Mecánica Computacional Vol. XXI, pp. 1606–1622
S. R. Idelsohn, V. E. Sonzogni and A. Cardona (Eds.)
Santa Fe–Paraná, Argentina, October 2002

THE MESHLESS FINITE ELEMENT METHOD FOR SOLVING FLUID MECHANICS PROBLEMS

Facundo Del Pin*, Sergio Idelsohn*†, Eugenio Oñate†and Nestor Calvo*

*International Center for Computational Methods in Engineering (CIMEC)
Universidad Nacional del Litoral and CONICET, Santa Fe, Argentina
e-mail: gtm@venus.ceride.gov.ar, web page: http://www.cimec.gov.ar

†International Center for Numerical Methods in Engineering (CIMNE)
Universidad Politécnica de Cataluña, Barcelona, Spain
email: onate@cimne.upc.es

Key Words: Particle method, Lagrange formulation, Incompressible Fluid Flows, Meshless

Abstract. *The presented work consists in analyzing the Meshless Finite Element Method (MFEM) in solving the Euler equations for an incompressible flow. This method is a meshless method that uses polyhedral elements. These polyhedral use special shape functions that behave as linear finite element shape functions when the polyhedral are tetrahedral or a triangle in 2-D. The time integration is evaluated by means of the fractional step method. Numerical diffusion for convective terms are unnecessary due to the Lagrangian formulation. MFEM has found to have remarkable results in high complicated geometries.*

1 INTRODUCTION

In the last twenty years computer simulation of the incompressible fluid flow equations using Eulerian formulations has been required to analyze complex geometry and physics problems. However there are still difficulties to analyze problems in which the shape of the interface changes continuously or fluid structure interaction problems where large deformations should be considered.

Particle methods have been used[1–3] where each particle is followed in a Lagrangian manner. Moving interfaces and boundaries can be analyzed by meshless method much easier than with the Finite Element Method because it is difficult to fit and move a grid continuously. Furthermore, in Lagrangian formulations the convection terms are calculated by the motion without any numerical diffusion.

A family of methods called *Meshless Methods* have been developed as well as for structural[4] as for fluid mechanics problems.[5–7] All these methods use the idea of a polynomial interpolant which fits a number of points minimizing the distance between the interpolated function and the value of the unknown field on the points.

More recently, the meshless ideas were generalize to the Finite Element Method in order to obtain the same computing time in mesh generation than in meshless connectivities.[8]

In this work new ideas and results will be presented about how to solve a particle method in fluid mechanics using the Meshless Finite Element Method. In this way a more general formulation is presented in which all the classical advantages of the FEM may be used for the unknown functions and derivatives.

2 GOVERNING EQUATIONS

The mass and momentum conservation equations may be written in a semi Lagrangian formulation as:

$$\frac{D\rho}{Dt} = -\rho\nabla\mathbf{u} \tag{1}$$

$$\rho\frac{D\mathbf{u}}{Dt} = \nabla\sigma + \rho\mathbf{f} \tag{2}$$

with

$$\sigma_{ij} = \tau_{ij} - p\delta_{ij} \tag{3}$$

and

$$\tau_{ij} = 2\mu\gamma_{ij} \tag{4}$$

where σ_{ij} is the stress tensor, p the pressure, μ the viscosity, γ_{ij} the strain deviatoric tensor and $\mathbf{f}$ a source term.

For non-viscous flow ($\mu \to 0$) the two set of equations to be used becomes

$$\frac{D\rho}{Dt} = -\rho\nabla\mathbf{u} \tag{5}$$

$$\rho\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla + \mathbf{f} \tag{6}$$

with the boundary conditions

$$p = \overline{p} \quad \text{on} \quad \Gamma_p \tag{7}$$

$$\mathbf{u}\nu = \hat{\mathbf{u}} \quad \text{on} \quad \Gamma_u \tag{8}$$

3 THE TIME SPLITTING

The time integration of Eq. 5 and Eq. 6 present some difficulties when the fluid is incompressible or near incompressible. In this case, explicit time step can not be used, and even using implicit time step, the incompressibility introduce some wiggles in the pressure solution which must be stabilized. To overcome these difficulties, a fractional step method has been proposed which consist to split each time step in 2 steps as following

$$\frac{D\mathbf{u}}{Dt} \approx \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \frac{\mathbf{u}^{n+1} - \mathbf{u}^* + \mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{\Delta\mathbf{u}' + \Delta\mathbf{u}^*}{\Delta t} \tag{9}$$

and

$$\frac{D\rho}{Dt} \approx \frac{\rho^{n+1} - \rho^n}{\Delta t} = \frac{\rho^{n+1} - \rho^* + \rho^* - \rho^n}{\Delta t} = \frac{\Delta\rho' + \Delta\rho^*}{\Delta t} \tag{10}$$

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

Now from Eq. 6 and Eq. 9

$$\frac{\Delta\mathbf{u}'}{\Delta t} + \frac{\Delta\mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1} + \mathbf{f} \tag{11}$$

which can be split into

$$\frac{\Delta\mathbf{u}^*}{\Delta t} = \mathbf{f} \tag{12}$$

$$\frac{\Delta\mathbf{u}'}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1}. \tag{13}$$

Now doing the same with Eq. 5 and Eq. 10

$$\frac{1}{\rho}\left(\frac{\Delta\rho'}{\Delta t} + \frac{\Delta\rho^*}{\Delta t}\right) = -\nabla(\mathbf{u}^{n+1} - \mathbf{u}^* + \mathbf{u}^*) \tag{14}$$

which can be split into

$$\frac{1}{\rho}\frac{\Delta\rho^*}{\Delta t} = -\nabla\Delta\mathbf{u}^* \tag{15}$$

$$\frac{1}{\rho}\frac{\Delta\rho'}{\Delta t} = -\nabla\Delta\mathbf{u}'. \tag{16}$$

But from Eq. 13 and Eq. 16

$$\frac{\Delta\rho'}{\Delta t} = \nabla(\Delta t \nabla p^{n+1}) \tag{17}$$

or

$$\frac{\rho^{\mathbf{n+1}} - \rho^{*}}{\mathbf{\Delta t^2}} = \nabla^{\mathbf{2}}\mathbf{p}^{\mathbf{n+1}} \tag{18}$$

Another possibility is to use Eq. 13 and make use of the linearity of the ∇ operator. Then

$$\nabla\frac{\Delta\mathbf{u}'}{\Delta t} = -\nabla(\frac{1}{\rho}\nabla p^{n+1}) \tag{19}$$

which becomes

$$(\nabla\mathbf{u}^{*} - \nabla\mathbf{u}^{\mathbf{n+1}})\frac{\rho}{\mathbf{\Delta t}} = \nabla^{\mathbf{2}}\mathbf{p}^{\mathbf{n+1}} \tag{20}$$

4 INCOMPRESSIBILITY CONDITIONS

The simplest way to introduce the incompressibility condition is saying that

$$\rho^{n+1} = \rho^{n} = \rho^{0} \tag{21}$$

If now a volume Ω_j is associated to each node j, the mass conservation implies

$$\Omega_j^{n+1}\rho_j^{n+1} = \Omega_j^{n}\rho_j^{n} = \Omega_j^{*}\rho_j^{*} = \Omega_j^{0}\rho_j^{0} \tag{22}$$

Then the left hand side of Eq. 18 becomes

$$\rho_j^{n+1} - \rho_j^{*} = \rho_j^{0}\left(\frac{\Omega_j^{*} - \Omega_j^{0}}{\Omega_j^{*}}\right) \tag{23}$$

The volume associated to each node Ω_j^{*} may be evaluated in different ways. For instance computing the Voronoï diagram at each node position could be a good option.

Another way to introducing the incompressibility condition is imposing in Eq. 20 the condition $\nabla\mathbf{u}^{n+1} = 0$. Then Eq. 20 becomes

$$\frac{\rho}{\Delta t}\nabla\mathbf{u}^{*} = \nabla^{2}p^{n+1} \tag{24}$$

Both Eq. 23 and Eq. 24 might be used to impose the incompressibility condition. In this paper Eq. 24 will be used.

The steps to achieve a new time step $n+1$ having u^n and p^n from the previous are:

1. Evaluate the u^* velocity from Eq. 12,

2. evaluate the divergence $\nabla\mathbf{u}^{*}$,

3. evaluate the pressure p^{n+1} solving the laplacian from Eq. 24,

4. evaluate the velocity u^{n+1} using Eq. 13,

5. move the particles to their new position r^{n+1}.

5 SPATIAL DISCRETIZATION

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

1. The first step in the algorithm is linear and explicit. The use of a Lagrangian formulation eliminates the standard convection term found in Eulerian formulations. These convection terms are responsible for non-linearity, non symmetry and non self-adjoint operators and therefore the use 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 the previous section the only implicit term is the solution of the laplacian of pressure (step 3) that is a scalar equation, symmetric and positive definite. Then it is easy to solve using an iterative scheme.

The big disadvantage of the Lagrangian formulation is the permanent updating of the node positions. For this reason standard Finite Element Methods are not useful due to the expensive process of updating conforming non-structured finite element meshes.

The key of the Lagrangian formulation is the efficiency in the computing time to evaluate the permanent mesh update or node connectivity.

Some meshless methods as the Element Free Galerkin Method (EFGM)[4] or the Natural Element Method (NEM)[9,10] have serious difficulties to solve arbitrary point distribution in a 3-D domain due to the complicated shape function used.

In this paper, the Meshless Finite Element Method (MFEM)[8] will be used. The method will be briefly explained later in this paper.

The big advantage of the MFEM compared with the FEM is the possibility to generate meshes in a computing time of order n being n the total number of nodes. Compared with the EFGM or the NEM, the major advantage is the simplicity of the shape functions, which are the same as the FEM for most of the elements in the domain.

Using the MFEM, the unknown functions will be approximated in matrix form by

$$u_i = N_i^T U = \begin{bmatrix} N^T & 0 & 0 \\ 0 & N^T & 0 \\ 0 & 0 & N^T \end{bmatrix} \begin{bmatrix} U_u \\ U_v \\ U_w \end{bmatrix} \tag{25}$$

$$p = N_p^T P = N^T P \tag{26}$$

$$\rho = N_\rho^T \underline{\rho} = N^T \underline{\rho} \tag{27}$$

where N^T are the MFEM shape function and U, P and $\underline{\rho}$ are the nodal value of the unknown function.

Using the Galerkin Weighted Residual method to solve Eq. 12, Eq. 13 and Eq. 24 with boundary conditions given by Eq. 7 and Eq. 8 and after integrating by parts some terms, the following integral equations may be written

$$\int_V N_i N_i^T dV U^* = \int_V N_i N_i^T dV U^n + \Delta t \int_V N_i f_i dV \tag{28}$$

that can be written in matrix form as

$$\mathbf{M_u U^*} = \mathbf{M_u U^n} + \mathbf{\Delta t F}. \tag{29}$$

Then the equation for the laplacian of p takes the form

$$-\frac{\rho}{\Delta t}\int_V \left(\frac{\partial N_p}{\partial x_i} N_i^T\right) dV U^* + \frac{\rho}{\Delta t}\int_{\Gamma_u} N_p \overline{u}_n^{n+1} d\Gamma = -\int_V \left(\frac{\partial N_p}{\partial x_i}\frac{\partial N_p^T}{\partial x_i}\right) dV P^{n+1} \tag{30}$$

that can be written in matrix form as

$$\frac{\rho}{\mathbf{\Delta t}}\mathbf{BU^*} - \frac{\rho}{\mathbf{\Delta t}}\mathbf{\hat{U}} = \mathbf{LP^{N+1}}. \tag{31}$$

The last integral equation to evaluate the new time step velocity is

$$\int_V N_i N_i^T dV U^{n+1} = \int_V N_i N - i^T dV U^* - \frac{\Delta t}{\rho}\int_V \frac{\partial N_p^T}{\partial x_i} dV P^{n+1} \tag{32}$$

that it is possible to write in matrix form as

$$\mathbf{M_u U^{n+1}} = \mathbf{M_u U^*} - \frac{\mathbf{\Delta t}}{\rho}\mathbf{B^T P^{n+1}}. \tag{33}$$

The matrices written above are

$$M_p = \int_V N N^T dV \tag{34}$$

$$M = \begin{bmatrix} M_p & 0 & 0 \\ 0 & M_p & 0 \\ 0 & 0 & M_p \end{bmatrix} \tag{35}$$

$$B = \left[\int_V (\tfrac{\partial N}{\partial x} N^T) dV \quad \int_V (\tfrac{\partial N}{\partial y} N^T) dV \quad \int_V (\tfrac{\partial N}{\partial z} N^T) dV\right] \tag{36}$$

$$L = \int_V \left(\frac{\partial N}{\partial x}\frac{\partial N^T}{\partial x} + \frac{\partial N}{\partial y}\frac{\partial N^T}{\partial y} + \frac{\partial N}{\partial z}\frac{\partial N^T}{\partial z}\right) dV \tag{37}$$

$$F^T = \left[\int_V N^T f_x dV \quad \int_V N^T f_y dV \quad \int_V N^T f_z dV\right] \tag{38}$$

6 MESHLESS APPROXIMATION OF THE UNKNOWN FUNCTIONS

6.1 Extended Delaunay Tessellation (EDT)

Let a set of distinct nodes be

$$\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3, \cdots, \mathbf{n}_n\} \quad \text{in} \quad \Re^3$$

The Extended Delaunay Tessellation[11] within the set $\mathbf{N}$ is the unique partition of the convex hull Ω of all the nodes into regions Ω_i such that $\Omega = \cup\Omega_i$, where each Ω_i is the polyhedral defined by all the nodes laying on the same sphere, defined by four or more nodes without any node inside.

The main difference between the traditional Delaunay Tessellation and the Extended Delaunay Tessellation is that, in the latter, all the nodes belonging to the same empty sphere define a unique polyhedron. With this definition, the domain Ω will be divided into tetrahedral and other polyhedral, which are unique for a set of node distribution, satisfying then, the first statement of the definition of a meshless method.

Fig. 1, for instance, is a 2-D polygon partition with a triangle, a quadrangle and a pentagon. Fig. 2 is a classical 8-nodes polyhedron with all the nodes on the same sphere, which may appear in a 3-D problem.

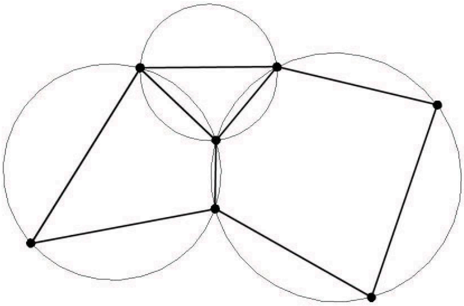

Figure 1: Two dimensional partition in polygons. The triangle, the quadrangle and the pentagon are each inscribed on a circle.

It must be noted that, for non-uniform node distribution, considering infinite precision, only 4 nodes are needed 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 polyhedral with more than four nodes, a parameter will be introduced. In such a way, the polyhedral are defined by all the nodes of the same sphere and nearby spheres where the distance between their center points is smaller than δ.

The parameter δ avoids the possibility of having zero volume or near zero volume tetrahedral. When δ is large, the number of polyhedral with more than four nodes will

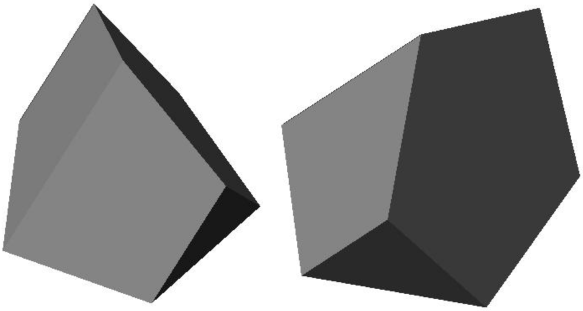

Figure 2: Eight-node polyhedron. All nodes are on the same sphere.

increase, and the number of tetrahedral with near zero volume will decrease, and vice versa.

The Extended Delaunay Tessellation allows the existence of a domain partition which:

1. is unique for a set of node distribution,
2. is formed by polyhedral with non-zero volume and
3. is obtained in a bounded time of order n.

Then it satisfies the conditions for a meshless method.

6.2 The Meshless Finite Element shape function

Once the domain partition in polyhedral is defined, shape functions must be introduced to solve a discrete problem. Limiting the study to second-order elliptic PDE's such as the Poisson's equation, C^0 continuity shape functions are necessary for a weak form solution. 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. The FEM typically uses linear or quadratic polynomial shape functions, which ensures C^0 continuity between elements. When the elements are polyhedral with different shapes, polynomial shape functions may only be used for some specific cases.

In order to define the shape functions inside each polyhedral the non-Sibsonian interpolation will be used.[12]

Let $\mathbf{R} = \{\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_n\}$ be the set of nodes belonging to a polyhedral. 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 the node $\mathbf{x}_i$ in the tessellation of the set $\mathbf{R} \cup \{\mathbf{x}\}$

and then by computing:

$$N_i(\mathbf{x}) = \frac{\frac{s_i(\mathbf{x})}{h_i(\mathbf{x})}}{\sum_{j=1}^{m} \frac{s_j(\mathbf{x})}{h_j(\mathbf{x})}} \tag{39}$$

where $s_i(\mathbf{x})$ is the surface of the Voronoï cell face corresponding to the node $\mathbf{n}_i$ and $h_i(\mathbf{x})$ is the distance between point $\mathbf{x}$ and then node $\mathbf{n}_i$ as seen in Fig. 3.

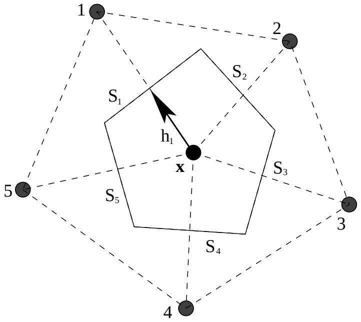

Figure 3: Five nodes and arbitrary internal point $\mathbf{x}$. Voronoï diagram and shape function parameters.

Non-Sibsonian interpolations have several properties:

1. $0 \leq N_i(\mathbf{x}) \leq 1$,
2. $\sum_i N_i(\mathbf{x}) = 1$,
3. $N_i(\mathbf{n}_j) = \delta_{ij}$ and
4. $\mathbf{x} = \sum_i N_i(\mathbf{x})\mathbf{n}_i$.

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:

- On a polyhedral surface, the shape function depends only on the nodes of this surface,[8]
- on triangular surfaces (or in all the polygon boundaries in 2-D), the shape functions are linear,
- if the polyhedral is a tetrahedral (or a triangle in 2-D) the shape functions are the linear finite element shape functions,
- the shape functions have C^0 continuity between two neighboring polyhedral (see Fig. 4),

- As a matter of fact, because all the elements nodes are on the same sphere, the evaluation of the shape functions and its derivatives becomes very simple.

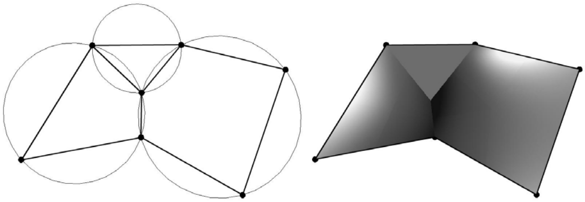

Figure 4: C^0 continuity of the shape function on a 2-D node connection.

The method defined here is termed the Meshless Finite Element Method (MFEM) because it is both a meshless method and a Finite Element Method. The algorithm steps for the MFEM are:

1. For a set of nodes compute all the empty spheres with 4 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 integration 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 are of order n avoiding all the mesh "cosmetics" often needed in mesh generation.

Fig. 5 shows the shape function and its first derivatives for a node of a 2-D pentagon. The shape function takes the value 1 at the node and 0 at the other nodes. The linear behavior on the boundaries can be appreciated.

There is an important difference between the MFEM shape functions proposed here and the Natural Element Method (NEM)[9,10] shape functions. Both method use shape functions based on Voronoï diagrams, but they are completely different. The NEM shape functions have C^∞ continuity everywhere except at the nodes where it is C^0 and are built using the Voronoï diagram of all the natural neighbor nodes at each point $\mathbf{x}$. In this way very complicated shape functions are obtained which are difficult to differentiate and which need several integration points for the numerical computation of the integrals. See Fig. 6 for a graphic representation of the NEM and MFEM shape functions.

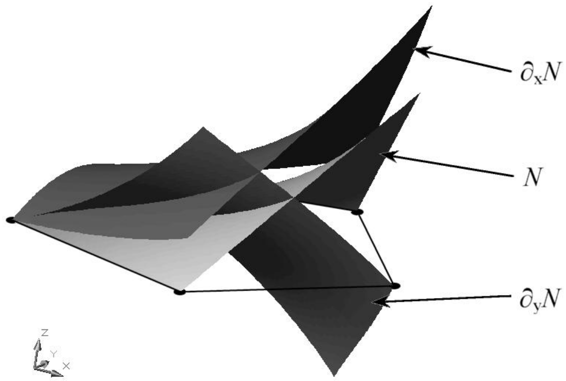

Figure 5: Shape function and its first derivatives for a typical node of a pentagon.

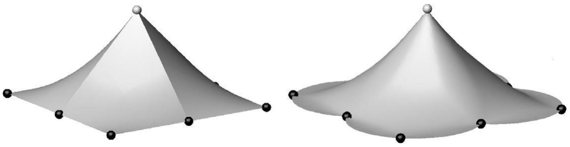

Figure 6: Shape functions in a 2-D regular node distribution. a)MFEM; b)NEM.

7 BOUNDARY SURFACES

One of the main problems in mesh-generation is the correct definition of the domain boundary. Sometimes, boundary surfaces 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 boundaries. Such is the case in Lagrangian formulation in which, at each time step, a new node distribution is obtained and the free surface must be recognized from the node positions.

The use of Voronoï diagram or Voronoï spheres can make it easier to recognize boundary surface nodes. By considering that the node follows a variable $h(\mathbf{x})$ distribution, where $h(\mathbf{x})$ is the minimum distance between two nodes, the following criterion has been used:

> All nodes which are on an empty sphere with a radius $r(\mathbf{x})$ bigger than $\alpha h(\mathbf{x})$, are considered as boundary nodes.

In this criterion, α is a parameter close to, but bigger than one. This criterion is known as the Alpha Shape concept.[13]

Once a decision has been made concerning on 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 a same sphere may define two different boundary surfaces, one concave and the other convex.

In this paper, the boundary surface is defined with all the polyhedral surfaces having all their nodes on the boundary and belonging to just one polyhedral.

The correct boundary surface may be important to define the correct normal external to the surface. Furthermore, in weak form it is also important a correct evaluation of the volume domain. 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.

8 NUMERICAL TEST

In order to show the possibilities of the method, two and three dimensional problems will be tested.

8.1 2-D problems

8.1.1 Water column collapse

This problem was solved by Koshizuka and Oka[2] experimentally and numerically. It became a classical example to test the validation of Lagrangian formulations in fluid flows. The water is first located at left supported by a removable board. The collapse starts at time $t = 0$, when the removable board is slide-up. Fig. 7 shows the point positions at different time steps.

8.1.2 Wave hitting a boat

This example (Fig. 8) shows a wave hitting the sourface of a boat.

8.2 3-D problems

A water column collapse problem as well as other collapse of water from other open recipients have been tested to show the powerful of the method developed.Fig. 9 and Fig. 10 show some of the results.

9 CONCLUSIONS

Lagrangian formulations in connection with meshless approximation is an excellent combination to solve fluid mechanical problems, particularly those in which the free surface moves continuously.

Contact problems, as well as breaking waves and collapse problems, may be solved easily without any additional constraint.

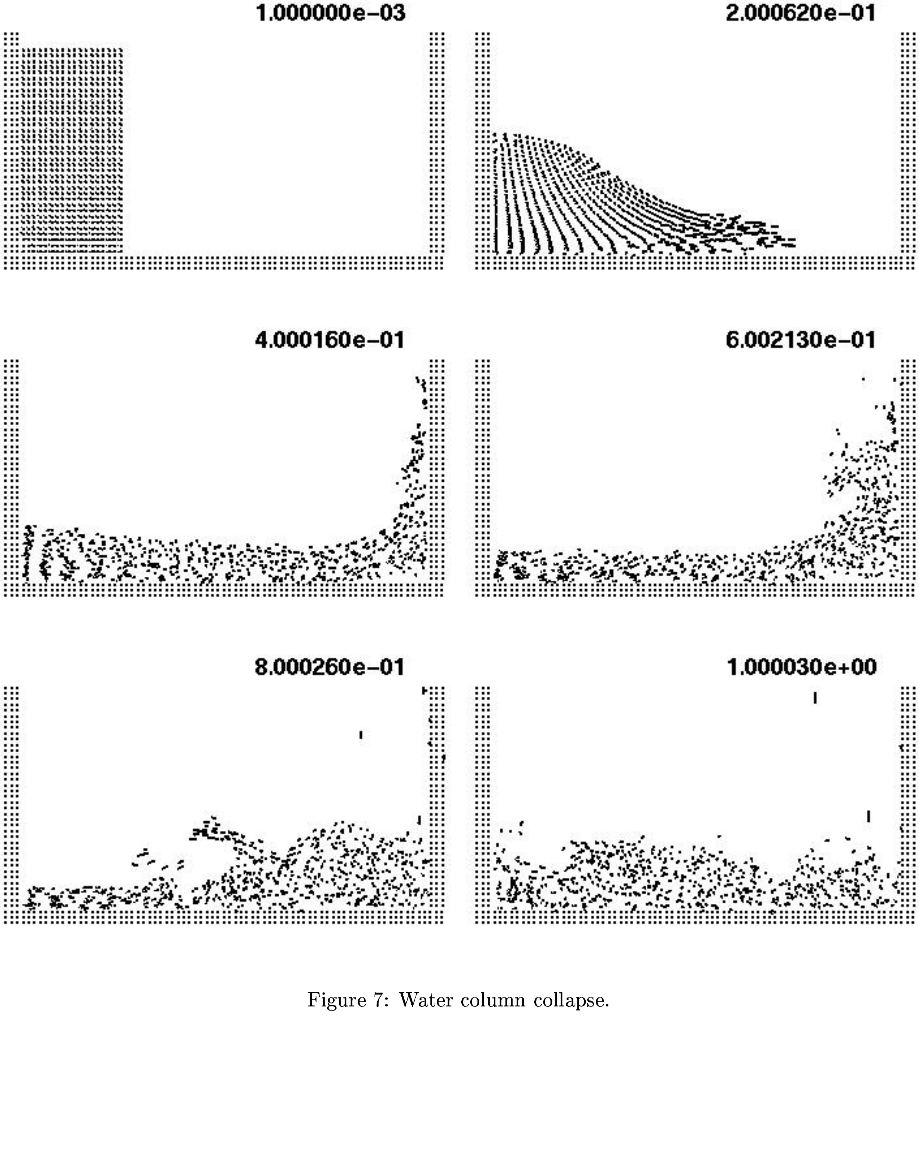

Figure 7: Water column collapse.

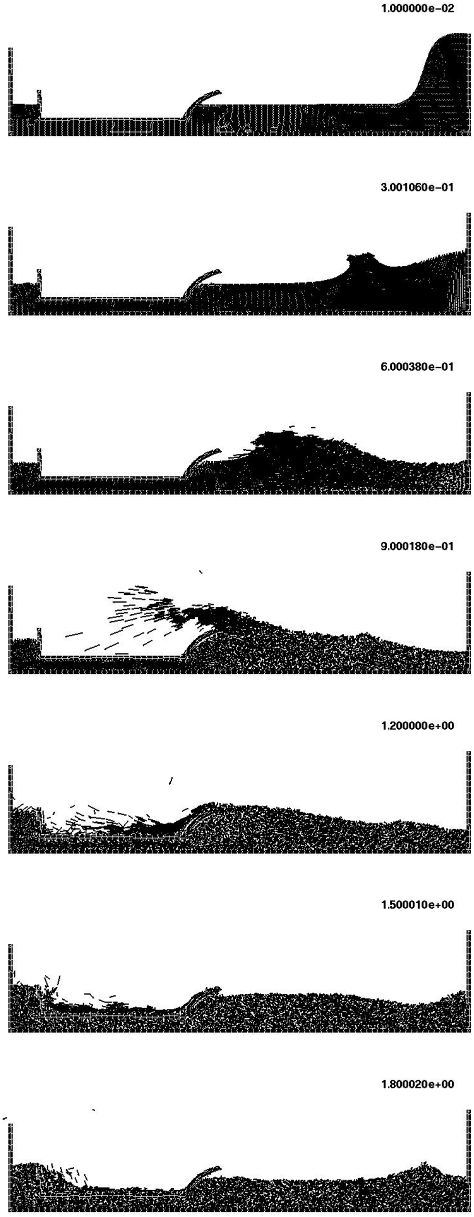

Figure 8: Wave hitting a boat.

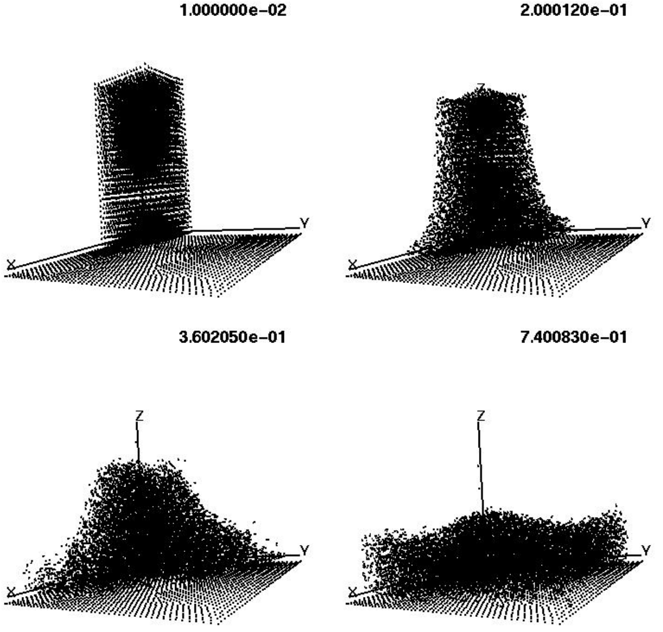

Figure 9: Water column collapse in 3-D.

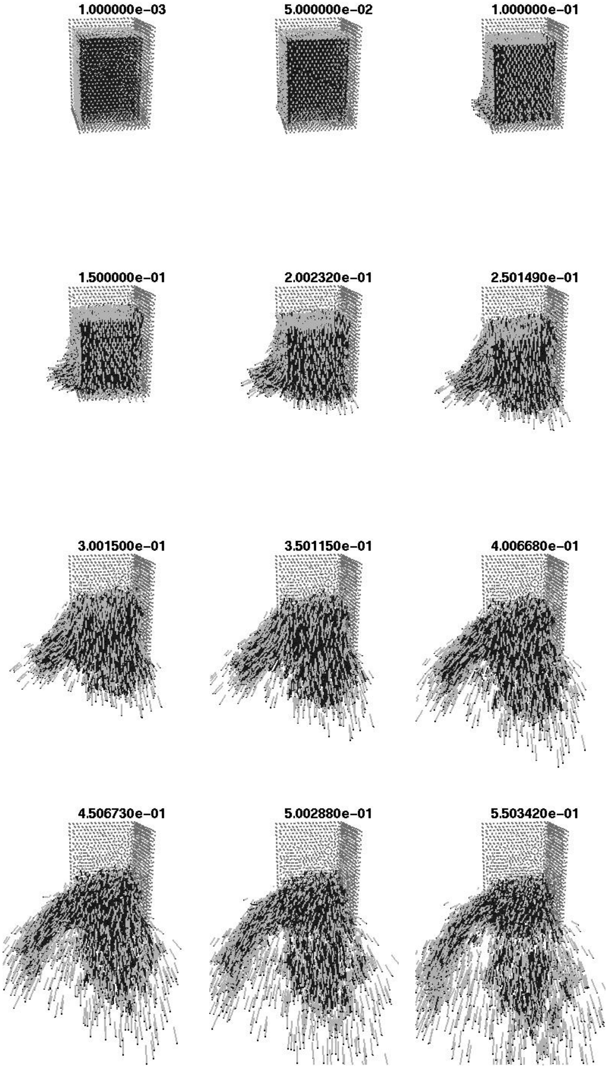

Figure 10: Water column collapse in 3-D with opened walls.

REFERENCES

[1] G.A. Dilts. Moving least squares particle hydrodynamics. *International Journal for Numerical Methods in Engineering*, **44**, 1115–1155 (1999).

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

[3] J.Bonet and S.Kulasegaram. Convection and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulation. *International Journal for Numerical Methods in Engineering*, (1999).

[4] T.Belytschko, Y. Liu, and L. Gu. Element free galerkin methods. *International Journal for Numerical Methods in Engineering*, **37**, 229–256 (1994).

[5] E.Oñate, S.R. Idelsohn, O.C.Zienkievicz, and R.L.Taylor. A finite point method in computational mechanics. applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, **39(22)**, 3839–3886 (1996).

[6] E.Oñate, S.R. Idelsohn, O.C.Zienkievicz, R.L.Taylor, and C.Sacco. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applaied Mechanics and Engineering*, **39**, 315–346 (1996).

[7] R.L.Taylor, O.C.Zienkievicz, E.Oñate, and S.R. Idelsohn. Moving least square approximations for solution of differential equations. *Internal Report, CIMNE, Barcelona, Spain*, **74** (1996).

[8] S.R. Idelsohn, E.Oñate, Nestor Calvo, and Facundo Del Pin. The meshless finite element method. *Submitted to International Journal for Numerical Methods in Engineering*, (2001).

[9] Jean Braun and Malcolm Sambridge. A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature*, **376** (August 1995).

[10] N.Sukumar, B.Moran, and T.Belytschko. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering*, **43**, 839–887 (December 1998).

[11] N.Calvo, S.R.Idelsohn, and E.Oñate. The extended delaunay tesselation. *Submitted to Computer Method in Applied Mechanics and Engineering*, (2002).

[12] V.Belikov and A.Semenov. Non-sibsonian interpolation on arbitrary system of points in euclidean soace and adaptative generating isolines algorithm. In 6^{th} *International Conference*, Greenwich University, (July 1998).

[13] H.Edelsbrunner and E.P.Mucke. Three dimensional alpha-shape. *ACM Transaction on Graphics*, **3** (1994).