New Advances in Meshless Methods: Coupling Natural Element and Moving Least Squares Techniques

Francisco Chinesta, Julien Yvonnet, Pierre Villon, Piotr Breitkopf, Pierre Joyot, Icíar Alfaro, Elías Cueto

To cite this version:

Francisco Chinesta, Julien Yvonnet, Pierre Villon, Piotr Breitkopf, Pierre Joyot, et al.. New Advances in Meshless Methods: Coupling Natural Element and Moving Least Squares Techniques. Leitao, V.M.A.; Alves, C.J.S.; Duarte, C. Armando. Advances in Meshfree Techniques, Springer, pp.97-121, 2007, Computational Methods in Applied Sciences Bookseries, 10.1007/978-1-4020-6095-3_6 . hal-01007523

HAL Id: hal-01007523
https://hal.science/hal-01007523

Submitted on 11 Oct 2017

New Advances in Meshless Methods: Coupling Natural Element and Moving Least Squares Techniques

F. Chinesta[1], J. Yvonnet[2], P. Villon[3], P. Breitkopf[3], P. Joyot[4], I. Alfaro[5] and E. Cueto[5]

[1]*LMSP UMR 8106 CNRS-ENSAM, 151 Boulevard de l'Hôpital, F-75013 Paris, France; E-mail: francisco.chinesta@paris.esnam.fr*
[2]*Université de Marne-la-Vallée, 5 Boulevard Descartes, F-77454 Marne-la-Vallée Cedex, France; E-mail: Julien.Yvonnet@univ-mlv.fr*
[3]*Université de Technlogies de Compiègne, BP 20529, F-60205 Compiègne Cedex, France; E-mail: pierre.villon@utc.fr*
[4]*LIPSI-ESTIA, Technopole Izarbel, F-64210 Bidart, France; E-mail: p.joyot@estia.fr*
[5]*Aragón Institute of Engineering Research, University of Zaragoza, Edificio Betancourt, María de Luna, 7, E-50018 Zaragoza, Spain; E-mail: ecueto@unizar.es*

Abstract. We explore in this work the connections between NN and MLS approximations, coming from the introduction of the NN approximation functions as the weights in the scope of MLS. Thus, it is easy to adjust the approximation consistency (with the possibility to enrich the approximation basis with some particular functions describing issues of the searched solution) in the framework of the MLS techniques, precribing exactly essential boundary conditions from the use of the NN approximation as MLS weight. This approach opens, as will be proved in the present paper, the way to a wide range of formulations: (i) NN collocation strategies; (ii) faster natural element discretizations; (iii) Hermite natural element formulations; (iv) hierarchical bubbles functions in the natural element method; and (v) and NN enriched approximations.

Key words: Meshless methods, Moving Least Squares, RKPM, Natural Neighbor approximations, Natural Element Method, Diffuse Finite Elements, Element Free Galerkin.

1 Introduction

The meshfree methods based on Moving Least Squares (MLS) approximation have been confronted to an active research during the last decade. These include Smooth Particle Hydrodynamics, Element Free Galerkin, Diffuse Elements, Reproducing Kernel Particle and other methods [4, 5, 12–14, 16]. However, one of the issues is the satisfaction of essential boundary conditions. This is due to the nature of the approximation itself. In fact, the MLS nodal domains of influence are the same as those of the corresponding weighting functions, who generally do not fit the boundary. The choice of neighboring nodes is also an issue. The approach based in considering the k closest nodes from any evaluation point results only in a C^0 continuity. Moreover, the geometrical complexity of the shape functions supports induces integration difficulties. Simpler integration and an arbitrary degree of continuity are obtained when nodes are associated with fixed, spherical or hexahedral domains of influence, whose optimal size constitutes the main difficulty of that approach. On the other hand, the Natural Neighbor (NN) approximation and associated family of computational methods [20] do not present these drawbacks. The boundary approximation is obtained naturally due to the fact that NN shape functions of internal nodes vanish at the boundary where only the boundary nodes contribute. The list of connected points – the natural neighbors – is also known in advance. However, the NN do not present all the advantages of the MLS. In particular, the shape function support is geometrically complex. Moreover, the NN shape functions have only C^0 continuity at the nodes and only linear consistency is guaranteed. The goal of the present paper is to connect the two approaches in order to get simultaneously the benefits of both NN and MLS approximations. The main idea lies in using NN functions as the weights in the scope of MLS. The two expected benefits are the imposition of the essential boundary conditions and a systematic framework for the search of neighbors nodes based on the Voronoi tessellation. In this way, the introduction of a visibility criterion in the NN framework, the so called constrained natural neighbour approximation (CNN) or the use of alpha-shapes – α-NEM – allow to remove the usual problems related to the application of the MLS in non-convex domains. Moreover, it allows to define nodal derivatives in the NN framework, which are required in thermomechanical simulations in order to update the internal variables when one proceeds using an updated Lagrangian formulation. This approach opens the way to a wide range of both variational and collocation formulations.

1.1 Meshless Techniques Based on the MLS Approximation: DFE and EFG Methods

Let the following approximation scheme:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{1}$$

with $\mathbf{p}^T(\mathbf{x})$ a polynomial basis, i.e. $\mathbf{p}^T(\mathbf{x}) = [1, x, y, xy]$ and $\mathbf{p}^T(\mathbf{x}) = [1, x, y, xy, x^2, y^2]$ for a bilinear and quadratic basis, respectively, in 2D, and $\mathbf{a}(\mathbf{x})$ a vector of unknown coefficients. In order to determine $\mathbf{a}(\mathbf{x})$, we define the functional J that must be minimized with respect to $\mathbf{a}(\mathbf{x})$ [16]:

$$J = \frac{1}{2}\sum_{i=1}^{n} w_i(\mathbf{x})\left[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - u_i\right]^2 \tag{2}$$

where u_i are the nodal unknowns associated with the neighbors nodes $\mathbf{x}_i$ of point $\mathbf{x}$ and $w_i(\mathbf{x})$ is a weighting function whose value decreases as the distance between $\mathbf{x}_i$ and $\mathbf{x}$ increases (see [5] for more details about properties of this function and the ones the most used). The minimization of J with respect to the unknown coefficient $a_j(\mathbf{x})$ leads to:

$$\frac{\partial J}{\partial a_j(\mathbf{x})} = \sum_{k=1}^{n} a_k \left[\sum_{i=1}^{n} w_i(\mathbf{x}) p_j(\mathbf{x}_i) p_k(\mathbf{x}_i) \right] - \sum_{i=1}^{n} w_i(\mathbf{x}) p_j(\mathbf{x}_i) u_i = 0 \quad (3)$$

which leads to the linear system:

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u} \quad (4)$$

where the matrix $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are defined by:

$$A_{jk}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) p_j(\mathbf{x}_i) p_k(\mathbf{x}_i) \quad (5)$$

$$B_{ij}(\mathbf{x}) = w_i(\mathbf{x}) p_j(\mathbf{x}_i) \quad (6)$$

Substituting $\mathbf{a}(\mathbf{x})$ in Equation (1), results in:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \quad (7)$$

By identification, the new shape functions are given by:

$$\boldsymbol{\psi}^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \quad (8)$$

The difference between the diffuse finite element and the element free Galerkin schemes comes from the evaluation of the shape function derivatives. In the first scheme only the term $\mathbf{p}^T(\mathbf{x})$ in Equation (8) is derived, whereas all terms depending on $\mathbf{x}$ are derived in the element free Galerkin approach.

1.2 Meshless Techniques Based on the Smooth Particles Approximation: The RKPM and the Enriched RKPM Methods

Let Ω be a 1D domain where the problem is defined (all the results have a direct 2D or 3D counterpart). The points within this domain will be noted by x or s.

1.2.1 Reproduction Conditions

The approximation $u^h(x)$ of $u(x)$ is built from the convolution integral

$$u^h(x) = \int_{\Omega} w(x - s, h) u(s) d\Omega \quad (9)$$

where $w(x-s,h)$ is the kernel function and h a parameter defining the size of the approximation support.

The main idea in the enriched RKPM method is to enforce the reproduction of a general function that we can write in the form of a polynomial plus another function noted by $u^e(x)$:

$$u^h(x) = a_0 + a_1 x + \cdots + a_n x^n + a_{n+1} u^e(x) \tag{10}$$

In the following paragraphs we analyze the required properties of the kernel function $w(x-s,h)$ for reproducing a function expressed by (10).

From Equation (9), the reproduction of a constant function a_0 is given by

$$\int_\Omega w(x-s,h) a_0 d\Omega = a_0 \tag{11}$$

which implies

$$\int_\Omega w(x-s,h) d\Omega = 1 \tag{12}$$

which constitutes the partition of unity.

Now, the required condition to reproduce a linear function $u^a(x) = a_0 + a_1 x$ is

$$\int_\Omega w(x-s,h)(a_0 + a_1 s) d\Omega = a_0 + a_1 x \tag{13}$$

By using the partition of unity (12), Equation (13) can be rewritten as

$$\begin{cases} \int_\Omega w(x-s,h) d\Omega = 1 \\ \int_\Omega w(x-s,h) s d\Omega = x \end{cases} \tag{14}$$

which implies the linear consistency of the approximation. Repeating this reasoning, we can write the n-order consistency as

$$\begin{cases} \int_\Omega w(x-s,h) d\Omega = 1 \\ \int_\Omega w(x-s,h) s d\Omega = x \\ \qquad\qquad \vdots \\ \int_\Omega w(x-s,h) s^n d\Omega = x^n \end{cases} \tag{15}$$

and consequently, the reproduction of the function given by (10) implies

$$\begin{aligned} &\int_\Omega w(x-s,h)(a_0 + a_1 s + \ldots + a_n s^n + a_{n+1} u^e(s)) d\Omega \\ &\quad = a_0 + a_1 x + \cdots + a_n x^n + a_{n+1} u^e(x) \end{aligned} \tag{16}$$

from which it results

$$\begin{cases} \int_\Omega w(x-s,h)d\Omega = 1 \\ \int_\Omega w(x-s,h)sd\Omega = x \\ \qquad\qquad \vdots \\ \int_\Omega w(x-s,h)s^n d\Omega = x^n \\ \int_\Omega w(x-s,h)u^e(s)d\Omega = u^e(x) \end{cases} \tag{17}$$

In the original procedure proposed by Liu et al. [12] only n-order consistency was imposed, but it can not be directly used to enforce the reproduction condition associated with $u^e(x)$.

1.2.2 The Moment Matrix

We will note by $u^r(x)$ the approximation function verifying the conditions (17). Usually a cubic spline is considered as kernel function, and consequently the conditions given by Equation (17) are not satisfied. Liu et al. [12] propose the introduction of a correction function $C(x, x-s)$ for satisfying the reproduction conditions. In our case we consider the more general form $C(x, s, x-s)$ whose pertinence will be discussed later. Thus $u^r(x)$ will be expressed by

$$u^r(x) = \int_\Omega C(x,s,x-s)w(x-s,h)u(s)d\Omega \tag{18}$$

where $C(x, s, x-s)$ is assumed to have the following form

$$C(x,s,x-s) = \mathbf{H}^T(x,s,x-s)\mathbf{b}(x) \tag{19}$$

where $\mathbf{H}^T(x, s, x-s)$ represents the vector containing the functions considered in the approximation basis, and $\mathbf{b}(x)$ is a vector containing unknown functions that will be determined for satisfying the reproduction conditions. Thus, Equation (17) can be rewritten as

$$\begin{cases} \int_\Omega \mathbf{H}^T(x,s,x-s)\mathbf{b}(x)w(x-s,h)d\Omega = 1 \\ \int_\Omega \mathbf{H}^T(x,s,x-s)\mathbf{b}(x)w(x-s,h)sd\Omega = x \\ \qquad\qquad \vdots \\ \int_\Omega \mathbf{H}^T(x,s,x-s)\mathbf{b}(x)w(x-s,h)s^n d\Omega = x^n \\ \int_\Omega \mathbf{H}^T(x,s,x-s)\mathbf{b}(x)w(x-s,h)u^e(s)d\Omega = u^e(x) \end{cases} \tag{20}$$

In fact, the reproduction conditions must be enforced in a discrete form. For this purpose we consider N points (also refereed as nodes) which allow to compute the discrete form of Equation (20), i.e.

$$\begin{cases} \sum_{i=1}^{N} \mathbf{H}^T(x, x_i, x-x_i)\mathbf{b}(x)w(x-x_i, h)\Delta x_i = 1 \\ \sum_{i=1}^{N} \mathbf{H}^T(x, x_i, x-x_i)\mathbf{b}(x)w(x-x_i, h)x_i\Delta x_i = x \\ \qquad\qquad\vdots \\ \sum_{i=1}^{N} \mathbf{H}^T(x, x_i, x-x_i)\mathbf{b}(x)w(x-x_i, h)x_i^n\Delta x_i = x^n \\ \sum_{i=1}^{N} \mathbf{H}^T(x, x_i, x-x_i)\mathbf{b}(x)w(x-x_i, h)u^e(x_i)\Delta x_i = u^e(x) \end{cases} \tag{21}$$

that in a matrix form results

$$\left[\sum_{i=1}^{N} \mathbf{R}(x_i)\mathbf{H}^T(x, x_i, x-x_i)w(x-x_i, h)\Delta x_i\right]\mathbf{b}(x) = \mathbf{R}(x) \tag{22}$$

where $\mathbf{R}(x)$ is the reproduction vector

$$\mathbf{R}^T(x) = \left[1, x, \ldots, x^n, u^e(x)\right] \tag{23}$$

Equation (22) allows the computation of vector $\mathbf{b}(x)$,

$$\mathbf{b}(x) = \mathbf{M}(x)^{-1}\mathbf{R}(x) \tag{24}$$

where the moment matrix $\mathbf{M}(x)$ is defined by

$$\mathbf{M}(x) = \sum_{i=1}^{N} \mathbf{R}(x_i)\mathbf{H}^T(x, x_i, x-x_i)w(x-x_i, h)\Delta x_i \tag{25}$$

This moment matrix differs from the usual moment matrix proposed in [12], and in fact it becomes non symmetric.

1.2.3 Discrete Form of the Approximation Function

The discrete form $u^r(x)$ of $u^h(x)$ derives from Equations (18), (19) and (24)

$$\begin{aligned} u^r(x) &\cong \sum_{i=1}^{N} \mathbf{H}^T(x, x_i, x-x_i)\mathbf{M}(x)^{-1}\mathbf{R}(x)w(x-x_i, h)u(x_i)\Delta x_i \\ &= \sum_{i=1}^{N} \psi_i(x)u_i \end{aligned} \tag{26}$$

where ψ_i is the enriched RKP approximation shape function

$$\psi_i(x) = \mathbf{H}^T(x, x_i, x-x_i)\mathbf{M}(x)^{-1}\mathbf{R}(x)w(x-x_i, h)\Delta x_i \tag{27}$$

As in the classical RKPM we take $\Delta x_i = 1$. Different quadrature rules exist and they have been tested without a significant incidence on the reproducing condition accuracy.

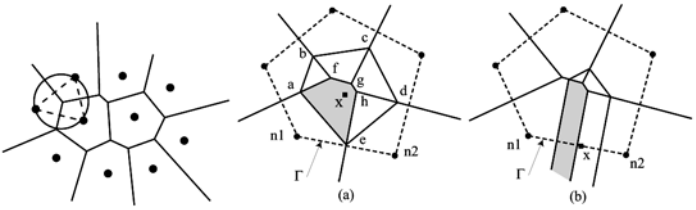

Fig. 1. (Left) Voronoi diagram, Delaunay triangle and Delaunay circle. (Right) Construction of the Sibson shape functions.

1.3 Meshless Techniques Based on the Natural Neighbor Approximation: The NEM Method

We briefly touch upon the foundation of Sibson's natural neighbor (NN) coordinates (shape functions) that are used in the natural element method. For a more in-depth discussion on the Sibson interpolant and its application for solving second-order partial differential equations, the interested reader can refer to Sambridge and Braun [19] and Sukumar et al. [20]. The NEM interpolant is constructed on the basis of the Voronoi diagram (see Figure 1). The Delaunay tessellation is the topological dual of the Voronoi diagram.

Consider a set of nodes $S = \{n_1, n_2, \ldots, n_N\}$ in $\Re^{dim}$. The Voronoi diagram is the subdivision of $\Re^{dim}$ into regions T_i (Voronoi cells) defined by:

$$T_i = \{\mathbf{x} \in \Re^{dim} : d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j), \forall j \neq i\}, \quad \forall\, i \tag{28}$$

The Sibson coordinates of $\mathbf{x}$ with respect to a natural neighbor n_i (see Figure 1 (right)) is defined as the ratio of the overlap area (volume in 3D) of their Voronoi cells to the total area (volume in 3D) of the Voronoi cell related to point $\mathbf{x}$. If we consider the 2D example of Figure 1(a), we have:

$$\phi_1(\mathbf{x}) = \frac{Area(afghe)}{Area(abcde)} \tag{29}$$

Remark. From now on, we denote by $\phi_i(\mathbf{x})$ the shape functions related to the NEM, whereas $\psi_i(\mathbf{x})$ is used to denote the ones associated with the MLS or RKPM techniques.

If the point $\mathbf{x}$ coincides with the node n_i, i.e. ($\mathbf{x} = \mathbf{x}_i$), $\phi_i(\mathbf{x}_i) = 1$, and all other shape functions are zero, i.e. $\phi_j(\mathbf{x}_i) = \delta_{ij}$ (δ_{ij} being the Kronecker delta). The properties of positivity, interpolation, and partition of unity are then verified [20]:

$$\begin{cases} 0 \leq \phi_i(\mathbf{x}) \leq 1 \\ \phi_i(\mathbf{x}_j) = \delta_{ij} \\ \sum_{i=1}^{n} \phi_i(\mathbf{x}) = 1 \end{cases} \tag{30}$$

where n is the number of neighbor nodes related to point $\mathbf{x}$.

The natural neighbor shape functions also satisfy the local coordinate property [22], namely:

$$\mathbf{x} = \sum_{i=1}^{n} \phi_i(\mathbf{x})\mathbf{x}_i \tag{31}$$

which combined with Equation (30), implies that the natural neighbor interpolant spans the space of linear polynomials (linear completeness).

Sibson natural neighbor shape functions are C^1 at any point except at the nodes, where they are only C^0. The C^1 continuity away from the nodes can be improved by using special classes of natural neighbor shape functions [11].

The support (domain of influence) of a shape function ϕ_i is the union of the Delaunay spheres (circumscribing the Delaunay tetrahedrons) containing the node n_i. This support is thus not radial and automatically adapts to the relative position of n_i and its neighbors, whether is the density or the regularity of the local nodal distribution.

Another important property of this interpolant is its strict linearity over the boundary of convex domains. The proof can be found in Sukumar et al. [20]. An illustration is depicted in Figure 1: as the areas associated to points on the boundary become infinite, the contribution of internal points vanish in the limit when the point approaches the convex boundary, and the shape functions associated with nodes n_1 and n_2 become linear on the segment $(n_1 - n_2)$. This is not true in the case of non convex boundaries, and an appropriate treatment must be introduced to maintain this property over non-convex boundaries [9, 24]. In tandem with the delta Kronecker property, essential boundary conditions can thus be enforced directly, as in the finite element method. This property also guarantees strict continuity of the approximation across material interfaces [23], which is an issue in most meshfree methods.

Consider an interpolation scheme for a vector-valued function $\mathbf{u}(\mathbf{x}) : \Omega \subset \Re^2 \rightarrow \Re$, in the form:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i(\mathbf{x})\,\mathbf{u}_i \tag{32}$$

where $\mathbf{u}_i$ are the nodal values of the field at the n natural neighbor nodes of point $\mathbf{x}$, and $\phi_i(\mathbf{x})$ are the shape functions associated with each neighbor node. It is noted that Equation (32) defines a local interpolation scheme. Thus, the trial and test functions used in the discretization of the variational formulation describing the problems treated in this paper take the form of Equation (32).

2 Coupling NN and MLS Approximations

In this section, that constitutes the main contribution of this paper, different hybrid schemes combining the natural neighbor and the moving least squares interpolations will be proposed and analyzed.

2.1 Defining Natural-Neighbor Collocation Schemes

The proposed approximation scheme consists in using the NN shape functions as weights in the MLS approximation. In this way we want to increase the consistency of the NN approximation in an adjustable manner and get simpler formulas for computing the derivatives. We will show firstly, that this approach does not change the shape functions when choosing linear polynomial basis $\mathbf{P}^T(\mathbf{x}) = \{1, x, y\}$. When we minimize the usual moving least square criterion

$$J = \frac{1}{2} \sum_{i=1}^{n} w_i(\mathbf{x}) \left[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - u_i \right]^2 \tag{33}$$

with $w_i(\mathbf{x})$ the NN shape function, i.e. $w_i(\mathbf{x}) = \phi_i(\mathbf{x})$, it results (see [7] for details) $\Psi_i(\mathbf{x}) = \phi_i(\mathbf{x})$ and that diffuse derivatives are discontinuous at the nodes. The same results are obtained by considering quadratic reproduction conditions, i.e. $P^T(\mathbf{x}) = \{1, x, y, x^2, xy, y^2\}$.

Thus, the use of collocation techniques becomes delicate. To circumvent this difficulty, we consider another cloud of auxiliary points $\mathbf{x}_j^*$, as in the double grid technique [6]. Now, the approximation field derivatives can be defined at those points, and then the nodal approximation derivatives defined at nodes using the moving least squares technique.

If we denote by u^h the natural neighbor approximation, its derivative can by calculated at points $\mathbf{x}_j^*$. Now, the diffuse derivatives can be computed at any point $\mathbf{x}$ using the standard MLS technique, from the functionals:

$$J = \frac{1}{2} \sum_{j=1}^{m} w_j(\mathbf{x}) \left[\mathbf{p}^T(\mathbf{x}_j^*)\mathbf{a}(\mathbf{x}) - \frac{\partial u^h}{\partial x}(\mathbf{x}_j^*) \right]^2 \tag{34}$$

and

$$J = \frac{1}{2} \sum_{j=1}^{m} w_j(\mathbf{x}) \left[\mathbf{p}^T(\mathbf{x}_j^*)\mathbf{a}(\mathbf{x}) - \frac{\partial u^h}{\partial y}(\mathbf{x}_j^*) \right]^2 \tag{35}$$

where m is the number of auxiliary points. Thus, after minimization, the coefficients $\mathbf{a}(\mathbf{x})$ of both approximations are obtained, and then the shape function diffuse derivatives, allowing to define the diffuse derivatives at any point $\mathbf{x}$:

$$\frac{\delta^2 u}{\delta x^2}(\mathbf{x}) = \sum_{j=1}^{m} \Psi_{x,j}(\mathbf{x}) \, \frac{\partial u^h}{\partial x}(\mathbf{x}_j^*) \tag{36}$$

and

$$\frac{\delta^2 u}{\delta y^2}(\mathbf{x}) = \sum_{j=1}^{m} \Psi_{y,j}(\mathbf{x}) \, \frac{\partial u^h}{\partial y}(\mathbf{x}_j^*) \tag{37}$$

where $\Psi_{x,j}(\mathbf{x})$ and $\Psi_{y,j}(\mathbf{x})$ denote the diffuse shape function derivatives related to point $\mathbf{x}_j^*$ with respect to the x and y coordinates, evaluated at point $\mathbf{x}$.

Now, considering that:

$$\frac{\partial u^h}{\partial x}(\mathbf{x}_j^*) = \sum_{i=1}^{n} \phi_{x,i}(\mathbf{x}_j^*) u_i \tag{38}$$

and

$$\frac{\partial u^h}{\partial y}(\mathbf{x}_j^*) = \sum_{i=1}^{n} \phi_{y,i}(\mathbf{x}_j^*) u_i \tag{39}$$

where $\phi_{x,i}(\mathbf{x}_j^*)$ and $\phi_{y,i}(\mathbf{x}_j^*)$ denote the Natural Neighbor shape function derivatives related to node $\mathbf{x}_i$ with respect to the x and y coordinates evaluated at point $\mathbf{x}_j^*$, Equations (36) and (37) could be used to define collocation schemes.

Remark. When the point $\mathbf{x}$ approaches to $\mathbf{x}_i$, Equations (36) and (37) give the nodal diffuse derivatives, that can be used in the postprocessing or in the context of a collocation technique.

Linear convergence of the second derivative may be demonstrated for an appropriate election of the auxiliary points. For this purpose we locate the auxiliary points (for quadratic approximation consistency) verifying $\forall \mathbf{x}_j^*$ the following conditions:

$$\sum_{i=1}^{n} \left\{ \phi_{x,i}(\mathbf{x}_j^*) u_i - \phi_i(\mathbf{x}^*) u_{x,i} \right\} = 0 \tag{40}$$

and

$$\sum_{i=1}^{n} \left\{ \phi_{y,i}(\mathbf{x}_j^*) u_i - \phi_i(\mathbf{x}^*) u_{y,i} \right\} = 0 \tag{41}$$

where n is the number of neighbor nodes of point $\mathbf{x}_j^*$, and

$$\begin{cases} u_i = u(\mathbf{x}_i) = a + bx_i + cy_i + dx_i^2 + ex_i y_i + fy_i^2 \\ u_{x,i} = \frac{\partial u}{\partial x}(\mathbf{x}_i) = b + 2dx_i + ey_i \\ u_{y,i} = \frac{\partial u}{\partial y}(\mathbf{x}_i) = c + 2fy_i + ex_i \end{cases} \tag{42}$$

This strategy was successfully applied for solving second order partial differential equations using a collocation discretization in [7].

2.2 Faster Natural-Neighbor Interpolation Formulas

The natural neighbor meshfree method provides equivalent quality compared to quadrilateral/hexaedral finite elements, but only uses the Delauny triangulation (which is automatic and unique for a given cloud of nodes), to construct the shape functions, which avoids the burden of mesh generation with these elements. Furthermore, as the shape functions satisfy the Kroenecker delta property, the imposition of essential boundary conditions is direct, unlike in the vast majority of meshless methods. Nevertheless, the computation of 3D natural neighbor shape functions is complex and costly, involving geometric constructions in the Voronoi diagram.

The computation of the natural neighbor shape functions is not direct and requires some geometric operations (intersection, volume and area computations) at each integration point. A classical algorithm for the computation of the shape functions at a point $\mathbf{x}$ involves the following steps: (a) Find the natural neighbor of the point $\mathbf{x}$; (b) Construct the new Voronoi cell associated with point $\mathbf{x}$; (c) Compute the volumes or areas associated with Voronoi cells entities used in the shape function computations; and (d) Compute the shape functions. Step (a) can be performed in constant time by performing local search in the Voronoi diagram. In our experience, steps (b) and (c) are the most expensive from a CPU point of view. In the next section, we propose new natural neighbor shape functions which avoid the geometric operations involved in the steps (b) and (c).

For this purpose, a particular weight function $w_i(\mathbf{x})$ based on the Delaunay spheres is used, which posses the main features of natural neighbor shape functions support [25]. The introduction of this particular weight in the EFG methology leads to shape functions which posses the same properties of the natural neighbor shape functions (i.e. interpolation and connectivity based on the natural neighbors, and linear consistency), but without any geometric construction based on the Voronoi diagram, which simplifies the extension of the method to the 3D case and reduces the computational costs.

2.2.1 Pseudo Natural Neighbor Weight Functions

When we consider the MLS technique, summarized in the first section of this paper, the approximation of a field $u(\mathbf{x})$, $u^h(\mathbf{x})$, can be written as:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}\mathbf{B}\mathbf{u} = \boldsymbol{\psi}^T(\mathbf{x})\,\mathbf{u} \tag{43}$$

where $\boldsymbol{\psi}(\mathbf{x})$ is the vector containing the shape functions associated with neighbors of point x. In the following, we are interested in defining an appropriate weight function $w_i(\mathbf{x})$ such as the resulting shape functions satisfy: (a) the Kroenecker delta property

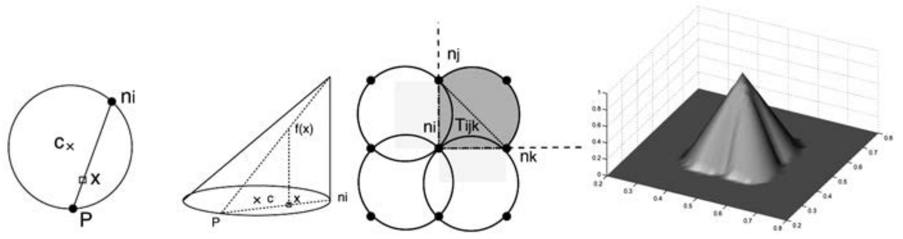

Fig. 2. (Left) Eccentric conical function. (Center) Support of a cone function portion. (Right) Weight function.

($\psi_i(\mathbf{x}_j) = \delta_{ij}$); (b) the linear consistency (which is automatically satisfied in the MLS framework); (c) the linearity of the shape functions on the domain boundary; and (d) $\psi_i(\mathbf{x}_j)$ vanishes on the Delaunay spheres containing the node n_i.

The following weight function is proposed to satisfy the former conditions. The definition is given here in 2D, but is straightforward in 3D. Let a cone function which basis matches one of the Delaunay circle containing the point x, and where the projection of the tip matches the node n_i (see Figure 2).

The value of the conic function computed at point x is given by:

$$f(\mathbf{x}) = \frac{\|\mathbf{n_i P}\| - \|\mathbf{n_i x}\|}{\|\mathbf{n_i P}\|} \tag{44}$$

with:

$$\mathbf{n_i P} = -2\left(\frac{\mathbf{cn_i} \cdot \mathbf{n_i x}}{\mathbf{n_i x} \cdot \mathbf{n_i x}}\right)\mathbf{n_i x} \tag{45}$$

In order to avoid the overlapping of cone functions whereas conserving the continuity of the weight function, a cone portion is associated with each of the Delaunay triangles connected to node n_i. The cone function is thus non-zero if a point x belong to the intersection between the Delaunay circumcircle and the portion of the plane such as any point in the basis formed by the origin node n_i and the vectors $n_i - n_j$ and $n_i - n_k$ has positive coordinates in this basis. n_j et n_k are the other two vertices of the triangle (see Figure 2). Due to the particular shape of its support, this weight function guarantees interpolation conditions ($w_i(\mathbf{x}_j) = \delta_{ij}$), as Delaunay circles passes through the nodes. Furthermore, the properties of positiveness and monotonically decreasing are verifed. As the cone functions are linear between two nodes, the continuity of the weight function is guaranteed.

In order to guarantee strict linearity of the shape functions over the boundaries of the domain, the shape functions associated with interior node must vanish on the external boundaries. For this purpose, we multiply the weight function in Equation (44) by a function $\Xi(\mathbf{x})$ which vanish over the external boundaries. A simple solution is to define $\Xi(\mathbf{x})$ by:

$$\Xi(\mathbf{x}) = \mathbf{N}^T(\mathbf{x})\boldsymbol{\delta} \tag{46}$$

where $\mathbf{N}(\mathbf{x})$ is the vector containing the linear finite element shape functions associated with the Delaunay triangles, and $\boldsymbol{\delta}$ the nodal values of a field taking a unit value inside the domain, i.e. $\delta_i = 1$ if $\mathbf{x}_i \in \Omega$, vanishing on the domain boundary $\Gamma \equiv \partial\Omega$, i.e. $\delta_i = 0$ if $\mathbf{x}_i \in \Gamma$.

2.2.2 Numerical Example

The following Poisson's problem is considered in a 3D unit cube:

$$\begin{cases} \Delta u = 0 \; in \; \Omega =]0,1[^3 \\ u(\mathbf{x}) = u_g(\mathbf{x}) = 2x^2 - y^2 - z^2 \;\text{ on }\; \Gamma = \partial\Omega \end{cases} \tag{47}$$

whose exact solution results:

$$u^{ex}(\mathbf{x}) = 2x^2 - y^2 - z^2 \;\text{ in }\; \Omega \tag{48}$$

The weak form associated with the problem defined in Equation (47) is expressed by:

Find $u \in H^1(\Omega)$ ($u = u_g$ on Γ) such that:

$$\int_\Omega \nabla u^* \cdot \nabla u \, d\Omega = 0, \; \forall u^* \in H_0^1(\Omega) \tag{49}$$

where $H^1(\Omega)$ and $H_0^1(\Omega)$ are usual Sobolev functional spaces.

The problem has been solved by using several refined meshes: $3\times3\times3$, $5\times5\times5$, $7\times7\times7$ and $10\times10\times10$ nodes. The energy norm has been computed to determine the convergence of the solution. Results are depicted in Figure 3. A comparison between the computational times associated with standard Sibson and pseudo-NEM shape functions is depicted in Figure 3 (right).

2.2.3 CPU Time Comparison between Different NN Approximations

As mentioned before, CPU time is one of the major drawbacks of the NEM, when compared to that of the FEM. In order to compare how important this cost could be, we analyze here a problem with know analytical solution, solved through the four different techniques, namely, NEM-Sibson, NEM-Laplace, Pseudo-NEM and, of course, FEM.

The problem here considered is the compression of a cubic block of a linear elastic material (or, equivalently, a stress *patch test*). A displacement of 0.01 on z direction is prescribed on the top face and the nodes of the bottom face are constrained in the z direction, see Figure 4.

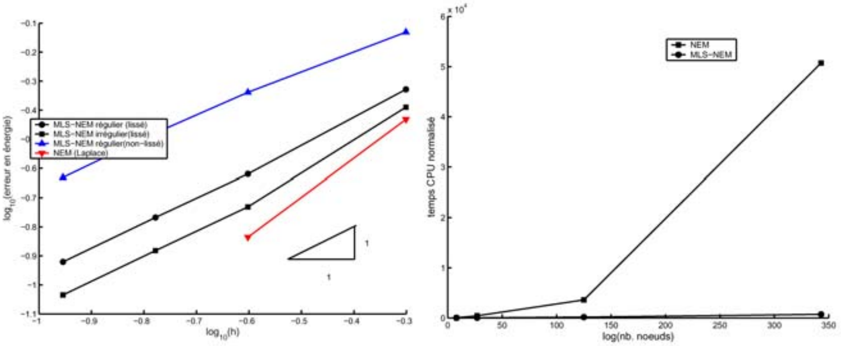

Fig. 3. (Left) Convergence analysis of the Poisson's problem. (Right) Comparison of CPU times of Sibson-NEM and pseudo-NEM discretizations.

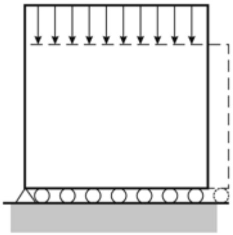

Fig. 4. Geometry of the compression test.

Different clouds of nodes have been employed, both in regular and irregular distributions, in an attempt to check wether the relative location of the nodes has an influence on the neighbour search time.

In the FEM context, Delaunay triangulation has been used, and three integration points are used in each tetrahedron. All meshless simulations use the same integration points.

Figure 5 shows the time needed to solve different problems. It is obvious that solving the problem using Sibson's shape functions (denoted by NEM-S) employs the highest amount of time, increasing very fastly with the number of nodes. Furthermore, the time needed for irregular distributions is higher than regular ones, since regular ones have less Delaunay tetrahedra and thus less integration points.

It is also interesting to evaluate the ratio of total time employed in calculating the shape functions and their derivatives, compared with the CPU time of the whole simulation. This is shown in Figure 6, where the pseudo NEM (P-NEM) approach needs less than 10% of total time and decreases comparatively when the size of the problem increases. Laplace shape functions (NEM-L) also decreases, but takes about 30% of the total time. Again, Sibson approach is very time consuming.

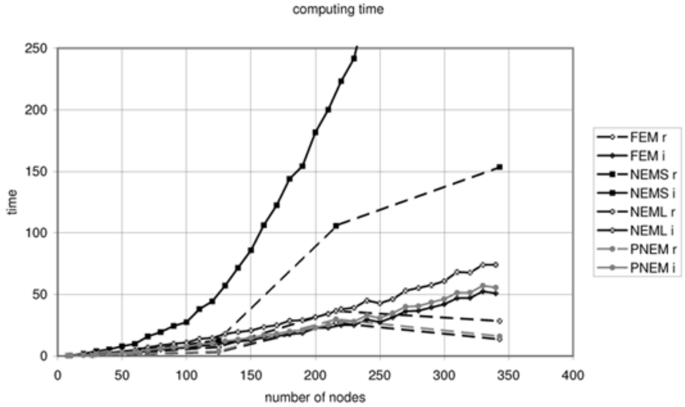

Fig. 5. Total computing time comparison.

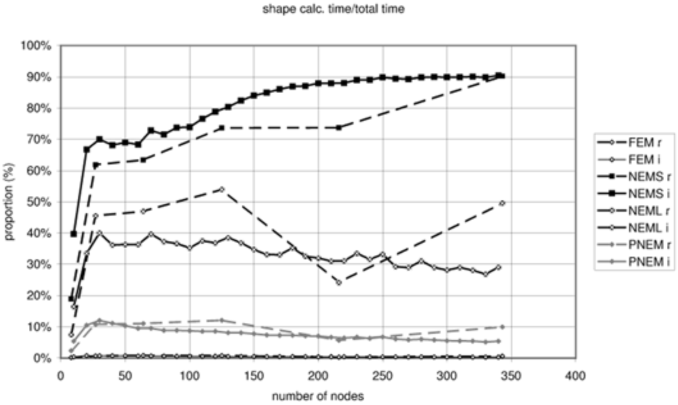

Fig. 6. Time employed in shape function calculation vs. total simulation time.

2.3 Hermite Natural Element Formulation

In this section, quadratic approximation consistency is achieved through a diffuse Hermite interpolation [18], by using natural neighbor weights in the moving least square approximation. Compared to standard moving least square method, the minimization is performed both with respect to the primary variable, and the diffuse spatial derivatives. For this purpose, we consider an interpolation scheme in the form:

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \psi_i(\mathbf{x}) u_i + \sum_{i=1}^{n} \psi_i^x(\mathbf{x}) \frac{\partial u_i}{\partial x} + \sum_{i=1}^{N} \psi_i^y(\mathbf{x}) \frac{\partial u_i}{\partial y} \tag{50}$$

where $\psi_i(\mathbf{x})$ are the shape function associated with the unknown variable u_i, $\psi_i^x(\mathbf{x})$ and $\psi_i^y(\mathbf{x})$ are the shape function associated with the space derivative of u_i with respect to x and y, respectively. For the sake of simplicity, we consider the sum in Equation (50) extended to all the nodes N instead to the n natural neighbors, but both expressions are equivalent because the shape functions related to non-neighbor nodes vanish. In the above framework, u_i, $\partial u_i/\partial x$ and $\partial u_i/\partial y$ are unknown (degrees of freedom). In order to construct the shape functions, we consider the following approximation scheme:

$$u^h(\mathbf{x}) = \mathbf{p}(\mathbf{x})^T \mathbf{a}(\mathbf{x}) \tag{51}$$

where $\mathbf{p}(\mathbf{x})$ is a polynomial basis, i.e. $\mathbf{p}(\mathbf{x}) = \{1, x, y, xy, x^2, y^2\}$ and $\mathbf{a}(\mathbf{x})$ is a vector of unknown coefficients. In order to determine $\mathbf{a}(\mathbf{x})$, we consider the following functional:

$$J = \frac{1}{2}\sum_{i=1}^{N} w_i(\mathbf{x}) \left\{ \left[\mathbf{p}^T(\mathbf{x})\mathbf{a} - u_i\right]^2 + \right.$$
$$\left. + \alpha \left[\frac{\partial \mathbf{p}^T}{\partial x}(\mathbf{x})\mathbf{a} - \frac{\partial u_i}{\partial x}\right]^2 + \alpha \left[\frac{\partial \mathbf{p}^T}{\partial y}(\mathbf{x})\mathbf{a} - \frac{\partial u_i}{\partial y}\right]^2 \right\} \tag{52}$$

where $w_i(\mathbf{x})$ are the natural neighbor shape functions computed at point x, i.e. $w_i(\mathbf{x}) = \phi_i(\mathbf{x})$, $\partial \mathbf{p}^T/\partial x(\mathbf{x})$ and $\partial \mathbf{p}^T/\partial y(\mathbf{x})$ represent the derivative of the basis $\mathbf{p}(\mathbf{x})$ with respect to x and y, respectively. α is a dimensional parameter which is fixed to 1 in our simulations. Minimizing J with respect to $\mathbf{a}(\mathbf{x})$, $(\partial J/\partial \mathbf{a}(\mathbf{x}) = 0)$, leads to the following system of equations:

$$\mathbf{A}\mathbf{a}(\mathbf{x}) = \mathbf{B}\mathbf{q} \tag{53}$$

with

$$\mathbf{q} = \left\{ u_1, \frac{\partial u_1}{\partial x}, \frac{\partial u_1}{\partial y}, u_2, \frac{\partial u_2}{\partial x}, \frac{\partial u_2}{\partial y}, \ldots, u_N, \frac{\partial u_N}{\partial x}, \frac{\partial u_N}{\partial y} \right\}.$$

Derivatives of the shape functions are obtained through standard procedure [4], involving the derivative of the weight functions $w_i(\mathbf{x})$. The derivatives involved in $\mathbf{q}$ are in fact pseudo-derivatives (diffuse derivatives) and they cannot be used for discretizing variational formulations, however they can be used in a collocation framework. Closed form of Sibson shape functions derivatives can be found in [17]. The obtained shape functions are depicted in Figure 7.

According to Equation (52), the new degrees of freedom associated with the derivatives can be interpreted like pseudo-derivatives which do not coincide with the real derivatives. Thus, imposition of essential boundary conditions becomes delicate. Nevertheless, in order to investigate the accuracy of the technique without being polluted by this issue, we consider in next section a Poisson's whose solution and its derivatives on the boundary vanish.

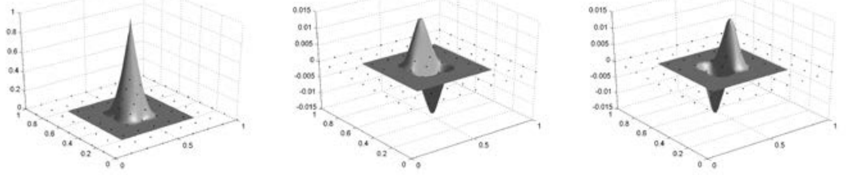

Fig. 7. Hermite natural neighbor shape functions.

2.3.1 Numerical Example

The boundary value problem is defined by:

$$\begin{cases} -\Delta u = f \text{ in } \Omega =]0,1[\times]0,1[\\ u = u_g \qquad \text{on } \Gamma \equiv \partial\Omega \end{cases} \tag{54}$$

we consider from now on:

$$\begin{cases} u_g = 0, \\ f = 4\pi^2 \{2\cos(2\pi x)\cos(2\pi y) - \cos(2\pi x) - \cos(2\pi y)\} \end{cases} \tag{55}$$

whose exact solution results in:

$$u^{ex}(\mathbf{x}) = \{1 - \cos(2\pi x)\}\{1 - \cos(2\pi y)\} \tag{56}$$

The weak form associated with Equation (54) is given by:
Find $u \in H_0^1(\Omega)$ such that:

$$\int_\Omega \nabla u \cdot \nabla \delta u \, d\Omega = \int_\Omega f \delta u \, d\Omega, \quad \forall \, \delta u \, \in \, H_0^1(\Omega) \tag{57}$$

where $H_0^1(\Omega)$ is the usual Sobolev functional space. The Hermite-NEM interpolation just described is used for approximating the trial and test functions u and δu, respectively, which are built with the only contribution of internal nodes.

The error using the energy norm is computed according to:

$$\left\| \mathbf{u} - \mathbf{u}^h \right\|_{E(\Omega)} = \left(\frac{1}{2} \int_\Omega (\nabla u^{ex} - \nabla u^h)^T (\nabla u^{ex} - \nabla u^h) \right)^{1/2} \tag{58}$$

For the evaluation of both Equation (57) and (58), the Voronoi cells associated with each node are triangulated and a Gauss quadrature scheme is applied in each subtriangle, using 3, 6 and 12 integration points. Figure 8 compares the accuracy of the Hermite-NEM (H-NEM) approximation with the standard Sibson-NEM. If only three Gauss points quadrature is used, the accuracy of the H-NEM exceeds the accuracy of the NEM, but the difference in the convergence rate is not significant. If a fine enough quadrature scheme is applied (6 points or more), the H-NEM reaches, as expected, a second-order convergence rate.

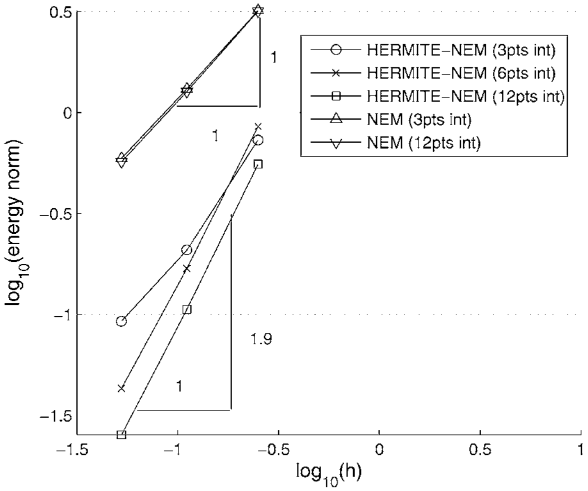

Fig. 8. Convergence analysis using the energy norm for the 2D Poisson's problem.

2.4 Hierarchical Bubbles Functions in the Natural Element Method

The interest of this approach lies in the construction of mixed approximations to be applied in the stable discretization of mixed variational formulations as encountered in mechanics of incompressible media where the mixed approximation must verify the well known LBB condition.

Consider an open bounded domain $\Omega \in \Re^{dim}$ with boundary Γ, dim being the space dimension. Assume that Ω is discretized by a set of nodes S. Let $D(S)$ the simplicial complex associated with the Delaunay tesselation of S. A simplicial complex K in $\Re^{dim}$ is a collection of simplices (hypertetrahedra) in $\Re^{dim}$ such that:

(i) Every face of a simplex K is in K;
(ii) The intersection of any two simplices of K is a face of each of them [15].

If we denote F_k the set of $k - simplices$ $(0 \leq k \leq 3)$, in R^3 the Delaunay tessellation $D(S)$ will be defined as the simplicial complex defined by the tetrahedra in F_3, the triangles in F_2, the edges in F_1, and the vertices in F_0. We denote these collections $T(S)$, $F(S)$, $E(S)$ and $V(S)$, respectively.

In order to construct richer approximations, new shape functions can be associated with the different *k-simplices*. The case $1 < k < 3$ is related to the concept of hierarchical methods [27]. The concept of hierarchical bubble shape functions is a very simple way to construct richer approximations. The extension to meshfree methods is not an easy matter in general, in the absence of topology related to some elements. In the natural element, the underlying Delaunay triangulation allows the use of such approach.

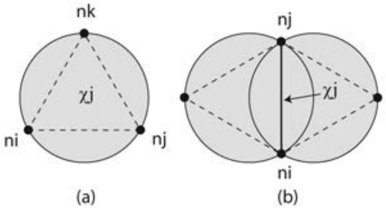

Fig. 9. Supports of the bubble shape functions associated with the Delaunay k-simplex; (a) support of a Delaunay triangle $\chi_j\ n_i - n_j - n_k$; (b) support of a Delaunay edge $\chi_j\ n_i - n_j$.

The key idea is to associate new shape functions to the *k-simplices* of the Delaunay tessellation, i.e. tetrahedra $T' \in T(S)$, triangular facets $F' \in F(S)$ and edges connecting two nodes in the Delaunay triangulation $E' \in E(S)$ [26].

2.4.1 b-NEM Approximation

A k-simplex (K-S) (vertex, edge, triangular facet or tetrahedron) is generated by $K = k + 1$ vertices ($k = 0$, 1, 2 and 3, respectively). The bubble shape function of an entity χ_j generated by K vertices is computed like:

$$\phi_j^*(\mathbf{x}) = \prod_{p=1}^{K} \phi_p(\mathbf{x}) \tag{59}$$

where $\phi_p(\mathbf{x})$ is the NEM shape function associated with node n_p computed at point $\mathbf{x}$.

The support of a K-S generated by K vertices (nodes) in S is the union of the Delaunay spheres containing the K nodes. It results, in 2D:

(i) if χ_j is a Delaunay triangle ($\chi \in F(S)$) ($k = 2$), the support of χ_j is composed with one circle containing the 3 generating nodes of the triangle (see fig. 9 (a));
(ii) if χ_j is an edge of a Delaunay triangle ($\chi \in E(S)$) ($k = 1$) , the support of χ_j is composed with the union of two circles (if $\chi_j \notin \Gamma$), or one circle if $\chi_j \in \Gamma$ (see Figure 9(b)), containing the 2 generating nodes of χ_j.

We now consider the following approximation scheme:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i(\mathbf{x})\,\mathbf{u}_i + \sum_{j=1}^{m} \phi_j^*(\mathbf{x})\,\gamma_j \tag{60}$$

where n is the number of natural neighbors of point $\mathbf{x}$, $\phi_i(\mathbf{x})$ is the NEM shape function related to node $n_i \in S$ computed at point $\mathbf{x}$, $\phi_j^*(\mathbf{x})$ is the bubble shape function defined in Equation (59) associated with the m influent K-S, and γ_j is an additional degree of freedom.

Remarks.

(i) Different combinations can be chosen for enriching the approximation, i.e. using only bubble functions associated with the edges, with the Delaunay triangles, or both.

(ii) The evaluation of the bubble shape functions associated with the K-S is not costly as it only requires the product of available NEM shape functions computed at point $\mathbf{x}$.

(iii) Despite that the approximation scheme defined in Equation (60) is richer than standard NEM approximation, it does not satisfy any polynomial reproducing property other than the linear consistency.

(iv) In this paper, two approximations schemes are investigated and compared: (i) one using bubble functions associated with the Delaunay triangles (that we call b1-NEM); and (ii) one using bubble functions associated with the Delaunay edges (called b2-NEM).

2.4.2 b-NEM with Reproducing Properties

In this section we proceed to correct the shape functions previously constructed defining the approximation scheme (60) within a standard moving least squares framework, in order to evaluate the benefits provided by the higher approximation consistency. The MLS procedure has been summarized in the first section. Let $w_i(\mathbf{x})$ some weight function either associated with a standard or a bubble-NEM shape function, i.e. $w_i(\mathbf{x}) = \phi_i(\mathbf{x})$ or $w_i(\mathbf{x}) = \phi_i^*(\mathbf{x})$, computed at point $\mathbf{x}$.

The MLS procedure leads to:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \tag{61}$$

where we can identify the vector containing the approximation shape functions:

$$\boldsymbol{\psi}^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \tag{62}$$

As just commented, the reproducing b-NEM shape functions are computed by setting $w_i(\mathbf{x}) = \left\{\phi_i(\mathbf{x});\ \phi_j^*(\mathbf{x})\right\}$, $\phi_i(\mathbf{x})$ and $\phi_j^*(\mathbf{x})$ being the shape functions defined in (29) and (59).

Remark. The main difference between the reproducing-b-NEM and the b-NEM without additional reproducing properties is that physical coordinates must be associated with each K-S shape function, in order to evaluate the terms $p_j(\mathbf{x}_i)$ and $p_k(\mathbf{x}_i)$ in Equations (5) and (6). A simple solution is to consider the K-S centroid coordinates.

In the following, the b1-NEM and b2-NEM schemes just described are corrected using the MLS procedure just described. In the most unfavourable case a point $\mathbf{x}$ is influenced by four shape functions in the b1-NEM (3 NEM shape functions, and 1 bubble shape function associated with the Delaunay triangle), and being these weight functions independent, the method is stable if the basis $\mathbf{p}^T(\mathbf{x})$ contains 4 monomials. We call b1-NEM$^+$ the enrichment of the b1-NEM from $\mathbf{p}^T(\mathbf{x}) = \{1, x, y, xy\}$. Following similar assumptions, b2-NEM$^+$ results from the enrichment of the b2-NEM using $\mathbf{p}^T(\mathbf{x}) = \{1, x, y, xy, x^2, y^2\}$.

We have shown in [26] that essential boundary conditions can be enforced directly in all the proposed approximation schemes, as the bubble-NEM shape functions vanish over all external boundaries. For further details, see the proofs for the different schemes in that paper.

2.4.3 Natural Element Discretization

We consider the usual mixed variational formulation of the incompressible linear elastostatics problem where displacement trial and test functions are interpolated using the same shape functions, as the same for the pressure trial and test functions. In the following, the pressure is interpolated using the standard (Sibson) NEM shape functions, while the displacements are interpolated using the b-NEM or the b-NEM$^+$ shape functions previously defined (see [26] for more details).

In order to perform the inf-sup test a sequence of successive refined meshes is considered (uniform distributions) according to the procedure proposed in [2, 8]. The objective is to monitor the inf-sup values, $\lambda_{\min}$, when h decreases. If $\log \lambda_{\min}$ decreases with $logh$ the approximation scheme does not pass the LBB numerical test, which requires that $\log \lambda_{\min}$ remains bounded by a positive constant when $\log h$ decreases.

Figure 10 shows numerical test comparing some mixed NEM approximation schemes, i.e. b-NEM/NEM, NEM/Thiessen [21] (NEM approximation for the displacements and constant pressure within each Voronoi cell), and the P1/P0 and P2/P1 mixed FEM approximation schemes. The FEM computations are carried out using directly the Delaunay triangles. As claimed in other previous works [10], the mixed NEM/Thiessen approximation scheme does not pass the numerical inf/sup test. The mixed FEM P1/P0 also violates the LBB condition [8]. All the bubble-NEM schemes are clearly LBB compliant, being the results similar to the ones computed by using the P2/P1 FEM, which satisfy the LBB condition.

As noticed previously, the NEM shape functions only possesses linear completeness [22]. The enrichment of bubble in the context of MLS does not seem to increase the convergence rate with standard integration despite the proved increase in the approximation consistency. The reasons of this strange behavior constitutes a work in progress.

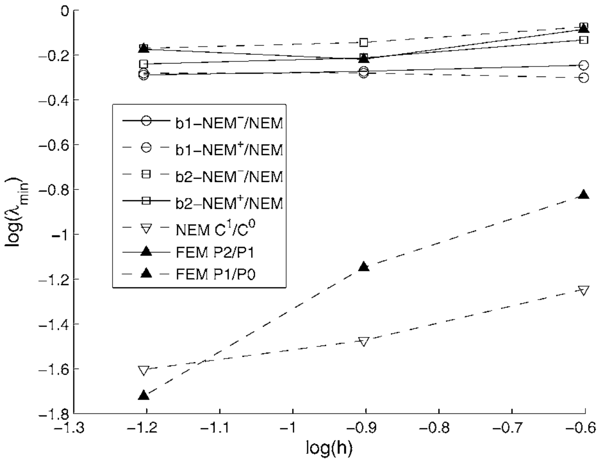

Fig. 10. inf-sup numerical test.

2.5 Natural Neighbor Interpolation with Discontinuous Derivatives

To define NN-approximations with discontinuous derivatives we could proceed in the context of the partition of unity (as in the extended finite element technique) [1]. However, in this work we propose an enrichment that does not involve additional degrees of freedom. For this purpose we start introducing the enriched reproducing kernel particle method, that by introducing the NN-interpolation as kernel function leads to NN-interpolation functions with discontinuous derivatives.

2.5.1 Introducing NN Approximations into E-RKPM: The Enriched NEM (E-NEM)

We consider a level set description $\Theta(\mathbf{x})$ of an interface where the field normal derivatives (with respect to the interface) are discontinuous. Now, we can introduce as enrichment function $u^e(x)$ the following function:

$$u^e(\mathbf{x}) = H_0(\Theta(\mathbf{x}))\Theta(\mathbf{x}) \tag{63}$$

where

$$\Theta(\mathbf{x}) = \begin{cases} \Theta(\mathbf{x}) < 0 \text{ if } \mathbf{x} \in \Omega_1 \\ \Theta(\mathbf{x}) > 0 \text{ if } \mathbf{x} \in \Omega_2 \\ \Theta(\mathbf{x}) = 0 \text{ if } \mathbf{x} \in \Gamma_d \end{cases} \tag{64}$$

and

$$\begin{cases} H_0(\Theta(\mathbf{x})) = 1 \; if \;\; \Theta(\mathbf{x}) \geq 0 \\ H_0(\Theta(\mathbf{x})) = 0 \; if \;\; \Theta(\mathbf{x}) < 0 \end{cases} \tag{65}$$

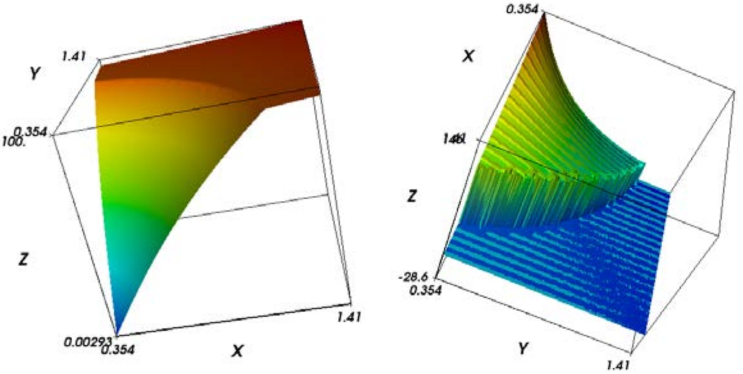

Fig. 11. (Left) Enriched Natural Neighbor approximation with discontinuous normal derivatives across a circular interface. (Right) x-derivative of the temperature field.

Now, we consider a linear consistency enriched with the function given by Equation (63) and the kernel function $w(\mathbf{x} - \mathbf{x}_i, h) = \phi_i(\mathbf{x})$ (the natural neighbor shame functions). The resulting approximation shape functions have the linear consistency but allows also to reproduce discontinuous normal derivatives across the interface Γ_d.

To illustrate the capabilities of the proposed technique we consider the exact solution of the Laplace's problem (modelling the temperature distribution in a steady heat transfer problem) defined in a bi-material consisting of two cylinders with different thermal conductivities. The reproduction tests have been carried out using the E-RKPM as well as the E-NEM, where the circular interface was modelled from the distance to that interface that multiplies the Heaviside's function related to that distance. Figure 11 illustrate a detail of the reconstructed temperature field where we can notice the accurate interface description. The discontinuity in the field derivatives is accurately accounted, as suggested by the representation of the x-derivative depicted in Figure 11.

Finally, in order to quantify the results accuracy we compare in Figure 12 the error (using the two usual norms) using the E-RKPM and the E-NEM techniques. In Figure 12 (right) we can notice that the E-NEM error is not affected by the slope change across the interface, that increases with the difference of thermal conductivities (for $K1 = 10$ the conductivities ratio is 10 whereas it is of 100 for $k1 = 100$.

3 Conclusions

We have explored the connections between NN and MLS approximations, coming from the introduction of the NN approximation functions as the weights in the scope of MLS. Thus, we can adjust the approximation consistency (with the possibility

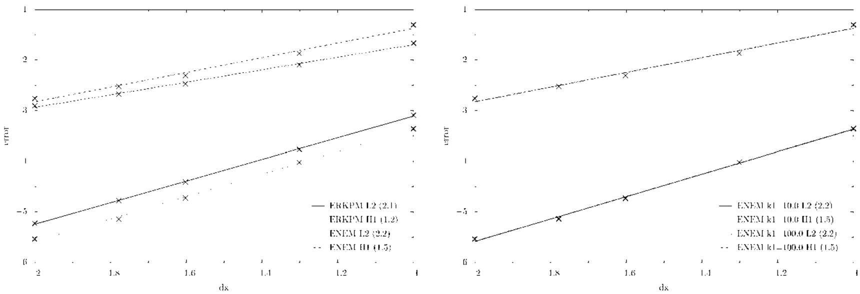

Fig. 12. (Left) Approximation errors using the E-RKPM and the E-NEM. (Right) E-NEM approximation error for different conductivities ratios.

to enrich the approximation basis with some particular functions describing issues of the searched solution) in the framework of the MLS techniques, imposing exactly essential boundary conditions from the use of the NN approximation as MLS weight.

This approach opens, as proved in the present paper, the way to a wide range of formulations: (i) NN collocation strategies; (ii) faster natural element discretizations; (iii) Hermite natural element formulations; (iv) hierarchical bubbles functions in the natural element method; and (v) and NN enriched approximations.

References

1. Babuška I. and Melenk J.M. *Computer Methods in Applied Mechanics and Engineering*, 4:289–314, 1996.
2. Bathe K.J. *Finite Element Procedures*. Prentice Hall, 1986.
3. Belikov V.V., Ivanov V.D., Kontorovich V.K., Korytnik S.A. and Semenov A.Y. *Computational Mathematics and Mathematical*, 37:9–15, 1997.
4. Belytschko T., Lu Y.Y. and Gu L. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.
5. Belytschko T., Krongauz Y., Organ D., Fleming M. and Krysl P. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.
6. Breitkopf P., Touzot G. and Villon P. *Computational Mechanics*, 25:199–206, 2000.
7. Breitkopf P., Chinesta F., Villon P., Rassineux A. and Yvonnet J. A mixed natural neighbor and diffuse element framework for meshfree methods development. In *Computational Mechanics WCCM VI*, in conjunction with APCOM'04, Beijing, China, Tsinghua University Press & Springer-Verlag, 2004.
8. Chapelle D. and Bathe K.J. *Computers & Structures*, 48:745–760, 1993.
9. Cueto E., Doblaré M. and Gracia L. *International Journal for Numerical Methods in Engineering*, 49:519–546, 2000.
10. Gonzalez D., Cueto E. and Doblare M. *International Journal for Numerical Methods in Engineering*, 61:611–632, 2004.
11. Hiyoshi H. and Sugihara K. *Computational Geometry*, 22:167–183, 2002.

12. Liu W.K., Jun S. and Zhang Y.F. *International Journal for Numerical Methods in Fluids*, 21:1081–1106, 1995.
13. Liu W.K., Chen Y., Jun S., Chen J.S., Belytschko T., Pan C., Uras R.A. and Chang C.T. *Archives of Computational Methods in Engineering: State of the Art Reviews*, 3:3–80, 1996.
14. Lucy L.B. *The Astronomic Journal* 88:1013–1024, 1977.
15. Munkres J.R. *Elements of Algebraic Topology*. Perseus Press, 1993.
16. Nayroles B., Touzot G. and Villon P. *Computational Mechanics*, 10:307–318, 1992.
17. Piper B. *Computing Suppl.* 8:227–239, 1993.
18. Rassineux A., Villon P., Savignat J.M. and Stab O. *International Journal for Numerical Methods in Engineering*, 49:10–20, 2000.
19. Sambridge M., Braun J. and McQueen M. *Geophys. J. Int.*, 122:837–857, 1995.
20. Sukumar N., Moran B. and Belytschko T. *International Journal for Numerical Methods in Engineering*, 43:839–887, 1998.
21. Sukumar N. The natural element method in solid mechanics. Ph.D. Thesis, Northwestern University, Evanston, IL, 1998.
22. Sibson R. *Math. Proc. Camb. Phil. Soc.*, 87:151–155, 1980.
23. Yvonnet J., Chinesta F., Lorong P. and Rynckelynck D. *International Journal of Thermal Sciences*, 44:559–569, 2005.
24. Yvonnet J., Ryckelynck D., Lorong P. and Chinesta F. *International Journal for Numerical Methods in Engineering*, 60:1451–1474, 2004.
25. Yvonnet J. and Chinesta F. An hybrid element free Galerkin and natural element Meshfree method for direct imposition of boundary conditions and faster three-dimensional computations. In *Third MIT Conference on Computational Fluid and Solid Mechanics*, MIT, Cambridge, MA, 2005.
26. Yvonnet J., Villon P. and Chinesta F. *International Journal for Numerical Methods in Engineering*, in press, 2006.
27. Zienkiewicz O.C., De J.P., Gago S.R. and Kelly D.W. *Computers & Structures*, 16:53–65, 1983.