# A hybrid element-free Galerkin and natural element meshfree method for direct imposition of boundary conditions and faster three-dimensional computations

J. Yvonnet*, F. Chinesta

*LMSP (Laboratoire de Mécanique des Systèmes et des Procédés) – UMR CNRS-ENSAM-ESEM Ecole Nationale Supérieure d'Arts et Métiers, 151 boulevard de l'Hôpital, F-75013 Paris, France*

**Abstract**

The natural neighbor meshfree method, or natural element method (NEM) provides equivalent quality compared to quadrangular or hexahedral finite elements but is based on the automatic technology of Delaunay triangulation arithmetic. Since the NEM shape functions possess the Kronecker delta property as well as strict linearity over the boundaries of the domain, the essential boundary conditions can be directly implemented with ease as in the conventional finite element method (FEM). Nevertheless, the extension of the method in 3D is complex and costly, involving many geometric constructions in the Vornoi diagram of the set of nodes. In this paper a new method combining the methodology of element-free Galerkin and natural neighbors is proposed to simplify the implementation and reduce the costs of 3D NEM, as well as to provide automatic connectivity in the element-free Galerkin. A 3D Poisson problem is proposed to evaluate the numerical solution as well as the computational costs.

*Keywords:* Natural element method; Element-free Galerkin; Meshfree methods; 3D

## 1. Introduction

In some fields of computational mechanics, including metal forming processes simulation (cutting, shearing, forging), polymer injection, Lagrangian fluid flows simulations, the initial finite element (FE) mesh is quickly distorted, leading to a poor quality associated with distorted elements. The meshless methods [1,2,3] have been proposed to avoid the numerical difficulties of remeshing in the FEM, by constructing the solution entierely in terms of nodes. Nevertheless, some issues associated with boundary condition enforcement, robustness (choice of appropriate shape function support size) and material discontinuities still remain. The natural element method (NEM) [4] avoids these difficulties due to the salient features of the shape functions, which are interpolant, strictly linear over the boundaries of the domain, possess linear consistency and compact support, based on the Delaunay spheres. In this paper, pseudo-NEM shape functions are proposed to simplify the implementatation of 3D NEM and to reduce the computational costs. For this purpose, a particular weight function, which possesses the features of the NEM shape functions, is introduced in the EFG methodology, avoiding the numerous geometric computations involved in 3D NEM calculations. A 3D Poisson problem is proposed to evaluate the numerical solution as well as the computational costs.

## 2. Natural element meshfree method

The natural neighbor Galerkin, or natural element meshfree method (NEM) has been introduced by Traversoni [5] and Sambridge et al [6]. This particular meshfree method uses the natural neighbor coordinates introduced by Sibson [7]. The natural neighbor shape functions possess some remarkable properties such as: (a) interpolation; (b) linear consistency; and (c) strict linearity over the convex boundary of the domain. The interested reader can refer to Sukumar et al. [4] for a detailed description and proofs of these properties. The properties (a) and (c) allow the direct imposition of

* Corresponding author. Tel.: +33 1 4424 6441; Fax: +33 1 4424 6468; E-mail: julien.yvonnet@paris.ensam.fr


*Computational Fluid and Solid Mechanics 2005*
K.J. Bathe (Editor)

essential boundary conditions, unlike in the vast majority of meshfree methods. The extension of the method for non-convex domains have been proposed in [8] or [9]. The natural element and its extensions have been applied in many problems to overcome the meshing problems in FEM, involving, among many other works, fluid flows [10], metal forming [11], or phase change problems [12]. The natural neighbor shape functions are constructed on the basis of the Voronoi diagram of the set of nodes, and the dual Delaunay triangulation.

The Sibson shape function of a node $n_i$ is defined like the Lebesgue measure (area in 2D, volume in 3D) of the overlap of the Voronoi cell associated with the node $n_i$ and the Voronoi cell of the point $x$ (typically an integration point). If we consider the 2D example depicted on Fig. 1, the shape function associated with the node $n_i$, computed at point $x$, is given by:

$$\phi_i(\mathrm{x}) = \frac{Area(abcd)}{Area(bcef)} \tag{1}$$

In the above equation, the $n$ influent nodes, sharing a Voronoi cell with the point $x$, are called the natural neighbors. The support of these shape functions is the union of the circumspheres associated with the Delaunay tetraedra connected to node $n_i$ (see Fig. 1). The global form of unknown variable approximation $u^h$ of point $x$ can be written as:

$$u^h = \sum_{i=1}^{n} \phi_i(\mathrm{x}) u_i \tag{2}$$

Some other natural neighbor shape functions exist, see the works of Belikov et al. [13] or Hiyoshi and Sugihara [14] for a mathematical generalization of this class of methods. Nowadays, some powerful and efficient algorithms [15] in $O(n)$ complexity are available for the construction of the Delaunay triangulation that is unique for a given cloud of nodes. Neverthless, 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 $x$ involves the following steps: (a) find the natural neighbor of the point $x$; (b) construct the new Voronoi cell associated with point $x$; (c) compute the volumes or areas associated with Voronoi cells' entities used in the shape function computations; and (d) compute the shape functions. The step (a) can be performed in constant time by performing a local search in the Voronoi diagram. In our experience, the steps (b) and (c) are an important part in the computation. In the next section, we propose new pseudo-natural neighbor shape functions which avoid the geometric operations and thus steps (b) and (c). The key idea is to introduce a particular weight function in classical meshless element-free Galerkin (EFG) shape functions computations that possesses the salient features of the natural neighbor shape functions.

## 3. Pseudo-natural neighbor shape functions

In the above, we briefly recall the EFG [2] approximation. Let:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{3}$$

with $\mathbf{p}^T(\mathbf{x})$ a polynomial basis, i.e. $\mathbf{p}^T(\mathbf{x}) = [1, x, y, z]$ for a linear basis in 3D, and $\mathbf{a}(\mathbf{x})$ a vector of unknown coefficients ($a_j$, $j = 1,2,3,4$). In order to determine $\mathbf{a}(\mathbf{x})$, a functional $J$ has to be minimized with respect to $\mathbf{a}$, expressed by:

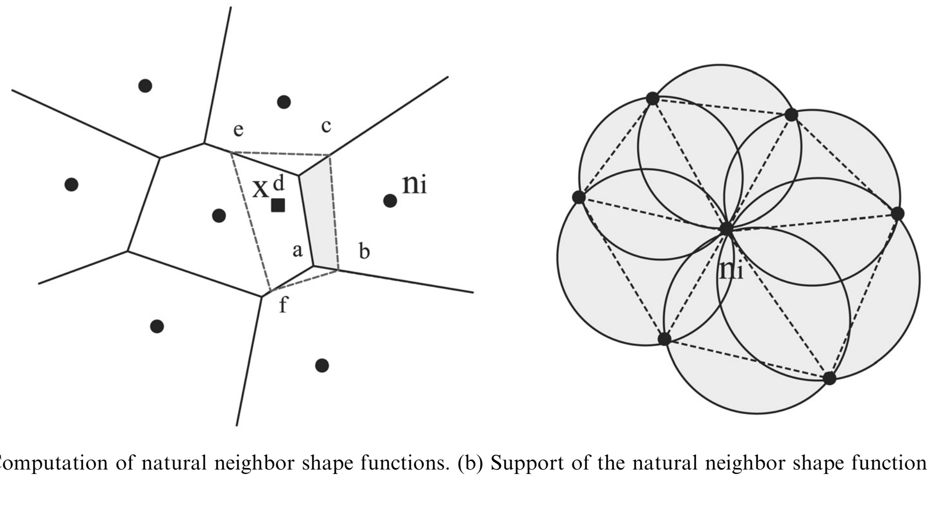


Fig. 1. (a) Computation of natural neighbor shape functions. (b) Support of the natural neighbor shape function of node $n_j$.

$$J = \frac{1}{2}\sum_{i=1}^{n} w_i(\mathbf{x})\left[\mathbf{p}^T(\mathrm{x})\mathbf{a}(\mathbf{x}) - u_i\right]^2 \tag{4}$$

where $u_i$ are the nodal unknown associated with neighbors of point $\mathbf{x}$. Minimizing $J$ with respect to unknown coefficients $a_j$ ($j = 1,2,3,4$) leads to:

$$\frac{\partial J}{\partial a_j} = \sum_{k=1}^{n} a_k\left[\sum_{i=1}^{n} w_i(\mathbf{x})p_j(\mathbf{x}_i)p_k(\mathbf{x}_i)\right] - \sum_{i=1}^{n} w_i(\mathbf{x})p_j(\mathbf{x}_i)u_i = 0 \tag{5}$$

which leads to the linear system:

$$\mathbf{Aa} = \mathbf{Bu}_i \tag{6}$$

Where the matrix $\mathbf{A}$ and $\mathbf{B}$ are defined by:

$$A_{jk} = \sum_{i=1}^{n} w_i(\mathbf{x})p_j(\mathbf{x}_i)p_k(\mathbf{x}_i) \tag{7}$$

$$B_{ij} = w_i(\mathbf{x})p_j(\mathbf{x}_i) \tag{8}$$

Substituting $\mathbf{a}$ in Eq. (6) leads to:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}\mathbf{Bu}_i \tag{9}$$

By identification, the new shape functions are given by:

$$\phi(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}\mathbf{B} \tag{10}$$

In the above, $\phi(\mathbf{x})$ is a vector containing the shape functions associated with neighbors of point $x$. In the following, we aim at defining an appropriate weight function, $w_i(\mathbf{x})$, such as the resulting shape functions satisfy the following properties (a) the Kroenecker delta property ($\phi_i(x_j) = \delta_{ij}$); (b) the linear consistency (which is automatically satisfied according to Eq. (3)); (c) the shape functions are non-zeros over the Delaunay spheres containing the node $n_i$ (see Fig. 1).

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 containaing the point $x$, and where the projection of the tip matches the node $n_i$ (see Fig. 2).

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

$$f(\mathbf{x}) = \frac{\|\mathbf{n_iP}\| - \|\mathbf{n_ix}\|}{\|\mathbf{n_iP}\|} \tag{11}$$

With:

$$\mathbf{n_iP} = -2\left(\frac{\mathbf{cn_i}\cdot\mathbf{n_ix}}{\mathbf{n_ix}\cdot\mathbf{n_ix}}\right)\mathbf{n_ix} \tag{12}$$

In order to avoid the overlapping of cone functions, whereby 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 $x$ belongs 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 $\mathbf{n_in_j}$ and $\mathbf{n_in_k}$ has positive coordinates in this basis. $n_j$ and $n_k$ are the other two vertices of the triangle (see Fig. 3). Due to the particular shape of its support, which is defined for any nodal distribution, this weight function guarantees interpolation conditions ($w_i(\mathbf{x}_j) = \delta_{ij}$); as Delaunay circles passes through the nodes. Furtermore, 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. The algorithm alowing the construction of this weight function is gien in Table 1.

In order to improve the continuity of the weight function over the Delaunay spheres, the weight function can be embedded into a smooth function such as:

$$w_i^*(\mathbf{x}) = 3w_i(\mathbf{x})^2 - 2w_i(\mathbf{x})^3 \tag{13}$$

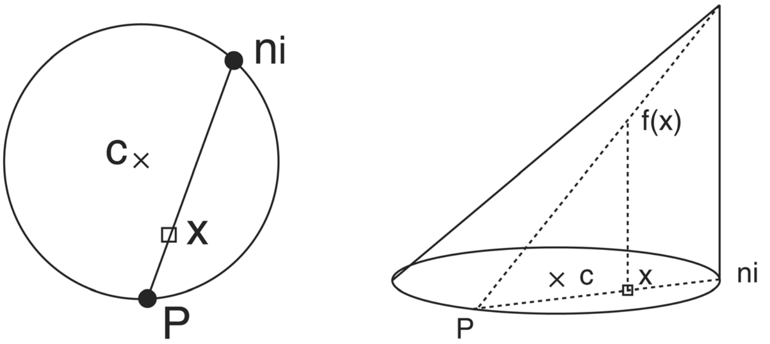


Fig. 2. Cone function.

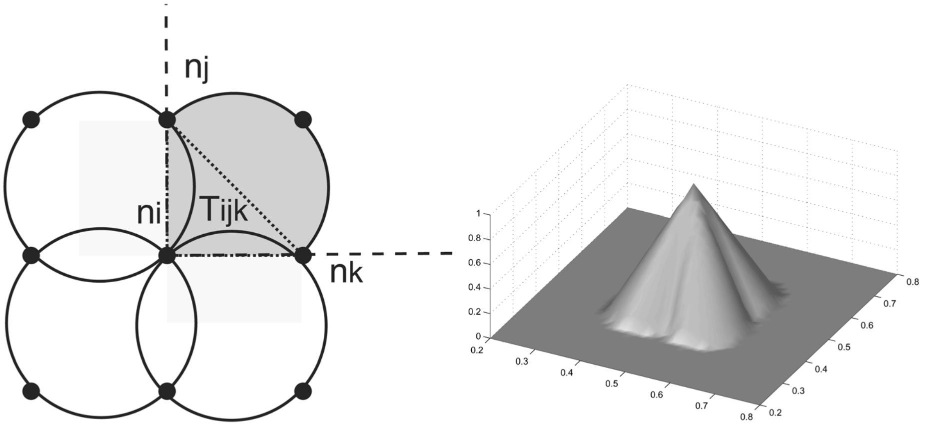


Fig. 3. (a) Zone associated with a cone portion; (b) weight function.

Table 1
Computation of the pseudo-NEM weight function

```
(i) Find natural neighbors of point x, tetraedra and circumspheres containing the point x;

(ii) LOOP over all natural neighbors n_i:
  LOOP over each tetraedron t_j containing the natural neighbors:
    IF the node n_i is one of the tetraedron's t_j vertices:
      IF the point x is in the basis associated with tetraedron t_j:
        w_i(x) = f(x) (see Eq. (11))
      END
    END
  END
END
```

## 4. Numerical example

The folowing Poisson's problem is considered in a unit cube in $\Re^3$:

$$\nabla u = 0 \; in \; \Omega \tag{14}$$

$$\bar{u} = u^{ex}(\mathbf{x}) = T^{ex}(\mathbf{x}) = 2x^2 - y^2 - z^2 \; on \; \Gamma_u \tag{15}$$

where $\Gamma_u$ is the boudary of the domain where the primary variable (essential boundary conditions) are imposed, which leads to the exact solution:

$$T^{ex}(\mathbf{x}) = 2x^2 - y - z^2 \; in \; \Omega \tag{16}$$

The weak form associated with the problem defined in Eq. (14) is given by:

*Find* $\mathbf{u} \in H^1(\Omega)$ ($\mathbf{u} = \bar{\mathbf{u}}$ *on* $\Gamma_u$) *such as:*

$$\int_\Omega \nabla u^* \cdot \nabla u d\Omega = 0, \forall u^* \in H_0^1(\Omega) \tag{17}$$

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

The problem has been solved by using several refined meshes: $3 \times 3 \times 3$, $5 \times 5 \times 5$, $7 \times 7 \times 7$ and $10 \times 10 \times 10$ nodes. For comparison, regular and irregular grids have been used. The energy norm has been computed to determine the convergence of the solution. Results are depicted in Fig. 4. A comparison between the computational times in MATLAB associated with standard NEM shape functions and pseudo-NEM shape functions is depicted in Fig. 5. We can see from these results that the pseudo-NEM (smoothed) possess good quality compared to the NEM, with equivalent rate of convergence. The slightly lower quality is compensated with substantial gains in computational times, and direct imposition of essential boundary conditions in EFG.

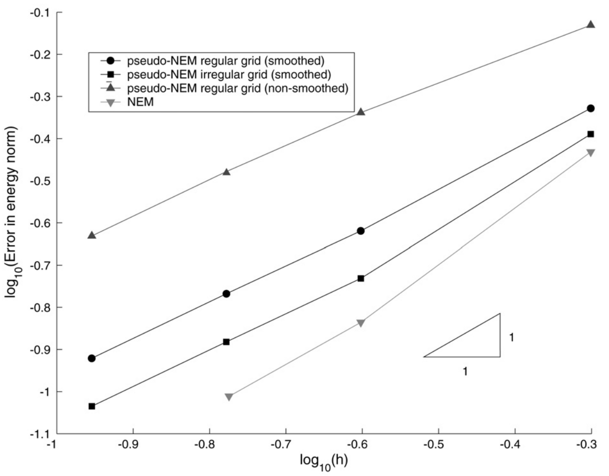


Fig. 4. Convergence for the cube problem.

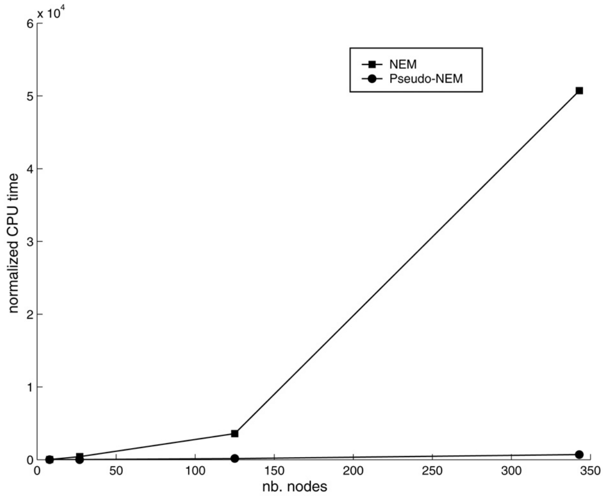


Fig. 5. Comparison of times computations between NEM and pseudo-NEM shape functions.

## 5. Conclusion

In this paper, a specific weight function that posses the salient features of natural neighbor shape functions is used in the element-free Galerkin procedure (but can also be used in a diffuse element method or reproducing the kernel particle method [3] This new method' combines the advantages of natural neighbor method and traditional meshless methods (DEM, EFG, RKPM). The contribution for usual meshless methods are: (a) interpolation properties which allows the direct imposition of essential boundary conditions; (b) direct connectivity defined by the natural neighbors. The contribution for the NEM is a lower computational time associated with the shape function computations, and a simplified computational procedure, involving no geometric operations, which is specially advantageous for 3D practical applications. Substantial benefits can be optain in time computations, and convergence rate of the proposed method is in the same order with the NEM.

## References

[1] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics 1992;10:307–318.

[2] Belytschko T, Lu YY, Gu L, Element-free Galerkin methods. Int J for Numer Methods in Engineering 1994;37:229–256.

[3] Liu WK, Jun S, Zhang YF. Reproducing Kernel particle methods. Int J Numer Meth Fluids 1995;21:1081–1106.

[4] Sukumar N, Moran B, Belytschko T. The natural elements method in solid mechanics. Int J for Numer Meth in Eng 1998;43:839–887.

[5] Traversoni L. Natural neighbor finite elements. In: International Conference on Hydraulic Engineering Software Proc 1994;2:291–297, Computational Mechanics Publications.

[6] Sambridge M, Braun J, McQueen M. Geophisical parameterization and interpolation of irregular data using natural neighbors. Geophysi J Int 1995;122:837–857.

[7] Sibson R. A vector identity for the Dirichlet tesselations. Math Proc Camb Phil Soc 1980;87:151–155.

[8] Yvonnet J, Ryckelynck D, Lorong P, Chinesta P, A new extension of the natural element method for non convex and discontinuous problems, the constrained natural element method (C-NEM) Int J for Numer Meth in Eng 2004;60:1451–1474.

[9] Cueto E, Cegoñino J, Calvo B, Doblaré M. On the imposition of essential boundary conditions in natural neighbor Galerkin methods. Comm in Numer Meth in Eng 2003;19:361–376.

[10] Martinez MA, Cueto MA, Alfaro I, Doblar EM, Chinesta F. Updated Lagrangian free surface flow simulations with natural neighbor Galerkin methods. Int J for Numer Meth in Eng 2004;60(13):2105–2129.

[11] Alfaro I, Yvonnet J, Cueto E, Chinesta F, Doblarè M. Meshless methods with application to metal forming. Computer Meth in Appl Mech and Eng 2004 (accepted for publication).

[12] Yvonnet J, Ryckelynck D, Lorong P, Chinesta P. The constrained natural element method for treating thermal models involving moving interfaces. Int J for Therm Sci 2004 (accepted for publication).

[13] Belikov VV, Ivanov VD, Kontorovich VK, Korytnik SA, Semenov AY. The non-Sibsonian interpolation: a new method of interpolation of the values of a function on an arbitrary set of points. Comp Math and Mathematical Phy 1997;37(1):9–15.

[14] Hiyoshi H, Sugihara K. Improving continuity of Voronoi-based interpolation over Delaunay spheres. Computational Geometry 2002;22:167–183.

[15] Barber CB. Computational geometry with imprecise data and arithmetic PhD thesis, Princeton, NJ, 1993.