitude of the velocity field.

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 = 5.10 sec and 6.00 sec. breaking waves and some water drops slipping along the ship deck can be observed. Figure 8.3 shows the pressure pattern at two time steps.

8.3. Tanker Sinking

This example represents the sinking of a tanker, which is being flooded by the ship prow. The ship has many connected compartments that are serially flooded.
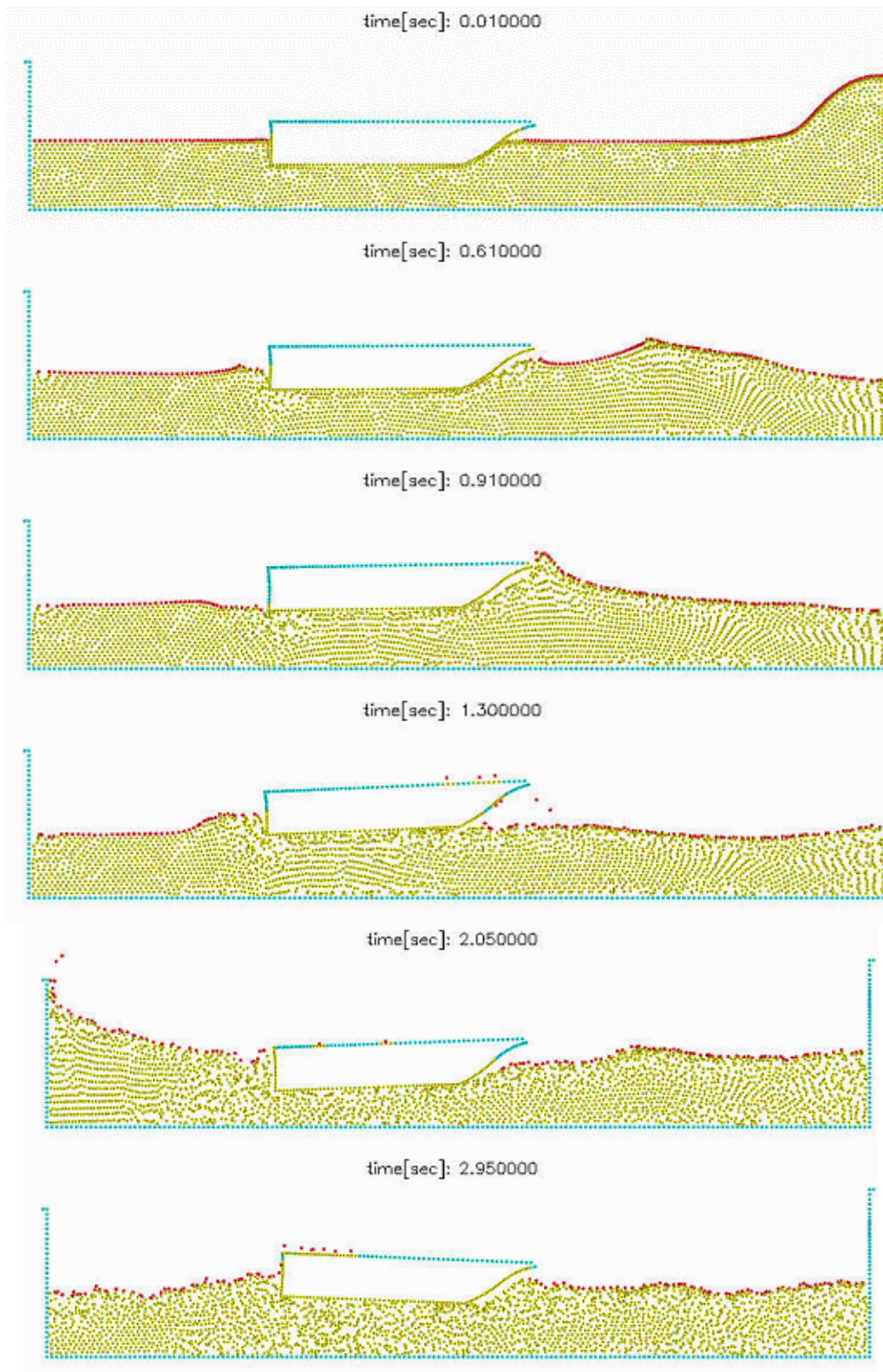
time[sec]: 0.010000

time[sec]: 0.610000

time[sec]: 0.910000

time[sec]: 1.300000

time[sec]: 2.050000

time[sec]: 2.950000

Fig. 8.1. *Ship profile hit by a wave: particle positions for different time steps*
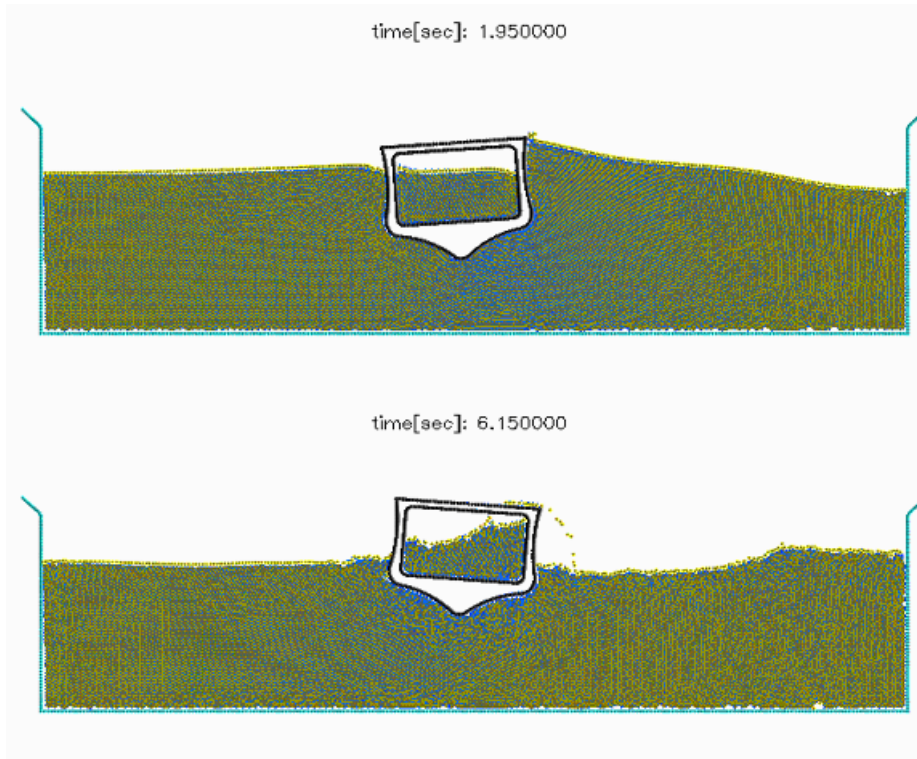
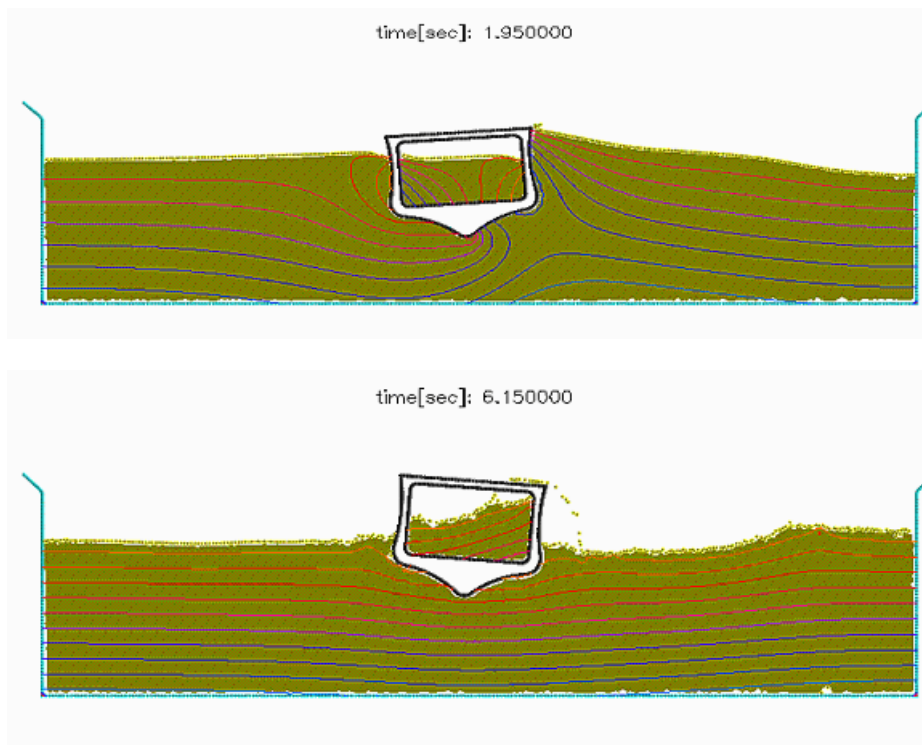**Fig. 8.2.** *Oil ship tank under a lateral wave: Particle distribution and velocity field.*

Fig. 8.3. *Oil ship tank under a lateral wave: Pressure profile at two different time steps.*
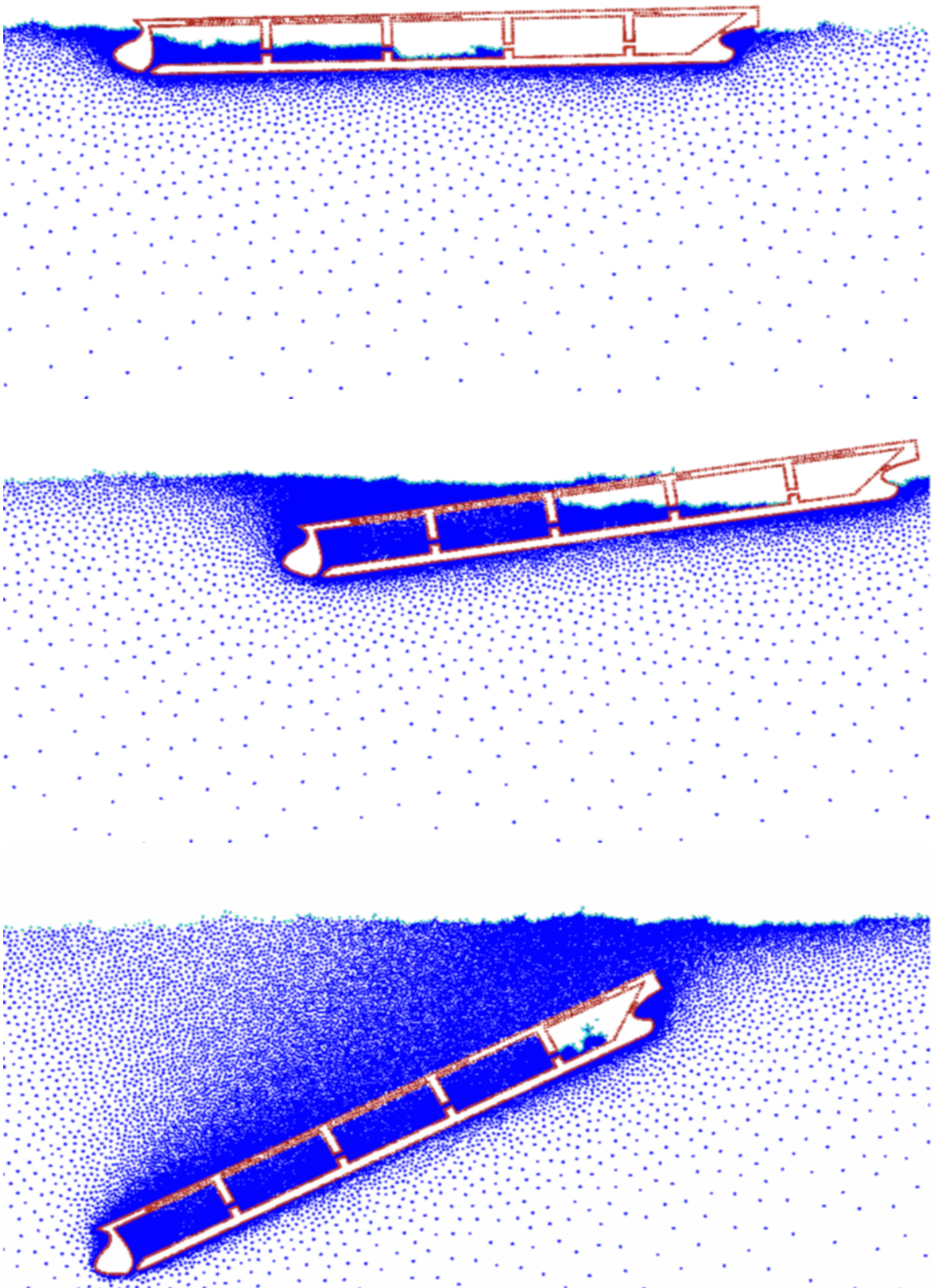
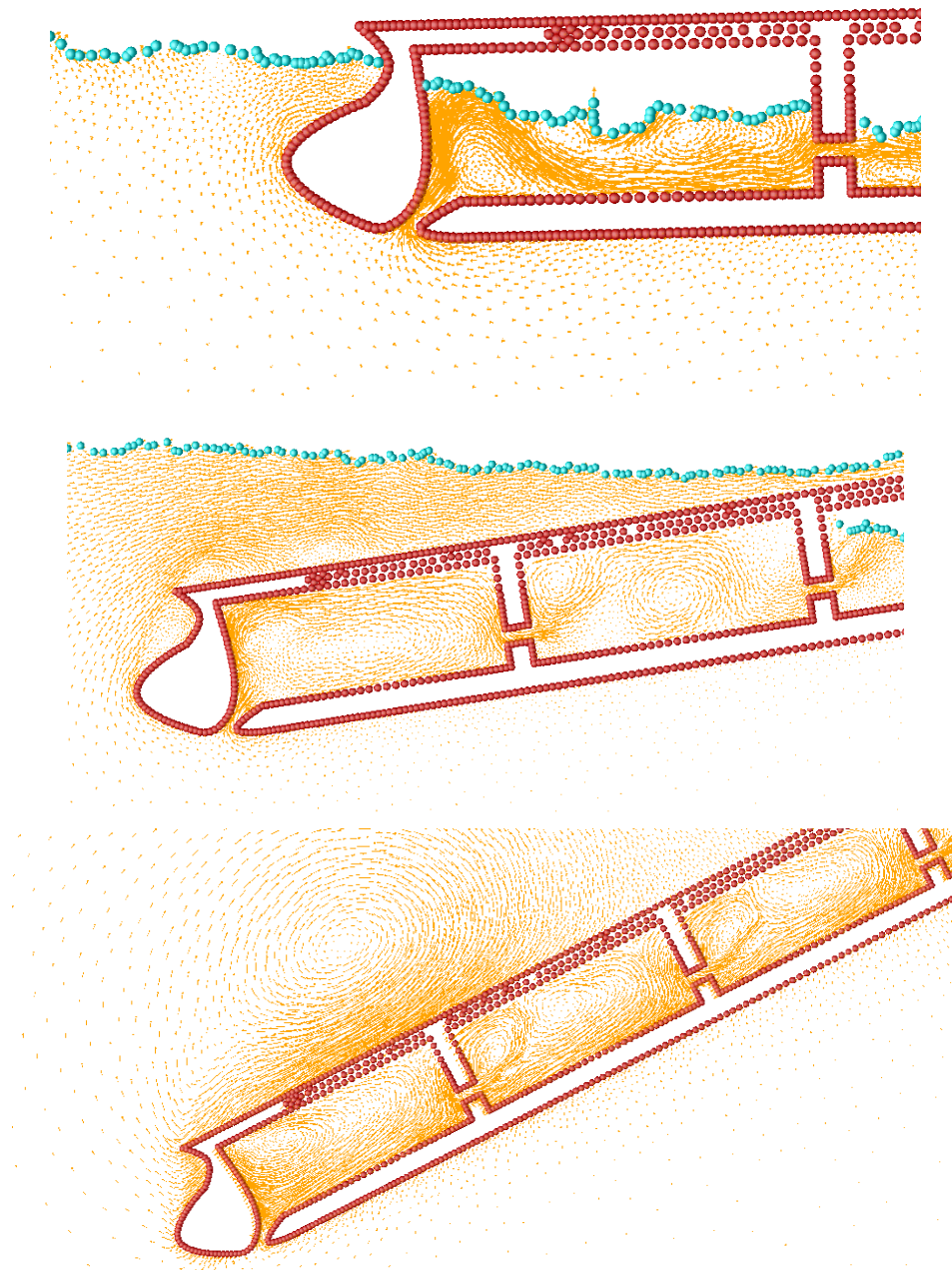***Fig. 8.4.*** *Sinking tanker. Particle distribution at three different time-steps.*

Fig. 8.5. *Sinking tanker. Velocity field.*

In this example proper fluid-structure interaction is displayed as the buoyancy, pressure and drag forces from the fluid are acting over the ship and, on the other hand, the ship displacement moves internal and external free surfaces on the fluid.

In Figure 8.4 there are three time-steps shown with the particle positions and the tanker in three different sinking stages.

Figure 8.5 displays velocity profile on a zoom of the first and last time steps form previous figure. In this figure, the vorticity is also easily shown.

This is an interesting example using a variable distance between particles to enhance the solution near the ship and free surfaces. This variable distribution was obtained following the method outlined in section 6 above.

The large cyan dots are representing the free surface recognized by the method as explained in section 5.

9. 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 at each time step of the equations of an incompressible fluid and the interacting solid. Essential ingredients of the numerical solution are the efficient regeneration of the polyhedral mesh using an extended Delaunay tessellation, the polyhedral finite element interpolation via the 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. Other examples of application of the PFEM can be found in [ona04].

Acknowledgements

References

[aub04] Aubry R., Idelsohn S.R. and Oñate. "*The Particle Finite Element Method in fluid mechanics including thermal convection-difusion*". Submitted to Computer & Structures. (2004).

[bel94] T. Belytschko, L. Gu and Y. Y. Lu, "*Fracture and crack growth by element free Galergin methods*", Modelling Simul. Mater Sci. Eng., vol. 2, pp. 519-534, 1994.

[cod01] R.Codina, "*Pressure Stability in fractional step finite element methods for incompressible flows*", Journal of Comput. Phisycs, 170, (2001) 112-140.

[cod96] Codina R, Cervera M. "*Block-iterative algorithms for nonlinear coupled problems*". In: Papadrakakis M, Bugeda G, editors, Advanced Computational Methods in Structural mechanics. Barcelona: CIMNE, 1996.

[dua95] C. Duarte and J. D. Oden, "*Hp clouds-a meshless method to solve boundary value problems*", Tech. Rep. 95-05, Texas institute for Computational and Applied Mechanics, Univerfsity of Texas, 1995.

[ede94] H. Edelsbrunner and E.P. Mucke, *"Three-dimensional alpha-shape"*. ACM Transactions on Graphics, 3, (1994), 43–72.

[ide03a] S.R. Idelsohn, E. Oñate, N. Calvo and F. Del Pin, *"The meshless finite element method"*, Int. J. for Numerical Methods in Engineering, Vol 58, Issue 4, (2003).

[ide04] S.R. Idelsohn, E. Oñate and F. Del Pin, *"The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves"*. Int. J. for Numerical Methods in Engineering, Vol 61, Issue 7, (2004), Pages: 964-989.

[ide03b] S.R. Idelsohn, N. Calvo and E. Oñate, *"Polyhedrization of an arbitrary 3D point set"*. Computer Method in Applied Mechanics and Engineering, 192, pp 2649-2667 (2003).

[kos96] S. Koshizuka and Y. Oka *"Moving particle semi-implicit method for fragmentation of incompressible fluid"*, Nuclear Engineering Science, 123, (1996), 421–434.

[loh03] R. Löhner, J.D. Baum, E.L. Mestreau, D. Sharov, Ch. Charman and D. Pelessone *"Adaptive Embedded Unstructured Grid Methods"*; AIAA-03-1116 (2003).

[mok01] Mok DP, Wall WA. *"Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures"*. In: Wall WA, Bletzinger K-U, Schweizerhof K, editors. Trends in Computational Structural Mechanics. Barcelona: CIMNE, 2001.

[mon81] R.A. Gingold and J.J. Monaghan, *"Kernel estimates as a basis for general particle methods in hydrodynamics"*, Journal of Computational Physics, vol. 46, pp. 429-453, 1981.

[mon97] R.A. Gingold and J.J. Monaghan, *"Smoothed particle hydrodynamics, theory and application to non-spherical stars"*, Mon. Nat. Roy. Astr. Soc., 181, (1997), 375–389.

[nay92] B. Nayroles, G. Touzot and P. Villon. *"Generalizing the finite element method: diffuse approximation and diffuse elements"*, Computational Mechanics, vol. 10, pp. 307-318, 1992.

[ona98] E. Oñate. *"Derivation of stabilized equations for advective-diffusive transport and fluid flow problems"*. Comput. Methods Appl. Mech. Engrg.,1998, **151:1-2**, 233--267

[ona00] E. Oñate. *"A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation"*. Comp. Meth. Appl. Mech. Engng. ,2000, **182:1-2**, 355-370.

[ona02] E. Oñate, *"Possibilities of finite calculus in computational mechanics"*. (2002) Submitted to Int. J. Num. Meth. Engng.

[ona04] E. Oñate, S.R. Idelsohn, F. Del Pin and R. Aubry. *"The Particle Finite Element Method: an overview"*. In press in International Journal of Computational Methods. (2004)

[pip95] Piperno S, Farhat C, Larrouturou B. *"Partitioned procedures for the transient solution of coupled aeroelastic problems"*. Comp Meth Appl Mech Eng 1995; 124:79-112.

[rad98] R. Radovitzky, M. Ortiz *"Lagrangian finite element analysis of a Newtonian flows"*. Int. J. Numer. Engng. 43, 607-619 (1998).

[ram87] B. Ramaswamy and M. Kawahara, *"Lagrangian finite element analysis of Newtonian fluid flows"*, Int. J. Num. Meth. Fluids, vol. 7, pp. 953-984, 1987.

[rug00] Rugonyi S, Bathe KJ. *"On the analysis of fully-coupled fluid flows with structural interactions - a Coupling and Condensation Procedure"*. Int. J of Comp Civil and Struct Engineering 2000; 1:29-41.

[rug01] Rugonyi S, Bathe KJ. "*On finite element analysis of fluid flows fully coupled with structural interactions*". Computer Modeling and Simulation in Engineering (CMES) 2001; 2:195-212.

[Sou00] M. Souli, A. Ouahsine and L. Lewin, *"Arbitrary Lagrangian Eulerian formulation for Fluid-Structure Interaction Problems"*. Comp. Meth. In Applied Mech. and Eng. Vol. 190, pp 659-675 (2000).

[suk98] N. Sukumar, B. Moran and T. Belytschko, "*The natural element method in solid mechanics*", International Journal for Numerical Methods in Engineering, vol. 43, pp. 839-887, 1998.

[zha01] Zhang H, Bathe KJ. "*Direct and iterative computing of fluid flows fully coupled with structures*", in *Computational Fluid and Solid Mechanics* (K.J. Bathe, ed.), Elsevier, 2001.