Available online at www.sciencedirect.com
SCIENCE DIRECT®
PERGAMON
Computers and Structures 81 (2003) 655–671
Computers & Structures
www.elsevier.com/locate/compstruc

A Lagrangian meshless finite element method applied to fluid–structure interaction problems

S.R. Idelsohn [a,b,*], E. Oñate [b], F. Del Pin [a]

[a] International Center for Computational Methods in Engineering (CIMEC), Universidad Nacional del Litoral and CONICET, Guemes 3450, 3000 Santa Fe, Argentina

[b] International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Edificio C1 Campus Nord, 08032 Barcelona, Spain

Abstract

A method is presented for the solution of the incompressible fluid flow equations using a Lagrangian formulation. The interpolation functions are those used in the meshless finite element method and the time integration is introduced in a semi-implicit way by a fractional step method. Classical stabilization terms used in the momentum equations are unnecessary due to the lack of convective terms in the Lagrangian formulation. Furthermore, the Lagrangian formulation simplifies the connections with fixed or moving solid structures, thus providing a very easy way to solve fluid–structure interaction problems.

Keywords: Fluid–structure interaction; Particle methods; Lagrange formulations; Incompressible fluid flows; Meshless methods; Finite element method

1. Introduction

Over the last 20 years, computer simulation of incompressible fluid flow has been based on the Eulerian formulation of the fluid mechanics equations. However, it is still difficult to analyze problems in which the shape of the interface changes continuously or in fluid–structure interactions (FSI) with free-surfaces where complicated contact problems are involved.

More recently, *particle methods* in which each fluid particle is followed in a Lagrangian manner have been used [1–4]. The first ideas on this approach were proposed by Gingold and Monaghan [1] for the treatment of astrophysical hydrodynamic problems with the so called *smooth particle hydrodynamics method* (SPH). This method was later generalized to fluid mechanic problems [2–4]. Kernel approximations are used in the SPH method to interpolate the unknowns.

On the other hand, a family of methods called *meshless methods* have been developed both for structural [5,6] and fluid mechanics problems [8–10]. All these methods use the idea of a polynomial interpolant that fits a number of points minimizing the distance between the interpolated function and the value of the unknown point. These ideas were proposed first by Nayroles et al. [7], they were later used in structural mechanics by Belytschko et al. [5] and in fluid mechanics problems by Oñate and co-workers [8–10]. In a previous paper, [11] the authors presented the numerical solution for the fluid mechanics equations using a Lagrangian formulation and a meshless method called the finite point method. Lately, the meshless ideas were generalized to take into account the finite element type approximations in order to obtain the same computing time in mesh generation as in the evaluation of the meshless connectivities [12,13]. This method was called the meshless

* Corresponding author. Address: International Center for Computational Methods in Engineering (CIMEC), Universidad Nacional del Litoral and CONICET, Guemes 3450, 3000 Santa Fe, Argentina.

E-mail addresses: sergio@ceride.gov.ar (S.R. Idelsohn), onate@cimne.upc.es (E. Oñate).

0045-7949/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0045-7949(02)00477-7

finite element method (MFEM) and uses the extended Delaunay tessellation (EDT) [14] to build the mesh in a computing time which is linear with the number of nodal points.

In this paper, new ideas and results for the solution of a particle method in the field of FSI using the MFEM are presented. A more general formulation is used in which all the classical advantages of the FEM for the evaluation of the unknown functions and derivatives are preserved.

Different strategies have been proposed to solve FSI problems. The selection of the most effective approach depends largely on the nature of the problem to be analyzed [15]. Depending on the degree of coupling between the equations for the fluid and the structure, two cases can be distinguished. The first one occurs when there is a strong coupling between the fluid flow and the elastic deformation of the structure [15–17]. The second case occurs when there is a weak interaction between the fluid and the rigid deformation of the structure. In the latter, the solid must undergo large rigid displacements interacting with the fluid. This is the case for instance of sea-keeping in ship hydrodynamics, rotating turbines, mills, and other engines with a moving solid inside a fluid. Both cases of FSI are more easily studied with a Lagrangian formulation of the fluid equations, which can be seen as a solid with a small shear coefficient or vice versa.

The Lagrangian fluid flow equations for the Navier–Stoke problem will be revised in the next section, the MFEM will be summarized in Appendix A and both techniques will be used to solve some FSI problems for rigid solids.

2. Governing equations

The mass and momentum conservation equations can be written in a Lagrangian formulations as
mass conservation:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t}+\rho\frac{\partial u_i}{\partial x_i}=0 \tag{1}$$

momentum conservation:

$$\rho\frac{\mathrm{D}u_i}{\mathrm{D}t}=-\frac{\partial}{\partial x_i}p+\frac{\partial}{\partial x_j}\tau_{ij}+\rho f_i \tag{2}$$

where ρ is the density u_i are the Cartesian components of the velocity field, p the pressure, τ_{ij} the deviator stress tensor, f_i the source term (normally the gravity) and $\mathrm{D}\phi/\mathrm{D}t$ represents the total or material time derivative of a function ϕ.

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

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

For near incompressible flows $\left(\frac{\partial u_i}{\partial x_i}\ll\frac{\partial u_k}{\partial x_l}\right)$ the term

$$\frac{2\mu}{3}\frac{\partial u_i}{\partial x_i}\approx 0 \tag{4}$$

and it may be neglected in Eq. (3). Then

$$\tau_{ij}\approx\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{5}$$

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

$$\begin{aligned}\frac{\partial}{\partial x_j}\tau_{ij}&=\frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\right)\\&=\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)+\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_j}{\partial x_i}\right)\\&=\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)+\mu\frac{\partial}{\partial x_i}\left(\frac{\partial u_j}{\partial x_j}\right)\approx\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)\end{aligned} \tag{6}$$

Then, the momentum equations can be finally written as

$$\begin{aligned}\rho\frac{\mathrm{D}u_i}{\mathrm{D}t}&=-\frac{\partial}{\partial x_i}p+\frac{\partial}{\partial x_j}\tau_{ij}+\rho f_i\\&\approx-\frac{\partial}{\partial x_i}p+\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)+\rho f_i\end{aligned} \tag{7}$$

Boundary conditions

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

$$\tau_{ij}v_j-pv_i=\bar{\sigma}_{ni}\quad\text{on }\Gamma_\sigma \tag{8}$$

$$u_iv_i=\bar{u}_n\quad\text{on }\Gamma_n \tag{9}$$

$$u_i\zeta_i=\bar{u}_t\quad\text{on }\Gamma_t \tag{10}$$

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

3. The time splitting

The time integration of Eqs. (7) and (8) presents some difficulties when the fluid is incompressible or nearly incompressible. In this case, explicit time steps cannot be used. Even when using an implicit time integration scheme, incompressibility introduces some wiggles in the pressure solution which must be stabilized. To overcome these difficulties, a fractional step method has been proposed [18] which consists in splitting each time step in 2 steps as follows.

Split of the momentum equations

$$\frac{\mathrm{D}u_i}{\mathrm{D}t} \approx \frac{u_i^{n+1}-u_i^n}{\Delta t} = \frac{u_i^{n+1}-u_i^*+u_i^*-u_i^n}{\Delta t} = \left(-\frac{1}{\rho}\frac{\partial}{\partial x_i}p + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j} + f_i \right)^{n+\theta} \tag{11}$$

where $\Delta t = t^{n+1}-t^n$ is the time step; $u_i^n = u_i(x^n,t^n)$; $u_i^{n+1} = u_i(x^{n+1},t^{n+1})$ and u_i^* are fictitious variables defined by the split

$$\text{(A)}\quad u_i^* = u_i^n + f_i\Delta t - \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}\gamma p^n + \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}\tau_{ij}^n \tag{12}$$

$$\text{(C)}\quad u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}(p^{n+1}-\gamma p^n) \tag{13}$$

where γ is parameter equal to zero or one defining a first or second order split, respectively [18].

Split of the mass conservation equations

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} \approx \frac{\rho^{n+1}-\rho^n}{\Delta t} = \frac{\rho^{n+1}-\rho^*+\rho^*-\rho^n}{\Delta t} = -\rho\frac{\partial(u_i^{n+1}-u_i^*+u_i^*)}{\partial x_i} \tag{14}$$

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

$$\frac{\rho^*-\rho^n}{\Delta t} = -\rho\frac{\partial u_i^*}{\partial x_i} \tag{15}$$

$$\frac{\rho^{n+1}-\rho^*}{\Delta t} = -\rho\frac{\partial(u_i^{n+1}-u_i^*)}{\partial x_i} \tag{16}$$

Coupled equations

From Eqs. (13) and (16) the coupled mass–momentum equation becomes

$$\text{(B)}\quad \frac{\rho^{n+1}-\rho*}{\Delta t^2} = \frac{\partial^2}{\partial x_i^2}(p^{n+1}-\gamma p^n) \tag{17}$$

Taking into account Eq. (15) above expression can be written as

$$\text{(B)}\quad \frac{\rho^{n+1}-\rho^n}{\Delta t^2} + \frac{\rho}{\Delta t}\frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2}(p^{n+1}-\gamma p^n) \tag{18}$$

4. Incompressibility conditions

The simplest way to introduce the incompressibility condition is to write

$$\rho^{n+1} = \rho^n = \rho^0 = \rho \tag{19}$$

Then, the first term of Eq. (18) disappears. Nevertheless, in a Lagrangian formulation it is better to evaluate this term in order to avoid possible numerical errors at each time step. The incompressibility condition is introduced by imposing that at time step t^{n+1} the density must be equal to the initial one, i.e.

$$\rho^{n+1} = \rho^0 = \rho \tag{20}$$

Due to numerical errors the density ρ^n is not necessarily equal to ρ_0 and it must be updated at each time step. A different way to evaluate ρ^n will be explained in Section 7. Eq. (18) is finally written as

$$\text{(B)}\quad \frac{\rho^0-\rho^n}{\Delta t^2} + \frac{\rho}{\Delta t}\frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2}(p^{n+1}-\gamma p^n) \tag{21}$$

Then, the total time step may be described as follows: starting with the known value u^n and p^n from the previous time increment, the computation of the new velocities and the pressure involves the following five steps.

(I) Evaluate the u^* velocity from (12).
(II) Evaluate the new density ρ^n (see Section 7).
(III) Evaluate the pressure p^{n+1} solving the Laplacian Eq. (21).
(IV) Evaluate the velocity u^{n+1} using (13).
(V) Move the particles to the x^{n+1} position.

5. Spatial discretization

The Lagrangian split scheme described in the previous section has two important advantages.

(1) Step I is linear and explicit. The use of a Lagrangian formulation eliminates the standard convection terms present in Eulerian formulations. The convection terms are responsible for non-linearity, non symmetry and non self-adjoint operators which require the introduction of high order stabilization terms to avoid numerical oscillations. All these problems are not present in this formulation.

(2) In all the five steps described in previous section, the only implicit step is the solution of the Laplacian of pressure (step III). This is a scalar, symmetric and positive definite equation. Then, it is very easy to solve it using an iterative scheme (such as the conjugate gradient method).

The big disadvantage of the Lagrangian formulation is the permanent updating of the node positions. That is the reason why standard finite element methods are not useful, as the process of updating conforming non-structured finite element meshes is expensive.

The key of the Lagrangian formulation is the efficiency in the mesh updating process. In a previous paper [11], the authors evaluated the use of a meshless method for this purpose. In [11] a meshless method based in point collocation was used. This introduces some difficulty in prescribing the boundary conditions.

Other meshless methods as the element free Galerkin method (EFGM) [5] or the natural element method (NEM) [19] have difficulties to solve arbitrary point distributions in a 3-D domain due to the complicated shape functions used.

In this paper, the MFEM proposed in Ref. [12] will be used. The method is summarized in Appendix A.

The big advantage of the MFEM compared with the FEM is the possibility of generating meshes in a computing time of order n, being n the total number of nodes [14]. Compared with EFGM or NEM, the advantages are the simplicity of the shape functions, which are coincident with the FEM shape functions in most parts of the domain.

Using the MFEM, the unknown functions are approximated using an equal order interpolation for all variables as (in matrix form)

$$u_i = N_i^{\mathrm{T}} U = \begin{bmatrix} N^{\mathrm{T}} & & \\ & N^{\mathrm{T}} & \\ & & N^{\mathrm{T}} \end{bmatrix} U \tag{22}$$

$$p = N_p^{\mathrm{T}} P = N^{\mathrm{T}} P \tag{23}$$

$$\rho = N_\rho^{\mathrm{T}} \underline{\rho} = N^{\mathrm{T}} \underline{\rho} \tag{24}$$

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

Using the Galerkin weighted residual method to solve Eqs. (12), (13) and (21) with boundary conditions (8)–(10) the following integral equations can be written

$$\text{(A)}\quad \int_V N_i \left\{ (u_i^* - u_i^n)\frac{\rho}{\Delta t} - f_i \rho + \frac{\partial}{\partial x_i}\gamma p^n - \mu \frac{\partial \tau_{ij}^n}{\partial x_j} \right\} \mathrm{d}V - \int_{\Gamma_\sigma} N_i (\bar{\sigma}_{ni}^n - (\tau_{ij}^n v_j - \gamma p^n v_i)\,\mathrm{d}\Gamma = 0 \tag{25}$$

$$\text{(B)}\quad \int_V N_p \left\{ \frac{\rho^0 - \rho^n}{\Delta t^2} + \frac{\rho}{\Delta t}\frac{\partial}{\partial x_i} u_i^* - \frac{\partial^2}{\partial x_i^2}(p^{n+1} - \gamma p^n) \right\} \mathrm{d}V + \frac{\rho}{\Delta t}\int_{\Gamma u} N_p (\bar{\boldsymbol{u}}_i^{n+1} v_i - u_i^{n+1} v_i)\,\mathrm{d}\Gamma = 0 \tag{26}$$

$$\text{(C)}\quad \int_V N_i \left\{ (u_i^{n+1} - u_i^*)\frac{\rho}{\Delta t} + \frac{\partial}{\partial x_i}(p^{n+1} - \gamma p^n \right\} \mathrm{d}V - \int_{\Gamma_\sigma} N_i (p^{n+1} - \gamma p^n) v_i\,\mathrm{d}\Gamma = 0 \tag{27}$$

where the boundary conditions have been also split.

Integrating by parts some of the terms, the above equations become

$$\text{(A)}\quad \int_V N_i (u_i^* - u_i^n - f_i \Delta t)\frac{\rho}{\Delta t}\,\mathrm{d}V + \int_V N_i \frac{\partial}{\partial x_i}\gamma p^n + \mu \int_V \frac{\partial N_i}{\partial x_i}\frac{\partial u_i^n}{\partial x_i}\,\mathrm{d}V - \int_{\Gamma\sigma} N_i \bar{\sigma}_{ni}^n\,\mathrm{d}\Gamma = 0 \tag{28}$$

$$\text{(B)}\quad \frac{1}{\Delta t^2}\int_V N_p(\rho^0 - \rho^n)\,\mathrm{d}V - \int_V \frac{\partial N_p}{\partial x_i}\left(\frac{\rho}{\Delta t} u_i^* - \frac{\partial (p^{n+1} - \gamma p^n)}{\partial x_i} \right) \mathrm{d}V + \frac{\rho}{\Delta t}\int_{\Gamma_u} N_p \bar{\boldsymbol{u}}_n^{n+1}\,\mathrm{d}\Gamma = 0 \tag{29}$$

$$\text{(C)}\quad \int_V N_i \left\{ (u_i^{n+1} - u_i^*)\frac{\rho}{\Delta t} + \frac{\partial}{\partial x_i}(p^{n+1} - \gamma p^n) \right\} \mathrm{d}V - \int_{\Gamma_\sigma} N_i (p^{n+1} - \gamma p^n)\,\mathrm{d}\Gamma = 0 \tag{30}$$

It must be noted than the essential and natural boundary conditions of equations (29) are

$$p = 0 \quad \text{on } \Gamma_\sigma \tag{31}$$

$$\bar{\boldsymbol{u}}^{n+1} \cdot v = 0 \quad \text{on } \Gamma_u \tag{32}$$

Discrete equations

Using the approximations (22)–(24) the discrete equations become:

$$\text{(A)}\quad \int_V N_i N_i^{\mathrm{T}}\,\mathrm{d}V U_i^* = \int_V N_i N_i^{\mathrm{T}}\,\mathrm{d}V U_i^n + \Delta t \int_V N_i f_i\,\mathrm{d}V - \frac{\gamma \Delta t}{\rho}\int_V N_i \frac{\partial N_p^{\mathrm{T}}}{\partial x_i}\,\mathrm{d}V P^n - \frac{\Delta t \mu}{\rho}\int_V \frac{\partial N_i}{\partial x_i}\frac{\partial N_i^{\mathrm{T}}}{\partial x_i}\,\mathrm{d}V U_i^n + \frac{\Delta t}{\rho}\int_{\Gamma_\sigma} N_i \overline{\sigma_n^n}\,\mathrm{d}\Gamma \tag{33}$$

In compact form

$$\text{(A)}\quad M_u U^* = M_u U^n + \Delta t F - \frac{\gamma \Delta t}{\rho} B^{\mathrm{T}} P^n - \frac{\Delta t \mu}{\rho} K U^n \tag{34}$$

In the same way

$$\text{(B)}\quad \frac{1}{\Delta t^2}\left(\int_V N_p N_p^{\mathrm{T}}\,\mathrm{d}V \underline{\rho}^0 - \int_V N_p N_\rho^{\mathrm{T}}\,\mathrm{d}V \underline{\rho}^n \right) - \frac{\rho}{\Delta t}\int_V \left(\frac{\partial N_p}{\partial x_i} N_i^{\mathrm{T}} \right) \mathrm{d}V U^* + \frac{\rho}{\Delta t}\int_{\Gamma_u} N_p \bar{\boldsymbol{u}}_n^{n+1}\,\mathrm{d}\Gamma = -\int_V \left(\frac{\partial N_p}{\partial x_i}\frac{\partial N_p^{\mathrm{T}}}{\partial x_i} \right) \mathrm{d}V (P^{n+1} - \gamma P^n) \tag{35}$$

In compact form

$$\text{(B)}\quad -\frac{M_p(\underline{\rho}^0 - \underline{\rho}^n)}{\Delta t^2} + \frac{\rho}{\Delta t} B U^* - \frac{\rho}{\Delta t}\widehat{U} + S\gamma P^n = S P^{n+1} \tag{36}$$

and

$$\text{(C)}\quad \int_V N_i N_i^{\mathrm{T}}\,\mathrm{d}V U^{n+1} = \int_V N_i N_i^{\mathrm{T}}\,\mathrm{d}V U^{*} - \frac{\Delta t}{\rho}\int_V N_i \frac{\partial N_p^{\mathrm{T}}}{\partial x_i}\,\mathrm{d}V (P^{n+1}-\gamma P^{n}) + \int_{\Gamma_\sigma} N_i N_p^{\mathrm{T}}\,\mathrm{d}\Gamma (P^{n+1}-\gamma P^{n}) \tag{37}$$

In compact form (noting that $p=0$ on Γ_σ)

$$\text{(C)}\quad M_u U^{n+1} = M_u U^{*} - \frac{\Delta t}{\rho} B^{\mathrm{T}} (P^{n+1}-\gamma P^{n}) \tag{38}$$

where the matrices are

$$M = \begin{bmatrix} M_p & 0 & 0 \\ 0 & M_p & 0 \\ 0 & 0 & M_p \end{bmatrix} \tag{39}$$

$$M_p = \int_V N N^{\mathrm{T}}\,\mathrm{d}V \tag{40}$$

$$B = \left[\int_V \left(\frac{\partial N}{\partial x} N^{\mathrm{T}} \right) \mathrm{d}V; \int_V \left(\frac{\partial N}{\partial y} N^{\mathrm{T}} \right) \mathrm{d}V; \int_V \left(\frac{\partial N}{\partial z} N^{\mathrm{T}} \right) \mathrm{d}V \right] \tag{41}$$

$$S = \int_V \left(\frac{\partial N}{\partial x}\frac{\partial N^{\mathrm{T}}}{\partial x} + \frac{\partial N}{\partial y}\frac{\partial N^{\mathrm{T}}}{\partial y} + \frac{\partial N}{\partial z}\frac{\partial N^{\mathrm{T}}}{\partial z} \right) \mathrm{d}V \tag{42}$$

$$\widehat{U} = \int_{\Gamma_u} N \bar{\boldsymbol{u}}_n^{n+1}\,\mathrm{d}\Gamma \tag{43}$$

$$K = \begin{bmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & S \end{bmatrix} \tag{44}$$

$$F^{\mathrm{T}} = \left[\int_V N^{\mathrm{T}} f_x\,\mathrm{d}V;\quad \int_V N^{\mathrm{T}} f_y\,\mathrm{d}V;\quad \int_V N^{\mathrm{T}} f_z\,\mathrm{d}V \right] + \frac{1}{\rho}\left[\int_V N^{\mathrm{T}} \bar{\sigma}_{nx}\,\mathrm{d}V;\quad \int_V N^{\mathrm{T}} \bar{\sigma}_{ny}\,\mathrm{d}V;\quad \int_V N^{\mathrm{T}} \bar{\sigma}_{nz}\,\mathrm{d}V \right] \tag{45}$$

6. 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 parameter must be added in equations (34) and (38). Nevertheless, the incompressibility condition must be stabilized in equal-order approximations to avoid possible pressure oscillations.

Then, Eq. (36) must be stabilized if smooth pressure results are important. It must be noted than pressure oscillations do not influence significantly in the velocity results. Nevertheless, in most physical problems, pressure is the main result to be obtained. That is why stabilization of Eq. (36) must be performed.

The so-called *finite calculus* (FIC) formulation [20–22] will be chosen here as the stabilization procedure. This formulation is based in the modification of the governing differential equations of the problem by accepting that the domain where the balance laws are established (balance of momentum and balance of mass) has a finite size. The modified equations in the FIC formulation for incompressible fluids are

momentum

$$r_i - \underline{\frac{h_k}{2}\frac{\partial r_i}{\partial x_k}} = 0 \tag{46}$$

mass conservation

$$r - \underline{\frac{h_k}{2}\frac{\partial r}{\partial x_k}} = 0 \tag{47}$$

where from Eqs. (1) and (2) the residuals are defined by

$$r_i = \rho\frac{\mathrm{D}u_i}{\mathrm{D}t} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} - \rho f_i \tag{48}$$

$$r = \frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho\frac{\partial u_i}{\partial x_i} \tag{49}$$

with $i,k = 1, n_d$ where n_d are the space dimensions of the problem.

Eqs. (46) and (47) are completed with the boundary and initial conditions. Note that for consistency, the Neumann boundary condition on Γ_σ must also be adequately modified by adding a residual term. The details can be found in [21].

The underlined terms in Eqs. (46) and (47) introduce the necessary stabilization in the numerical solution using whatever discretization method. Examples of the application of the FIC approach the convection–diffusion problems and incompressible problems in solids and fluid mechanics are presented in [21,22].

Distances h_i in Eqs. (46) and (47) are "characteristic length" parameters and their values control the relevance of the stabilization terms. The computation of the characteristic lengths is a critical issue in the stabilization process [20].

The new terms in the momentum and mass conservation equations stabilize the numerical solution in presence of high values of the convective terms and incompressibility zones, respectively. Obviously, in Lagrangian flows, as in incompressible solid mechanics problems, the relevant stabilization term is that of Eq. (47), as the convective terms are zero in the momentum equations.

For the practical application of the FIC formulation the stabilization term in the mass balance equation is expressed as a function of the residual of the momentum equations using Eq. (46) as

$$\frac{h_k}{2}\frac{\partial r}{\partial x_k} \cong \sum_{i=1}^{n_d} \tau_i \frac{\partial r_i}{\partial x_i} \tag{50}$$

where τ_i are *intrinsic time parameters* given by

$$\tau_i = \frac{3h_i^2}{8\mu} \tag{51}$$

The modified incompressibility equation is therefore written for the numerical computations as

$$r - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_i}{\partial x_i} = 0 \tag{52}$$

The stabilization terms in the momentum Eq. (46) are dropped here onwards for the numerical solution.

It is convenient to rewrite the residual r_i in Eq. (48) as

$$r_i = \frac{\partial p}{\partial x_i} + \pi_i \tag{53}$$

where π_i are pressure gradient projection terms. These terms are considered *as additional nodal variables*. The necessary additional equations to match the increase in the number of unknowns are obtained by expressing that the residual r_i as defined by Eq. (48), vanishes, in the average sense, over each element. This can be expressed in weighted integral form as

$$\int_V w_i \left(\frac{\partial p}{\partial x_i} + \pi_i \right) \mathrm{d}V = 0 \tag{54}$$

where w_i are appropriate weighting functions.

Discretization of the π_i terms using the same MFEM interpolation functions gives

$$\pi_i = N_i^{\mathrm{T}} \Pi \tag{55}$$

where Π represents the local value of the three components of the pressure gradient. Eq. (54) leads to an equation system of the form (for $w_i = N_i$)

$$M\Pi + B^{\mathrm{T}} P = 0 \tag{56}$$

Eq. (21) is now modified with the new stabilization term as

$$\frac{\rho^0 - \rho^n}{\Delta t^2} + \frac{\rho}{\Delta t}\frac{\partial u_i^*}{\partial x_i} = \frac{\partial^2}{\partial x_i^2}(p^{n+1} - \gamma p^n) + \sum_{i=1}^{n_d} \frac{\tau_i}{\Delta t}\frac{\partial r_i}{\partial x_i} \tag{57}$$

and Eq. (26) becomes now

$$\int_V N_p \left\{ \frac{\rho^0 - \rho^n}{\Delta t^2} + \frac{\rho}{\Delta t}\frac{\partial}{\partial x_i} u_i^* - \frac{\partial^2}{\partial x_i^2}(p^{n+1} - \gamma p^n) - \sum_{i=1}^{n_d} \frac{\tau_i}{\Delta t}\frac{\partial r_i}{\partial x_i} \right\} \mathrm{d}V + \text{boundary terms} \tag{58}$$

Integrating by parts, the equivalent to Eq. (29) is

$$\frac{1}{\Delta t^2}\int_V N_p(\rho^0 - \rho^n)\,\mathrm{d}V - \int_V \frac{\partial N_p}{\partial x_i}\left\{ \frac{\rho}{\Delta t} u_i^* - \frac{\partial (p^{n+1} - \gamma p^n)}{\partial x_i} - \sum_{i=1}^{n_d} \frac{\tau_i}{\Delta t}\left(\frac{\partial p^{n+1}}{\partial x_i} + \pi_i^{n+1} \right) \right\} \mathrm{d}V + \text{b.t.} = 0 \tag{59}$$

Introducing the discretization of the different fields, and using a compact notation gives

$$-\frac{M_p(\underline{\rho}^0 - \underline{\rho}^n)}{\Delta t^2} + \frac{\rho}{\Delta t} B U^* - \frac{\rho}{\Delta t}\widehat{U} + S\gamma P^n - B_\tau \Pi^n = (S + S_\tau) P^{n+1} \tag{60}$$

where the new stabilization matrices B_τ and S_τ are defined by

$$B_\tau = \left[\int_V \left(\frac{\partial N}{\partial x} N^{\mathrm{T}} \right) \mathrm{d}V \frac{\tau_x}{\Delta t}; \int_V \left(\frac{\partial N}{\partial y} N^{\mathrm{T}} \right) \mathrm{d}V \frac{\tau_y}{\Delta t}; \int_V \left(\frac{\partial N}{\partial z} N^{\mathrm{T}} \right) \mathrm{d}V \frac{\tau_z}{\Delta t} \right] \tag{61}$$

$$S_\tau = \int_V \left(\frac{\partial N}{\partial x}\frac{\partial N^{\mathrm{T}}}{\partial x}\frac{\tau_x}{\Delta t} + \frac{\partial N}{\partial y}\frac{\partial N^{\mathrm{T}}}{\partial y}\frac{\tau_y}{\Delta t} + \frac{\partial N}{\partial z}\frac{\partial N^{\mathrm{T}}}{\partial z}\frac{\tau_z}{\Delta t} \right) \mathrm{d}V \tag{62}$$

Note that the effect of the stabilization terms is the addition of a new Laplacian matrix S_τ and a new term in the r.h.s. of Eq. (60) depending on the pressure gradient projection variables π_i.

The pressure gradient projection may be evaluated explicitly using Eq. (56) by

$$\text{(D)} \quad \Pi^{n+1} = -M^{-1} B^{\mathrm{T}} P^{n+1} \tag{63}$$

The three steps (A)–(C) described before are now completed with a *fourth step* (D) where the lumped diagonal form of matrix M may be used.

7. Mass conservation

In a Lagrangian formulation a new mesh is generated at each time step, and all the information is transmitted with the nodes or particles. In that way, a local variation in the volume associated with the particles is used as the correct volume in the next time step. A permanent update of the initial volume is necessary to avoid large error accumulation.

Thus, the correct evaluation of the first term of Eq. (36) becomes important in a Lagrangian formulation and will be discussed below.

The term

$$M_p(\underline{\rho}^0 - \underline{\rho}^n) \tag{64}$$

may be evaluated in two different ways.

(I) Evaluation via a density update

From the mass conservation equation, the density at time t^n may be computed as

$$\rho^n = \rho^{n-1} - \rho\Delta t \frac{\partial u_i^n}{\partial x_i} \tag{65}$$

Making use of the spatial discretization (22) and (24) and the Galerkin residual method gives

$$\int_V N_p N_p^{\mathrm{T}}\,\mathrm{d}V\underline{\rho}^n = \int_V N_p N_p^{\mathrm{T}}\,\mathrm{d}V\underline{\rho}^{n-1} - \rho\Delta t\int_V N_p\frac{\partial N_i^{\mathrm{T}}}{\partial x_i}\,\mathrm{d}VU^n \tag{66}$$

Integrating by parts the last term

$$\int_V N_p N_p^{\mathrm{T}}\,\mathrm{d}V\underline{\rho}^n = \int_V N_p N_p^{\mathrm{T}}\,\mathrm{d}V\underline{\rho}^{n-1} + \rho\Delta t\int_V \frac{\partial N_p^{\mathrm{T}}}{\partial x_i}N_i^{\mathrm{T}}\,\mathrm{d}VU^n - \rho\Delta t\int_{\Gamma_u} N_p\overline{u_n^n}\,\mathrm{d}\Gamma \tag{67}$$

or in compact notation

$$M_p\underline{\rho}^n = M_p\underline{\rho}^{n-1} + \rho\Delta t\,B\,U^n - \rho\Delta t\hat{U} \tag{68}$$

In order to take into account that the shape functions N are different at each mesh update the following notation will be used: the shape functions or the matrices evaluated at the time t^n will be noted by N_p^n and M_p^n. Then Eq. (68) becomes

$$M_p^n\underline{\rho^n} = M_p^n\underline{\rho}^{n-1} + \rho\Delta^n t\,B^n U^n - \rho\Delta t\hat{U}^n \tag{69}$$

where $\Delta^n t$ represents the time incremental time t^n.

Then

$$\underline{\rho^n} = \underline{\rho}^{n-1} + \rho\Delta^n t(M_p^n)^{-1}B^n U^n - \rho\Delta t\hat{U}^n = \underline{\rho}^{n-1} + \underline{\Delta^n\rho} \tag{70}$$

where the density variation has been defined by

$$\underline{\Delta^n\rho} = \rho\Delta^n t(M_p^n)^{-1}B^n U^n - \rho\Delta t\hat{U}^n \tag{71}$$

representing the ρ variation at time t^n.

Successive application of Eq. (70) for all time steps gives:

$$\underline{\rho^n} = \underline{\rho}^0 + \sum_{l=1}^{n}\underline{\Delta^l\rho} = \underline{\rho}^0 + \rho\sum_{l=1}^{n}\left\{\Delta^l t(M_p^l)^{-1}B^l U^l - \Delta^l t\hat{U}^l\right\} \tag{72}$$

The term $M_p(\underline{\rho}^0 - \underline{\rho}^n)$ of the r.h.s. of Eq. (36) can be written as

$$M_p(\underline{\rho}^0 - \underline{\rho}^n) = -M_p\sum_{l=1}^{n}\underline{\Delta^l\rho} \tag{73}$$

This means that at each time step t^l, the vector

$$\underline{\Delta^l\rho} = \rho\left\{\Delta^l t(M_p^l)^{-1}B^l U^l - \Delta^l t\hat{U}^l\right\} \tag{74}$$

must be evaluated, added to the previous one and stored for the next time step.

(II) Evaluation via the initial associated volume

mass conservation implies

$$\int_{V_{(t=0)}}\rho^0\,\mathrm{d}V = \int_{V_{(t=n)}}\rho^n\,\mathrm{d}V \tag{75}$$

Using the shape functions at the corresponding time step

$$\int_{V_{(t=0)}}(N_\rho^0)^{\mathrm{T}}\,\mathrm{d}V\underline{\rho}^0 = \int_{V_{(t=n)}}(N_\rho^n)^{\mathrm{T}}\,\mathrm{d}V\underline{\rho}^n \tag{76}$$

Defining the volume associated to each particle by

$$(\Omega^n)^{\mathrm{T}} = \int_{V_{(t=n)}}(N_\rho^n)^{\mathrm{T}}\,\mathrm{d}V \tag{77}$$

Eq. (76) becomes

$$(\Omega^0)^{\mathrm{T}}\underline{\rho}^0 = (\Omega^n)^{\mathrm{T}}\underline{\rho}^n \tag{78}$$

which has the meaning of the total mass conservation. Vector Ω^n may be considered as the vector containing the volumes associated to each particle. It may be calculated using (77) or using the Voronoï diagram of the node distribution.

The concept of local mass conservation may be used next. This means that each particle (node) conserves his own local mass, i.e.

$$\Omega_i^0\rho_i^0 = \Omega_i^n\rho_i^n \tag{79}$$

The term $M_p(\underline{\rho}^0 - \underline{\rho}^n)$ may be written as

$$\underline{\underline{\Omega}}^n(\underline{\rho}^0 - \underline{\rho}^n) = \underline{\underline{\Omega}}^n\underline{\rho}^0 - \underline{\underline{\Omega}}^0\underline{\rho}^0 = \rho^0(\underline{\underline{\Omega}}^n - \underline{\underline{\Omega}}^0) \tag{80}$$

where $\underline{\underline{\Omega}}^0$ and $\underline{\underline{\Omega}}^n$ represent a diagonal matrix with the volume associated to each particle at time $t = t^0$ and $t = t^n$, respectively.

These matrices may be evaluated using the lumped matrices M_ρ^0 and M_ρ^n or directly using the associated volume to each particle obtained from a Voronoï diagram.

8. Boundary surfaces

One of the main problems in mesh generation is the correct definition of the boundary domain. Sometimes, boundary nodes are explicitly defined as special nodes, which are different from internal nodes. In other cases, the total set of nodes is the only information available and the algorithm must recognize the boundary nodes. Such is the case in the Lagrangian formulation in which, at each time step, a new node distribution is obtained

and the boundary-surface must be recognized from the node positions.

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

Considering that the node follows 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 $r(x)$ bigger than $\alpha h(x)$, are considered as boundary nodes.

Thus, α is a parameter close to, but greater than one. Note that this criterion is coincident with the alpha-shape concept [13].

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, a concave one and convex one.

In this work, the boundary surface is defined with all the polyhedral surfaces having all their nodes on the boundary and belonging to just one polyhedron. See Ref. [12].

The correct boundary surface may be important to define the correct normal external to the surface. Furthermore; in weak forms (Galerkin) a correct evaluation of the volume domain is also important. Nevertheless, it must 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.

9. Application to fluid–structure interactions

The fluid described above will interact with structures that are in contact with it. Three different cases of

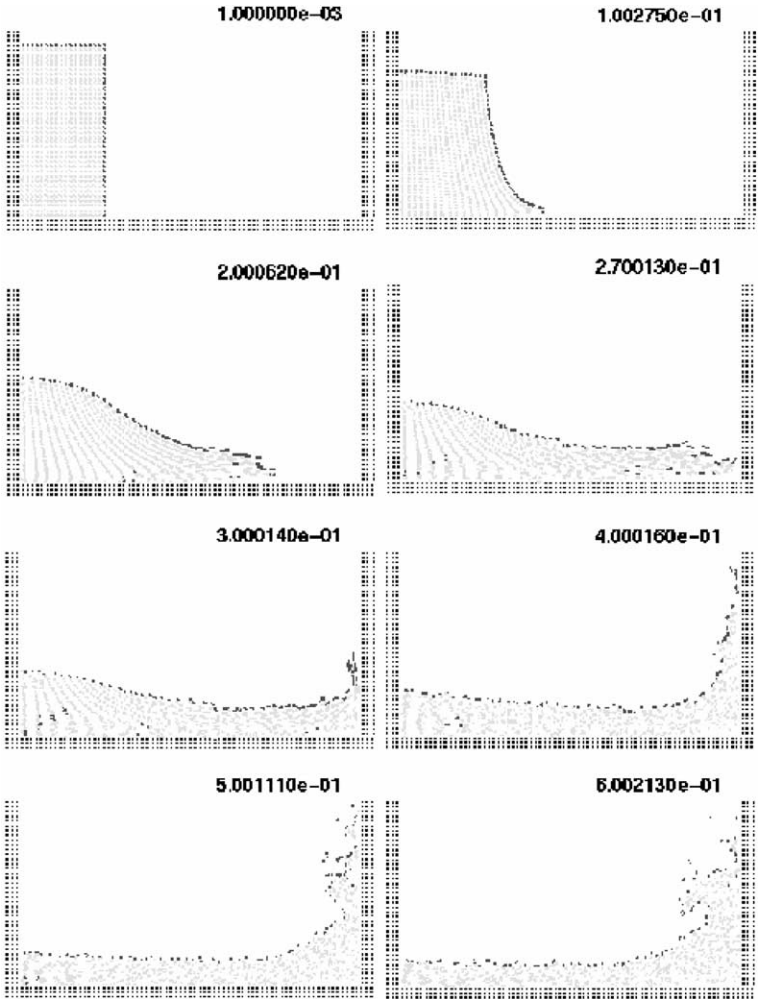

Fig. 1. Water column collapse at different time steps.

structures will be analyzed. In all three cases, the elastic strains will be neglected and only rigid solid motions will be considered.

9.1. Fixed structures

The first type of examples presented is structures in which there is a fixed wall, for instance, the recipient in which the fluid is contained. See Figs. 1 and 2.

This kind of structures will be analyzed by adding fixed particles at the boundaring with velocity $u_i = 0$. These particles will be included in the computation of equations (A) and (B) as standard nodes, but during equation (C) the velocity will be fixed to zero.

The inclusion of fixed boundary particles is very important to avoid contact problems. These fixed particles automatically force the fluid to remain inside a recipient. The moving particles cannot go across the wall due to the incompressibility condition and not to any other restriction of velocity or displacement. This condition solves the contact problems with complicated curved structures. See for instance example 2.

9.2. Moving structures with a known velocity

The second type of FSI is between the fluid and a moving wall of known velocity as a function of the time. This is the case of moving recipients, moving mills, or moving ships with prescribed velocity.

In this case, moving particles with known velocity are introduced in the domain boundaries. Note that the term

$$\frac{\rho}{\Delta t}\widehat{U} \tag{81}$$

with

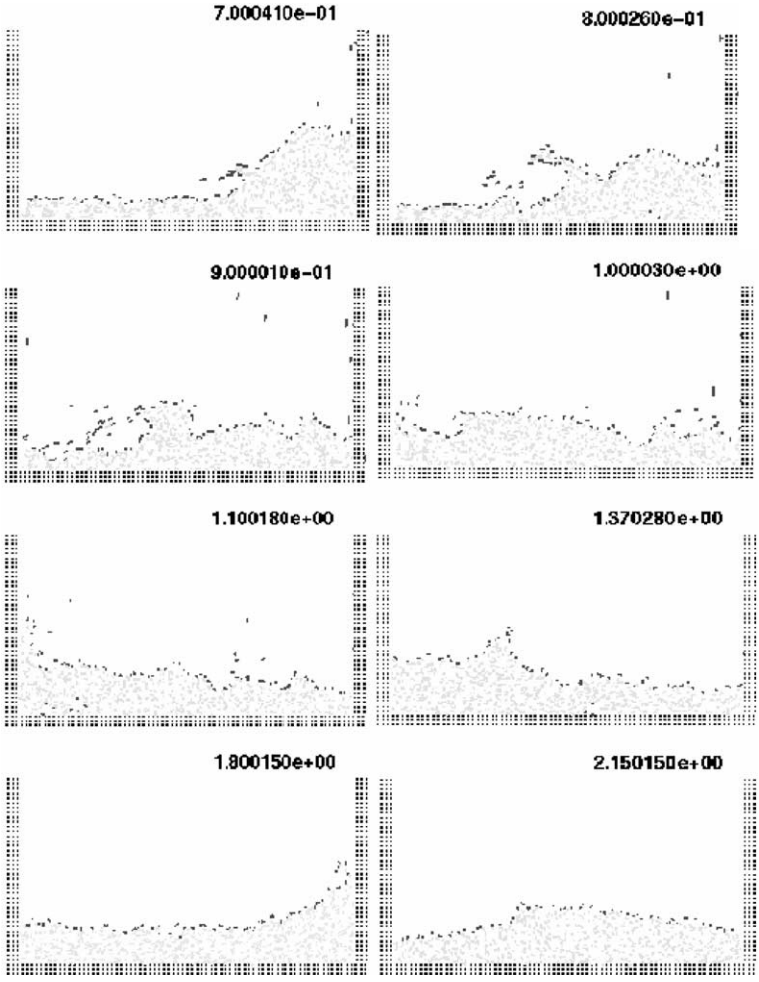

Fig. 1 (*continued*)

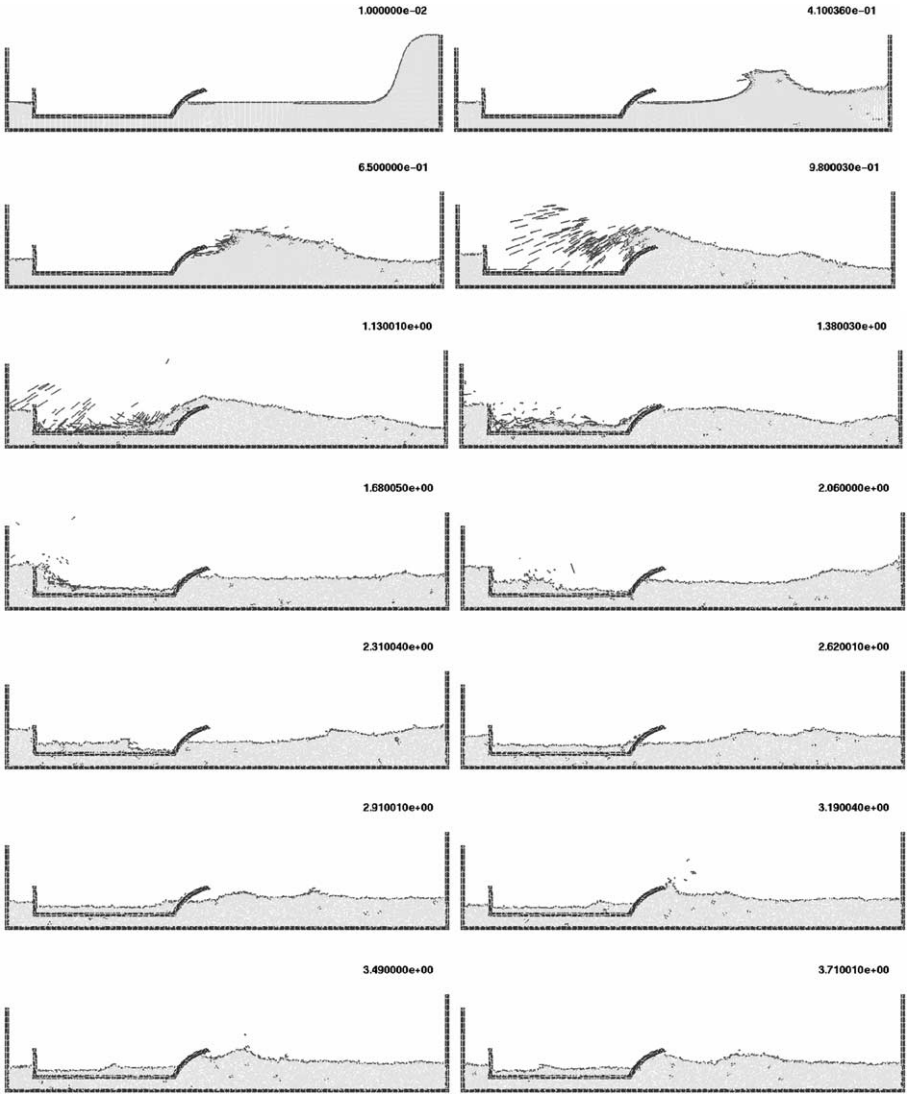

Fig. 2. Fixed ship under external waves.

$$\widehat{U} = \int_{\Gamma_u} N\bar{\boldsymbol{u}}_n^{n+1}\,\mathrm{d}\Gamma \tag{82}$$

must be added in equation (C) where $\bar{\boldsymbol{u}}_n^{n+1}$ are the known velocity on the boundaries. See for instance Figs. 4 and 5.

9.3. Moving structures

Finally, the case of moving rigid structures is considered. For instance, the case of a floating ship (see keeping). In this case, the solid will be considered as a domain with a high viscosity parameter, much higher than the fluid domain. For practical problems a value of $10^4\mu$ is enough to represent a solid without introducing numerical problems (see Figs. 5 and 6).

10. Numerical test

10.1. Water column collapse

This problem was solved by Koshizu and Oka [4] both experimentally and numerically. 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.

Fig. 1 shows the point positions at different time steps. The dark points represent the free-surface detected with the alpha-shape algorithm with an alpha parameter $\alpha = 1.1$. The internal points are gray and the fixed points are black.

The water is running on the bottom wall until, near 0.3 s, it impinges on the right vertical wall. Breaking waves appear at 0.6 s. Around 1 s, the water reaches the left wall. Agreement with the experimental results of Ref. [4] both in the shape of the free surface as well as in the time development are excellent.

10.2. Fixed ship under external waves

This example is a very schematic representation of a ship when it is hit by an external wave (Fig. 2). The ship cannot move and initially the free surface is horizontal with a rectangle on the right wall to produce a big wave. Fixed nodes represent the ship as well as the wall recipient.

The example was created in order to test the suitability of the method to solve contact problems with curved walls correctly. It is interesting to see the crash of the waves under the ship prow and the rebound of the wave on 3.15 s. It is also interesting to see the different

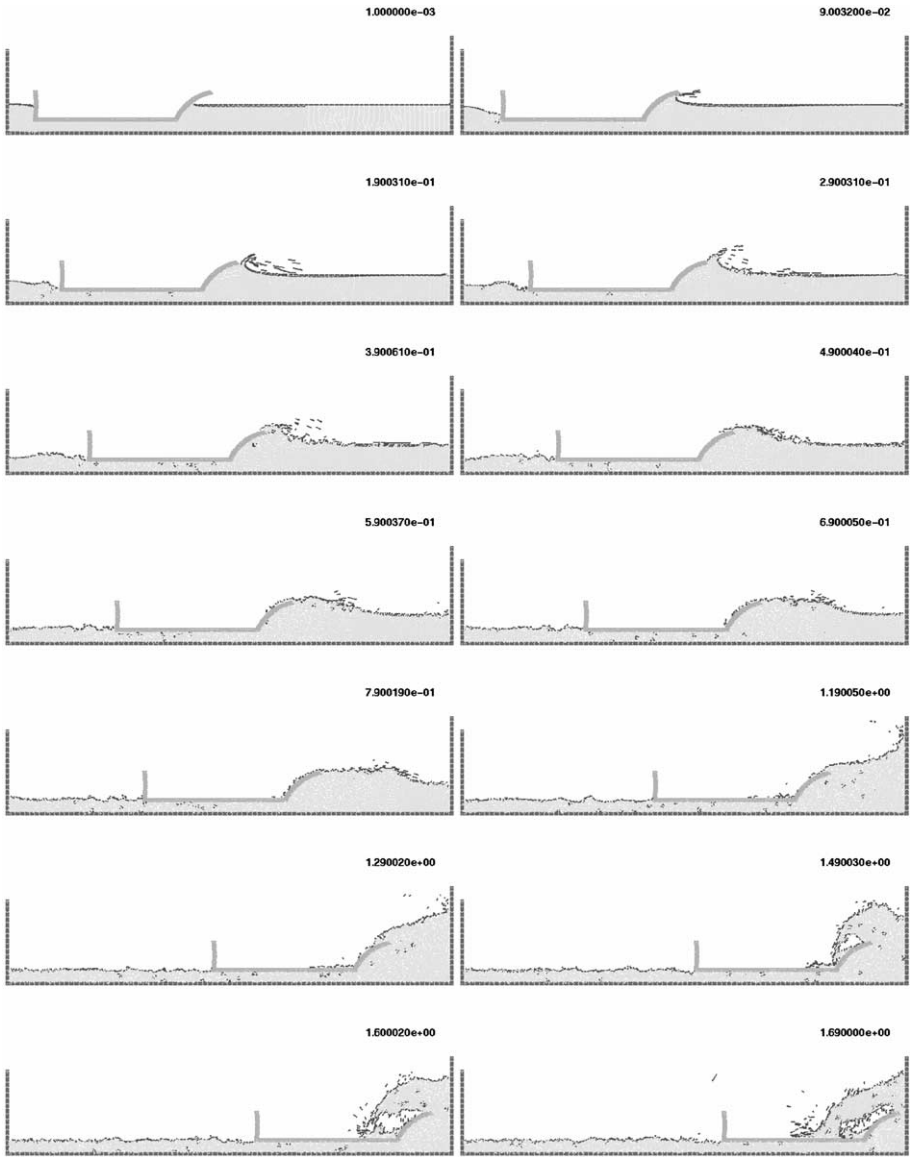

Fig. 3. Moving ship with known velocity.

contact walls with the internal and external ship surfaces and the moving free surface at the bottom of the ship.

10.3. Moving ship with known velocity

In this case (Fig. 3), the same ship of the previous example is now moving at a fixed velocity. All the nodes representing the ship have an imposed velocity. The free surface, which was initially horizontal, takes a correct position at the bottom of the ship, and again, the correct contact problem is realistically solved in the curved prow.

10.4. Rotating water mill

A schematic representation of a water mill is presented in Fig. 4. The blades of the mill have an imposed rotating velocity, while the water is initially in a stationary and flat position. Fluid structure interactions with free-surfaces and fragmentation are well reproduced in this example.

10.5. Solid falling into a recipient with water

In this example the fluid is interacting with a solid that is totally free, without any imposed velocity. Fig. 5 represents a free cube falling down into a recipient full of water. The solid cube was modeled by introducing a high viscosity parameter in the element in the following way: all the polyhedral elements formed by nodes contained in the solid have a high viscosity value. The other elements are inviscid.

The example represents correctly the contact problem when the cube hits the water and also the different speed during the falling process.

10.6. Solid floating on a free surface

The last example of Fig. 6 represents a very interesting problem of fluid structure interaction when there is a weak interaction between the fluid and a large rigid deformation of the structure. In this case, there is also a free-surface problem, representing a schematic case of see-keeping in ship hydrodynamics.

The example shows an initially stationary recipient with a floating piece of wood in which a wave is produced on the left side. The wave intercepts the wood piece producing a breaking wave and moving the floating wood.

All the previous examples are only schematic representations of real problems. Only the first example has an experimental reference. The rest are presented here in order to evaluate the suitability of the method to solve problems other methods have difficulties to solve.

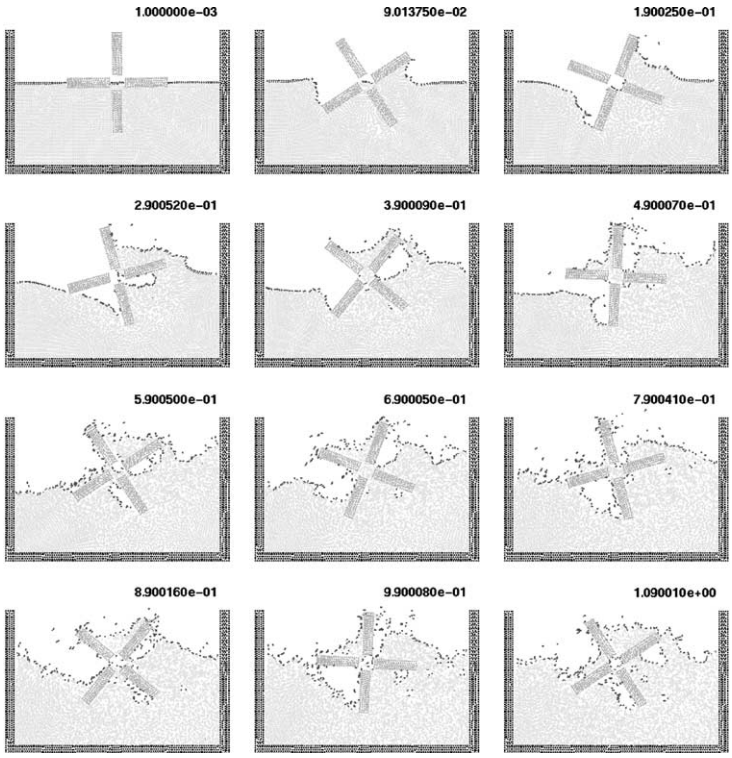

Fig. 4. Rotating water mill.

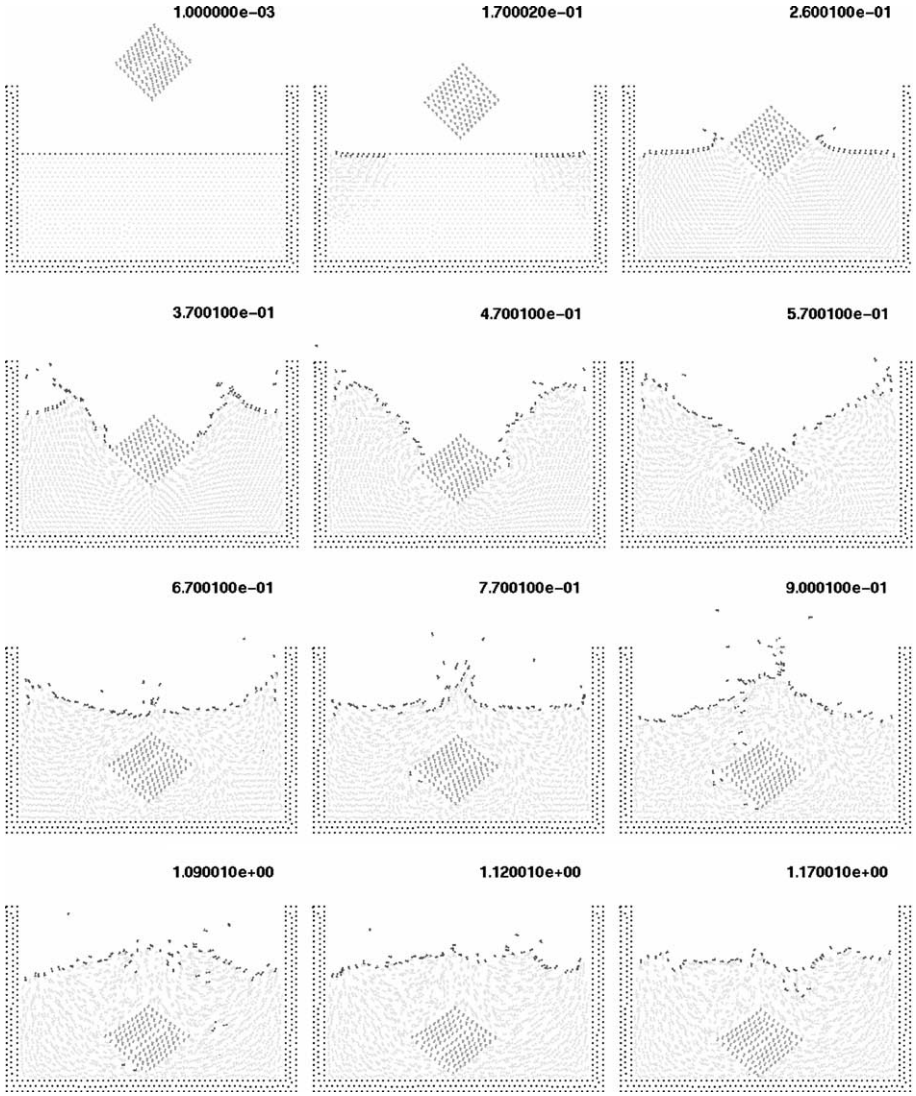

Fig. 5. Solid cube falling into a recipient with water.

11. Conclusions

Lagrangian formulation and the MFEM are an excellent combination to solve fluid mechanic problems, especially FSI with moving free-surface and contact problems.

Breaking waves, collapse problems, and contact problems can be solved easily without any additional constraint.

Furthermore, the MFEM presented, as opposed to other methods, has the advantages of a good meshless method concerning the easy introduction of the nodes connectivity in a bounded time of order n. The method proposed also shares some advantages with the FEM such as: (a) the simplicity of the shape functions, (b) C_0 continuity between elements, (c) an easy introduction of the boundary conditions, and (d) symmetric matrices.

The FIC formulation can be successfully used in a Lagrangian formulation in order to eliminate spurious pressure oscillations.

Both the Lagrangian formulation and the MFEM are the key ingredients to solve FSI problems including with free-surface, breaking waves and collapse situations.

Appendix A

All the shape functions N_i described in this paper are based on the MFEM. A full description of the MFEM may be found in Ref. [12]. Nevertheless and for the sake of completeness a summary is presented in this appendix.

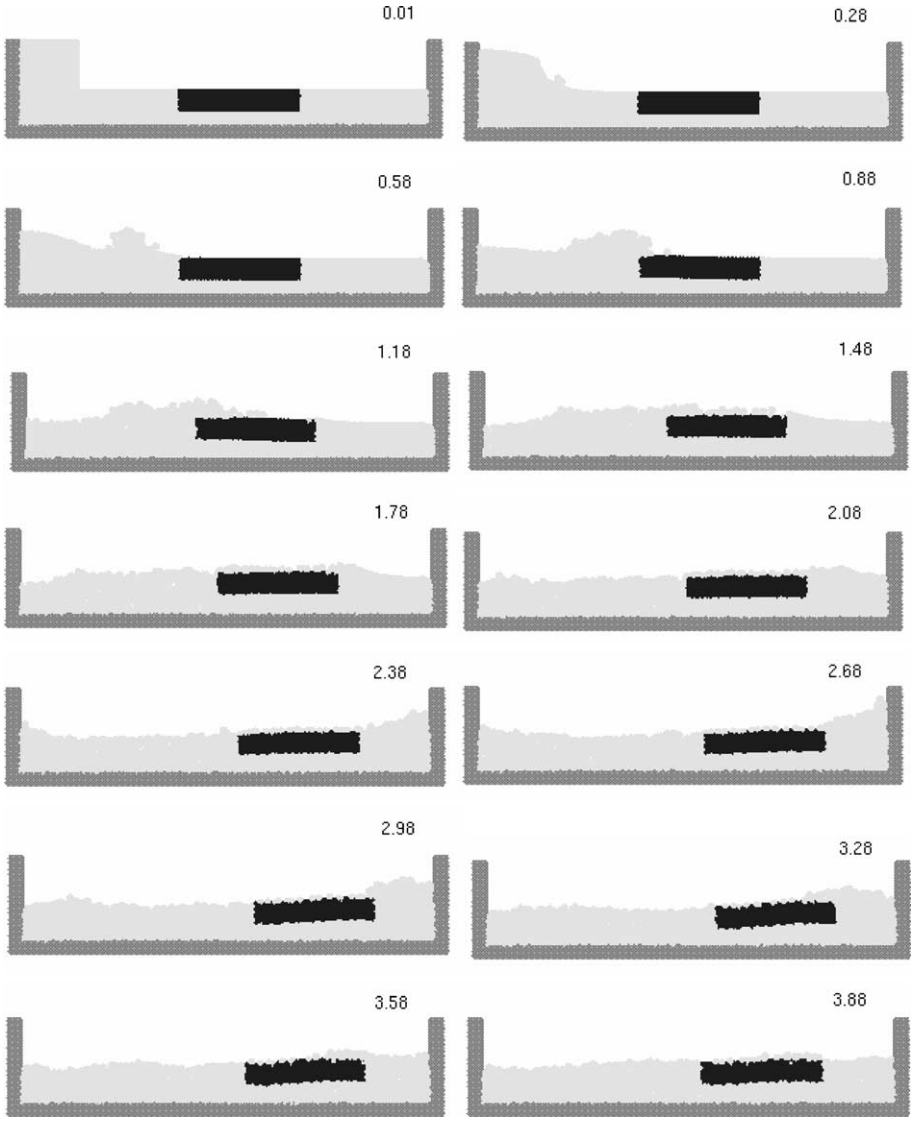

Fig. 6. Solid floating on a free surface.

The MFEM combines a particular finite element subdivision in polyhedral shape called the EDT and ad hoc shape functions for this kind of polyhedra.

A.1. The extended Delaunay tessellation (EDT)

Let a set of distinct nodes be: $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \ldots, \mathbf{n}_n\}$ in R^3.

(a) *The Voronoï diagram* of the set **N** is a partition of R^3 into regions V_i (closed and convex, or unbounded), where each region V_i is associated with a node $\mathbf{n}_i$, such that any point in V_i is closer to $\mathbf{n}_i$ (nearest neighbor) than to any other node $\mathbf{n}_i$. See Fig. 7 for a 2-D representation. There is a single Voronoï diagram for each set **N**.

(b) *A Voronoï sphere* within the set **N** is any sphere, defined by four or more nodes, that contains no other node inside. Such spheres are also known as empty circumspheres.

(c) *A Delaunay tessellation* within the set **N** is a partition of the convex hull of all the nodes into regions Ω_i such that $\Omega = U\Omega_i$, where each Ω_i is the tetrahedron defined by four nodes of the same Voronoï sphere. Delaunay tessellations of a set **N** are not unique, but each tessellation is the dual of the single Voronoï diagram of the set.

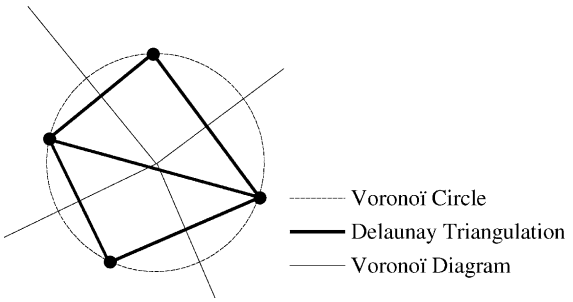

Fig. 7. Voronoï diagram, Voronoï circle and Delaunay triangulation for a four nodes distribution in 2-D.

The computing time required for evaluation of all these three entities is of order n^{α} with $\alpha \leqslant 1.333$. Using a very simple bin organization, the computation time may be reduced to a simple order n.

As stated above, the Delaunay tessellation of a set of nodes is non-unique. For the same node distribution, different triangulations (actually tetrahedrations, as it refers to 3-D) are possible. Therefore, an interpolation based on the Delaunay tessellation is sensitive to geometric perturbations of the position of the nodes. On the other hand, its dual, the Voronoï diagram, is unique. Thus, it makes more sense to define meshless shape functions based on the unique Voronoï diagram than on Delaunay tessellations. Furthermore, in 3-D problems the Delaunay tessellation may generate several tetrahedra of zero or almost zero volume, which introduces large inaccuracies into the shape function derivatives. The time to obtain a good mesh via a Delaunay tessellation becomes then an unbounded iterative operation.

These drawbacks appear in the so-called "degenerated case", which is the case where more than four nodes (or more than three nodes in a 2-D problem) are on the same empty sphere. For instance, when five nodes are on the same sphere, five tetrahedra may be defined satisfying the Delaunay criterion, but some of them may have zero or almost zero volumes, called *slivers*, as seen in Fig. 8:

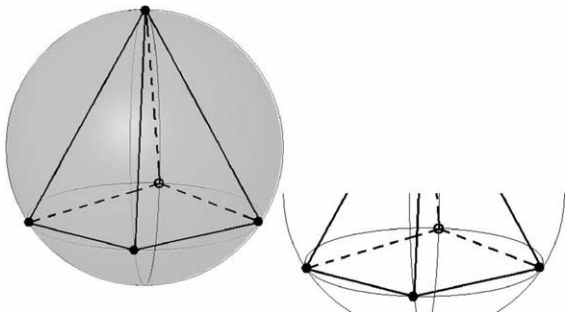

Fig. 8. Five nodes on the same sphere and possible zero or almost zero volume tetrahedron (sliver) on the right.

In order to overcome above drawbacks, a generalization of the Delaunay tessellation will be defined.

Definition. The extended Delaunay tessellation within the set **N** is the unique partition of the convex hull Ω of all the nodes into regions Ω_i such that $\Omega = U\Omega_i$, where each Ω_i is the polyhedron defined by all the nodes laying on the same Voronoï sphere.

The main difference between the traditional Delaunay tessellation and the EDT is that, in the latter, all the nodes belonging to the same Voronoï sphere define a unique polyhedron. With this definition, the domain Ω is divided into tetrahedra and other polyhedra, which are unique for a set of node distributions. Fig. 9 for instance, is a 2-D polygon partition with a triangle, a quadrangle and a pentagon. Fig. 10 is a classical eight-node polyhedron with all the nodes on the same sphere.

For non-uniform node distributions, considering infinite precision, only four nodes are necessary to define a sphere. Other nodes close to the sphere may define other spheres very close to the previous one. In order to avoid this situation, which may hide polyhedra with more than four nodes, a parameter δ will be introduced. In such a way, the polyhedra are defined by all the nodes of the same sphere and nearby spheres with a distance between center points smaller than δ.

The parameter δ avoids generating zero volume or near zero volume tetrahedra. When δ is large, the number of polyhedra with more than four nodes

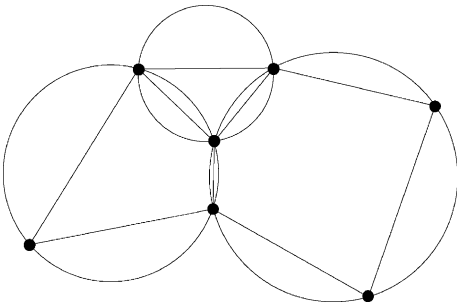

Fig. 9. Two-dimensional partition in polygons. The triangle, the quadrangle and the pentagon are each inscribed on a circle.

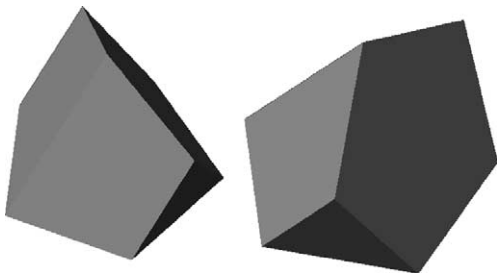

Fig. 10. Eight-node polyhedron. All nodes are on the same sphere.

increases, and the number of tetrahedra with near zero volume decreases, and vice versa.

The EDT leads to a domain partition which (a) is unique for a set of node distributions; (b) is formed by polyhedra with no zero volume, and (c) is obtained in a bounded time of order n. Then, it satisfies the conditions for a meshless method.

A.2. The meshless finite element shape functions

Once the domain partition in polyhedra is defined, shape functions must be introduced to solve a discrete problem. In fluid and solid mechanics problems typically, C^0 continuous shape functions are chosen. If possible, shape functions must be locally supported in order to obtain band matrices. They must also satisfy two criteria in order to have a reasonable convergence order, namely partition of unity and linear completeness.

In order to define the shape functions inside each polyhedron the non-Sibsonian interpolation is used [23].

Let $\mathbf{P} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_m\}$ be the set of nodes belonging to a polyhedron. The shape function $N_i(\mathbf{x})$ corresponding to the node $\mathbf{n}_i$ at an internal point $\mathbf{x}$ is defined by building first the Voronoï cell corresponding to $\mathbf{x}$ in the tessellation of the set $\mathbf{P} \cup \{\mathbf{x}\}$ and then by computing

$$N_i(\mathbf{x}) = \frac{\dfrac{s_i(\mathbf{x})}{h_i(\mathbf{x})}}{\displaystyle\sum_{j=1}^{m} \frac{s_j(\mathbf{x})}{h_j(\mathbf{x})}} \tag{A.1}$$

where $s_i(\mathbf{x})$ is the surface of the Voronoï cell face corresponding to node the node $\mathbf{n}_i$ and $h_i(\mathbf{x})$ is the distance between point $\mathbf{x}$ and the node $\mathbf{n}_i$ (Fig. 11).

Non-Sibsonian interpolations have the following properties [19].

(1)

$$0 \leqslant N_i(\mathbf{x}) \leqslant 1 \tag{A.2}$$

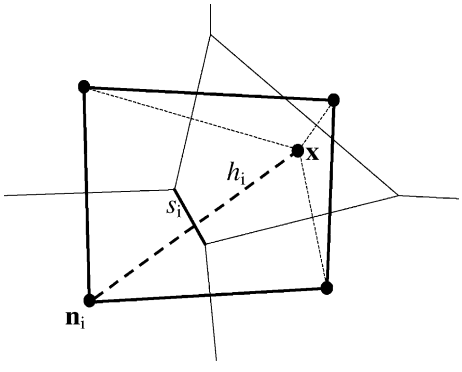

Fig. 11. Four nodes and arbitrary internal point x Voronoï diagram. Shape function parameters.

(2)

$$\Sigma_i N_i(\mathbf{x}) = 1 \tag{A.3}$$

(3)

$$N_i(\mathbf{n}_j) = \delta_{ij} \tag{A.4}$$

(4)

$$\mathbf{x} = \Sigma_i N_i(\mathbf{x}) \mathbf{n}_i \tag{A.5}$$

Furthermore, the particular definition of the non-Sibsonian shape function for the limited set of nodes on the same Voronoï sphere, adds the following properties.

(5) On a polyhedron surface, the shape functions depend only on the nodes of this surface.
(6) On triangular surfaces (or in all the polygon boundaries in 2-D), the shape functions are linear.
(7) If the polyhedron is a tetrahedron (or a triangle in 2-D) the shape functions are the linear finite element shape functions.
(8) Due to property 5, the shape functions have C^0 continuity between two neighboring polyhedra. See Fig. 12.
(9) As a matter of fact, because all the element nodes are on the same sphere, the evaluation of the shape functions and its derivatives becomes very simple.

The method MFEM defined here is both a meshless method and a FEM. The algorithm steps for the MFEM are

(1) for a set of nodes, compute all the empty spheres with four nodes;
(2) generate all the polyhedral elements using the nodes belonging to each sphere and the nodes of all the coincident and nearby spheres;
(3) calculate the shape functions and their derivatives, using the non-Sibsonian interpolation, at all the Gauss points necessary to evaluate the integrals of the weak form;

The MFEM is a truly meshless method because the shape functions depend only on the node positions. Furthermore, steps 1 and 2 of the node connectivity process are bounded with $n^{1.33}$, avoiding the mesh "cosmetics" often needed in mesh generators.

The number of Gauss points necessary to compute the element integrals depends, to a great extent, on the polyhedral shape of each element. Note that for an irregular node distribution, there remains a significant amount of tetrahedra (in the examples, more than 85% of the elements remains tetrahedral) with linear shape functions, for which only one Gauss point is enough. For the remaining polyhedra, the integrals are performed dividing them into tetrahedra and then using a single Gauss point in each tetrahedron. This subdivision

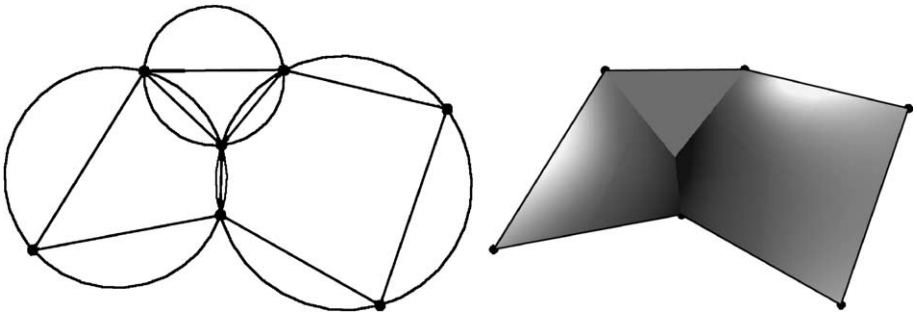

Fig. 12. C^0 continuity of the shape function on a 2-D node connection.

is only performed for the evaluation of the integrals and cannot be considered as a tetrahedral mesh because it is not conforming. The use of one Gauss point on each tetrahedron guarantee that the computing time in the evaluation of the matrices requires the same effort than the FEM.

References

[1] Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics, theory and application to non-spherical stars. Mon Nat R Astr Soc 1997;181:375–89.

[2] Bonet J, Kulasegaram S. Convection and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulation. Int J Numer Meth Eng 1999.

[3] Dilts GA. Moving least squares particle hydrodynamics. i. Consistency and stability. Int J Numer Meth Eng 1999;44:1115–55.

[4] Koshizuka S, Oka Y. Moving particle semi-implicit method for fragmentation of incompressible fluid. Nucl Eng Sci 1996;123:421–34.

[5] Belytschko T, Liu Y, Gu L. Element free Galerkin methods. Int J Numer Meth Eng 1994;37:229–56.

[6] De S, Bathe KJ. The method of finite spheres with improved numerical integration. Comput Struct 2001;79:2183–96.

[7] Nayroles B, Touzot G, Villon P. Generalizing the fem: diffuse approximation and diffuse elements. Computat Mech 1992;10:307–18.

[8] Oñate E, Idelsohn SR, Zienkievicz OC, Taylor RL. A finite point method in computational mechanics. Applications to convective transport and fluid flow. Int J Numer Meth Eng 1996;39(22):3839–86.

[9] Oñate E, Idelsohn SR, Zienkievicz OC, Taylor RL, Sacco C. A stabilized finite point method for analysis of fluid mechanics problems. Comput Meth Appl Mech Eng 1996;39:315–46.

[10] Taylor RL, Zienkiewicz OC, Oñate E, Idelsohn SR, Moving least square approximations for solution of differential equations. Internal report 74, CIMNE, Barcelona, Spain; 1996.

[11] Idelsohn SR, Storti MA, Oñate E. Lagrangian formulations to solve free surface incompressible inviscid fluid flows. Comput Meth Appl Mech Eng 2001;191:583–93.

[12] Idelsohn SR, Oñate E, Calvo N, DelPin F. The meshless finite element method. Int J Numer Meth Eng 2003 [in press].

[13] Edelsbrunner H, Mucke EP. Three-dimensional alpha-shape. ACM Trans Graph 1994;3:43–72.

[14] Idelsohn SR, Calvo N, Oñate E. Polyhedrization of an arbitrary 3D point set. Comput Meth Appl Mech Eng 2002 [submitted].

[15] Bathe KJ, Zhang H, Ji S. Finite element analysis of fluid flows coupled with structural interactions. Comput Struct 1999;72:1–16.

[16] Zhang H, Bathe KJ. Direct and iterative computing of fluid flows fully coupled with structures. Comput Fluid Solid Mech 2001:1440–3.

[17] Rugonyi S, Bathe KJ. CMES 2001;2(2):195–212.

[18] Codina R. Pressure Stability in fractional step finite element methods for incompressible flows. J Comput Phys 2001;170:112–40.

[19] Sukumar N, Moran B, Semenov AYU, Belikov VV. Natural neighbour Galerkin methods. Int J Numer Meth Eng 2001;50:1–27.

[20] Oñate E. Derivation of stabilized equations for advective–diffusive transport and fluid flow problems. Comput Meth Appl Mech Eng 1998;151(1–2):233–67.

[21] Oñate E. A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. Comput Meth Appl Mech Eng 2000;182 (1–2):355–70.

[22] Oñate E. Possibilities of finite calculus in computational mechanics. Int J Num Meth Eng 2002 [submitted].

[23] Belikov V, Semenov, A. Non-Sibsonian interpolation on arbitrary system of points in Euclidean space and adaptive generating isolines algorithm. Numerical grid generation in computational field simulation. In: Proceedings of the 6th international conference Greenwich University, July 1998.