A natural neighbour Galerkin method with octree structure

José Javier Laguardia — Elias Cueto — Manuel Doblaré

Aragón Institute of Engineering Research, University of Zaragoza
Edificio Betancourt, María de Luna, 7. E-50018 Zaragoza, Spain
ecueto@unizar.es

ABSTRACT. *We present in this paper a highly structured numerical method based on the employ of natural neighbour interpolation in a Galerkin framework. It employs an octree discretisation of the domain, thus being suitable for numerical simulations in voxelized domains, obtained after image processing, for instance, and medical image techniques in particular, with nearly no user intervention. The accuracy of the method and computational cost are also addressed in this paper. We present some examples that illustrate this behaviour.*

RÉSUMÉ. *Dans cet article, nous présentons une méthode numérique fortement structurée, basée sur l'emploi de l'interpolation de voisinage naturel dans un cadre de Galerkin. Cette méthode utilise une discrétisation « octree » du domaine. Grâce à cela, elle est acceptable pour des simulations numériques sur des domaines voxélisés qui sont obtenus à partir de traitements des images, par exemple, dans les techniques des images cliniques en particulier, où l'intervention de l'utilisateur existe à peine. La précision de la méthode ainsi que son coût informatique sont discutés dans cet article. Nous présentons quelques exemples illustrant ce comportement.*

KEYWORDS: *Natural Neighbour interpolation, structured meshes, octree, voxels, R-functions.*

MOTS-CLÉS : *Interpolation par voisinage naturel, maillage structuré, octree, voxels, fonctions-R.*

REMN – 15/2006. Alleviating mesh constraints, pages 529 to 548

1. Introduction

The tremendous success of meshless methods in the last decade has also generated a renewed interest on methods based on highly structured meshes (Belytschko *et al.*, 2003; Nagashima *et al.*, 2002). This is partly due to the vast effort paid in the study of imposition of essential boundary conditions or numerical integration, for instance. There are, actually, some problems where the presence of a mesh, and even a highly structured mesh, reports many advantages. Examples of these problems can be those in which the data are obtained after some kind of image processing, such as biomedical simulation of living organs, for instance.

Surprisingly, meshless methods have helped to improve these structured methods in an important way and many of the traditional "meshless" techniques have been applied to these highly structured methods. In some sense, in addition, methods that lye in a highly structured mesh that is generated automatically can be considered as "meshless", since they avoid the burden associated to the generation of a mesh, and this is performed in a process transparent to the user. These are among the first characteristics of a meshless method.

There exist other reasons for the study of methods based on a highly structured mesh, such as the speed of computation, for instance. The regularity of the mesh can sometimes help to improve the performance of the method. Some methods have been developed in order to take advantage of the *voxel* structure of image data, for instance.

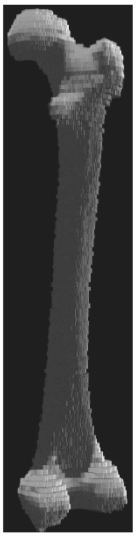


Figure 1. *A finite element mesh of a human femur obtained after treatment of the image so as to convert each voxel into a finite element*

In (van Rietbergen *et al.*, 1996) for instance, a method is developed that converts each *voxel* of a medical image (see Figure 1) into a perfectly hexahedral finite element. This generates a mesh of finite elements of equal shape, size and orientation, whose regularity can be exploited to speed-up computations. The obvious drawback of this technique is the non-conformity of the mesh to the actual geometry of the domain. Since this technique was employed to analyze bone tissue at a micro scale, this lack of conformity was considered to be not much important.

Another important drawback of this kind methods is that no refinement of the mesh can be done. Of course, higher-order FE can be employed, but only if the whole mesh is refined. Otherwise, the so-called *hanging nodes* will appear.

In this paper we study the possibility of developing a new method that could exploit the advantages of the highly structured character of the mesh, while keeping the conformity of the mesh to the domain and incorporating the possibility of mesh refinement, even in the presence of hanging nodes. The key ingredients of this methods are the following:

– we employ natural neighbour interpolation so as to be able to keep the conformity of the method in the presence of *hanging* nodes that appear in the process of mesh adaptive refinement;

– data is stored in a binary tree (*octree*) structure in order to perform fast natural neighbour search;

– essential boundary conditions are imposed through *R*-functions, a special class of functions that vanish at the essential boundary. These functions are extremely easy to construct.

The outline of the papers is as follows. In Section 2 we review the basics of natural neighbour interpolation. In Section 3 we describe the data structure of the method, as presented for the two-dimensional counterpart of this method in (Laguardia *et al.*, 2005). In Section 4 we describe the implemented enforcement of essential boundary conditions and, finally, in Section 5 we include some examples of the performance of the method.

2. Natural neighbour Galerkin methods

2.1. *Natural Neighbour interpolation*

The NEM (Sukumar *et al.*, 1998; Cueto *et al.*, 2003) is a Galerkin procedure based on the natural neighbor interpolation scheme. This interpolation scheme is the key ingredient of the method here proposed. It relies on the concepts of Voronoi diagrams and Delaunay triangulations (see Figure 2), to build Galerkin trial and test functions. These are defined as the Natural Neighbor coordinates of the point under consideration, that is, with respect to Figure 3, the value in the point *x* of the shape function associated with the node 1 is (Sibson, 1980; 1981).

$$\phi_1(x) = \frac{A_{abfe}}{A_{abcd}} \qquad [1]$$

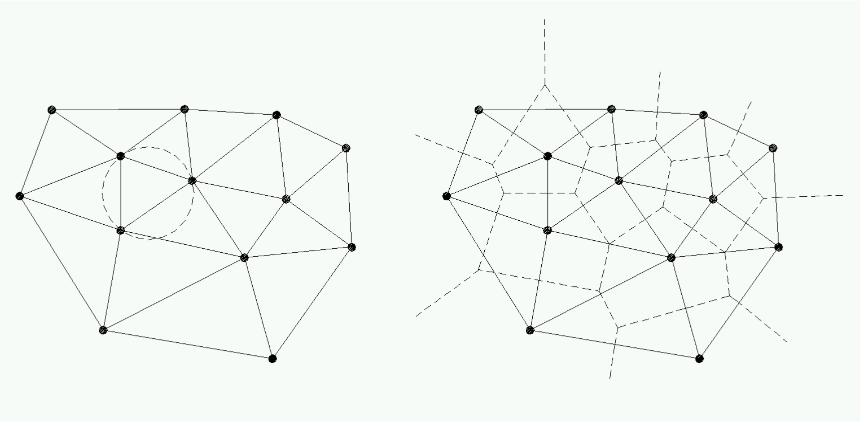


Figure 2. *Delaunay triangulation and Voronoi diagram of a set of points*

In addition, the NEM has other interesting properties such as linear consistency and smooth shape functions. These functions are dependent on the position and density of nodes, leading to standard FE constant strain triangle shape functions, bilinear shape functions or rational quartic functions in different situations (see Figure 4 for a typical shape function). These properties permit an exact reproduction of linear displacement fields on the boundary of convex domains.

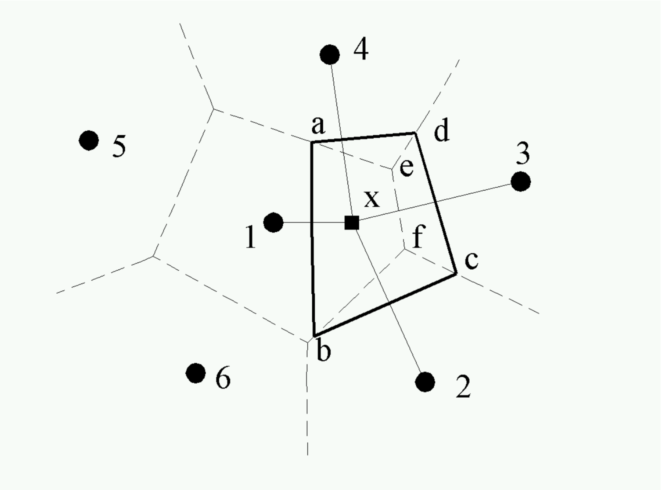


Figure 3. *Definition of natural neighbour coordinates*

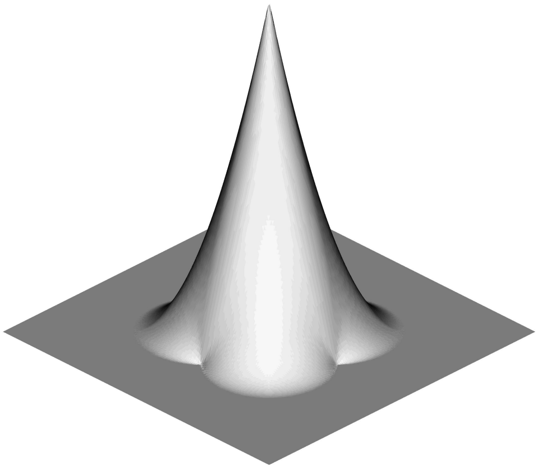


Figure 4. *Typical Sibson shape function (courtesy N. Sukumar)*

2.2. *Governing equations*

The work here addressed is concerned with the problem of linear or nonlinear elastostatics. Consider an open, bounded domain $\Omega \in \mathsf{R}^3$ with boundary $\Gamma=\delta\Omega$. The equilibrium equations reads:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \ \text{ in } \Omega \qquad [2]$$

where $\boldsymbol{\sigma}$ represents the Cauchy stress tensor and $\mathbf{b}$ the vector of body forces. The stress-strain relationship is given by:

$$\boldsymbol{\sigma} = \mathsf{C} : \nabla^s \mathbf{u} \qquad [3]$$

where $\nabla^s \mathbf{u}$ represents the symmetric part of the displacement gradient or, equivalently, the small strains tensor and C the Hooke tensor.

Boundary conditions are:

$$\boldsymbol{\sigma}\mathbf{n} = \bar{\mathbf{t}} \ \text{ in } \Gamma_t \qquad [4]$$

$$\mathbf{u} = \bar{\mathbf{u}} \ \text{ in } \Gamma_u \qquad [5]$$

where $\Gamma_u \cup \Gamma_t = \Gamma$ and $\Gamma_u \cap \Gamma_t = \varnothing$, as usual. $\mathbf{n}$ represents the outward normal vector to the surface of the solid and over-lined magnitudes represent prescribed values.

Let $\mathbf{u} \in \mathcal{S} = \left\{ \mathbf{u} \in H^1 \mid \mathbf{u} = \bar{\mathbf{u}} \text{ in } \Gamma_u \right\}$ and $\delta\mathbf{v} \in \mathcal{V} = \left\{ \delta\mathbf{v} \in H^1 \mid \delta\mathbf{v} = 0 \text{ in } \Gamma_u \right\}$. The weak form of the boundary-value problem can then be stated as:

$$\int_{\Omega} \nabla^s(\delta\mathbf{v}) : \mathbf{C} : \nabla^s \mathbf{u} d\Omega = \int_{\Omega} \delta\mathbf{v} \cdot \mathbf{b} d\Omega + \int_{\Gamma_t} \delta\mathbf{v} \cdot \bar{\mathbf{t}} d\Gamma , \ \forall \delta\mathbf{v} \in \mathsf{V} \qquad [6]$$

The discrete form of the problem [6] is obtained by considering suitable approximations of **u** and δ**v** of the form:

$$\mathbf{u}_i^h = \sum_I \phi_I \mathbf{u}_{Ii} \qquad [7]$$

$$\delta\mathbf{v}_i^h = \sum_I \phi_I \delta\mathbf{v}_{Ii} \qquad [8]$$

where ϕ_I represent the before-mentioned Sibson shape functions, so as to obtain a discrete form of the problem **Ku=f**.

3. Data structure of the method

In order to both speedup computations and to link mechanical models to geometric models, we have implemented our method in an octree (*i.e.*, a three-dimensional binary tree) structure.

3.1. *Description of the model's geometry*

In our method, a model can be described in either *voxel* or STL formats. STL, or stereolithography, format is a file containing a list of triangles describing the surface of the model.

In the first case, the surface of the model could be extracted by taking into account the associated voxel's values of the variable of interest (typically the value of density for hard tissue models, for instance) through the *marching cubes* method (Lorensen and Cline, 1987.) This method is based on the assumption that a given surface can cut a cube in a limited number of patterns (only 15 in three-dimensional cases, see Figure 5).

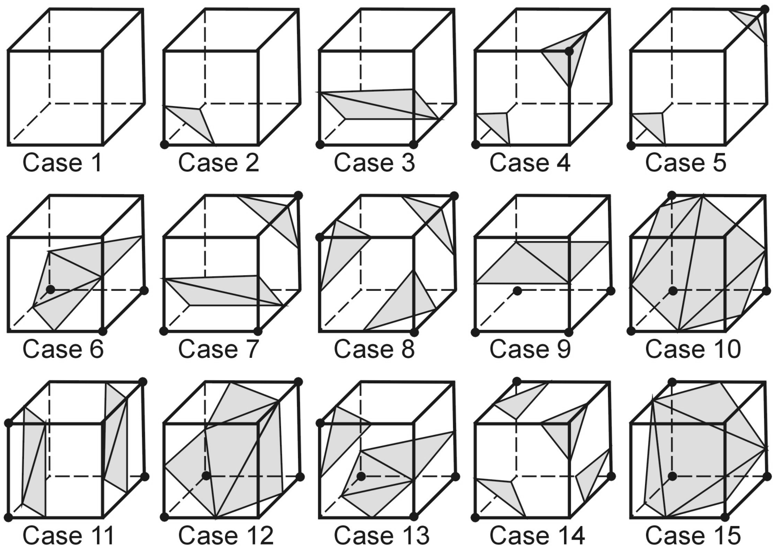


Figure 5. *Marching cubes algorithm*

In the STL format, each triangle is assumed to verify the *marching cubes* assumption. In the case of more than one triangle cutting a given cube, the surface is simplified so as to verify the before mentioned assumption.

Whatever the method employed, model's geometry will be embedded in a cube and recursively subdivided by recursion, thus generating an *octree*. The term octree (Samet, 1984) refers to a class of hierarchical data structures based on the principle of recursive decomposition of space.

Following (Samet, 1984), octrees can be classified attending to:

1) the type of data (in this particular instance, regions of the space);

2) the principle guiding the decomposition process and

3) the resolution, that can be variable or not.

We begin with a cubical volume and recursively subdivide it into eight congruent disjoint cubes, called octants, until a prescribed level of refinement is reached (in STL format) or an uniform cube is obtained (if we employ a voxel model), see Figure 6. In our case, the decomposition process can be guided, for instance, by the curvature of the curve representing the boundary, if the STL format is preferred. One of the big advantages of using an octree structure for representing the domain is that the vast majority of the operations needed for a given cell is performed in a recursive manner, see (Laguardia *et al.*, 2005) for further details. In Figure 7 an example of a simplified octree description of a human tibia and fibula is shown.

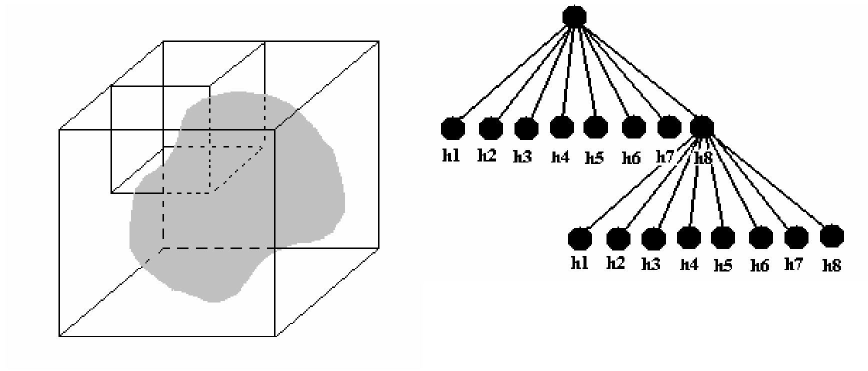


Figure 6. *Octree decomposition of the space and associated data representation*

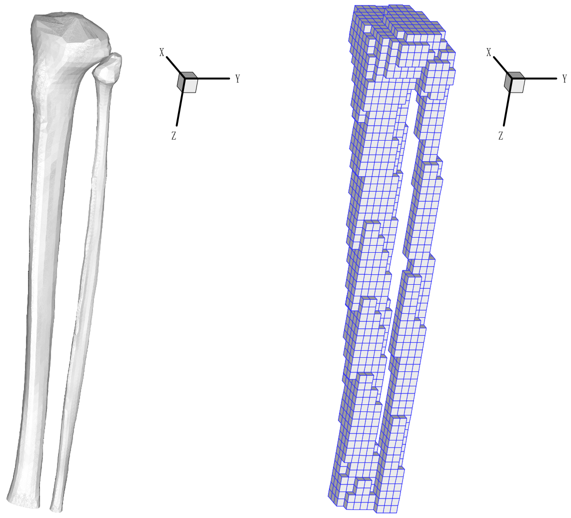


Figure 7. *Octree description of the human tibia and fibula. Left, the faceted description of the geometry. Right, octree model. Only the (at least partially) filled cubes are shown*

3.2. *Interpolation within an octree*

One of the main difficulties of establishing a finite element-like interpolation scheme within an octree structure is the presence of *hanging* nodes, *i.e.*, there are some nodes that do not verify conformity of the approximation.

In the work here presented, the proposed approach takes benefits from the limited number of neighbouring patterns that are possible for a given two- or three-dimensional cell, if the level difference between neighbouring cells is controlled. It has been demonstrated (Klaas and Shephard, 2000) that an octree enhanced by storing neighbour information at each octant is able to provide neighbour information in constant time if the level difference between neighbouring octants is controlled. In this work, this difference has been set to one.

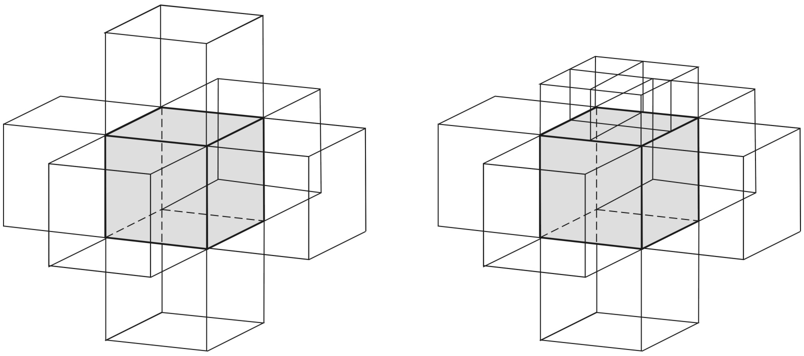


Figure 8. *Possible nodal neighbourhood (3d) for a cell surrounded by other sixt of equal or lower level. Only two examples are presented: uniform refinement (left) and one of the neighbouring cubes refined (right)*

Thus, given the limited number of possible neighbourhood patterns, it is advantageous to store shape functions values at the integration points rather than the element stiffness matrix itself. This also allows us to employ non homogeneous materials. Consider the possible neighbouring patterns for a given octree cell, surrounded by other six of the same or higher level, as shown in Figure 8. If the considered cell is surrounded by lower-level cells, similar patterns can be developed. The number of shape function's values to be stored depends, obviously, on the quadrature scheme used. Note that the consideration of the possible refinement of corner cells is not necessary, since they do not neighbour interior points of the cell (see Figure 8). This reduces the total amount of cases to be considered. In three-dimensional cases, the number of possible neighbourhood patterns is higher than in 2d. In this case, one recently developed possibility is to implement natural neighbour (Sibson) interpolation in hardware (Park *et al.*, 2005).

4. Imposition of boundary and interface conditions

4.1. *Essential (Dirichlet) boundary conditions*

In a non-conforming mesh, such as the octree meshes developed so far, the issue of imposing essential and interface boundary conditions is of utmost importance. In our method we enforce these conditions through a particular instance of the so-called *characteristic function* method (Babuška *et al.*, 2003). If the domain possesses a smooth boundary Γ, there exists a function $\boldsymbol{\omega}$ such that

$$\boldsymbol{\omega} > 0 \text{ in } \Omega \qquad [9]$$

$$\boldsymbol{\omega} = 0 \text{ on } \Gamma = \partial\Omega \qquad [10]$$

$$\left|\nabla\boldsymbol{\omega}\right| \geq \alpha > 0 \text{ on } \Gamma = \partial\Omega \qquad [11]$$

Then, it is possible to construct an approximation of the solution of the form:

$$\mathbf{u} = \mathbf{u}\big|_{\Gamma_u} + \boldsymbol{\omega}\sum_i \phi_i \mathbf{a}_i \qquad [12]$$

that obviously verifies essential boundary conditions. The problem then lies in finding such a function $\boldsymbol{\omega}$ for a general boundary. In (Rvachev *et al.*, 1995) a particular method to construct a particular family of functions $\boldsymbol{\omega}$, called *R*-functions, is developed.

R-functions behave like a toolkit to construct these functions and are used in this work to construct the characteristic functions so as to verify essential boundary conditions. An *R*-function is a real-valued function whose sign is completely defined by the sign of its arguments. Such functions encode boolean operations that help to construct combinations of simple, basis, functions. For instance, consider the following functions (Shapiro and Tsukanov, 1999), that behave like the logical operators *and* and *or*:

$$x \wedge y \equiv (x + y - \sqrt{x^2 + y^2}) \qquad [13]$$

$$x \vee y \equiv (x + y + \sqrt{x^2 + y^2}) \qquad [14]$$

Consider, for instance, the domain shown in Figure 9. The domain is defined by means of a set of six inequalities of the general form

$$f_1 = y \geq y_1 \qquad [15]$$

$$f_2 = x \geq x_2 \qquad [16]$$

$$f_3 = y \geq y_3 \qquad [17]$$

…

These inequalities can be combined to render a definition of the whole domain as:

$$\Omega = (f_1 \vee f_2) \wedge f_3 \wedge f_4 \wedge f_5 \wedge f_6 \qquad [18]$$

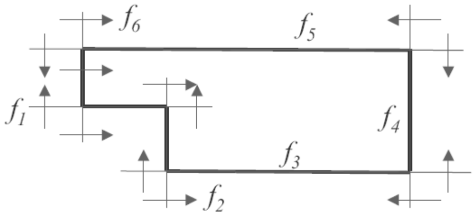
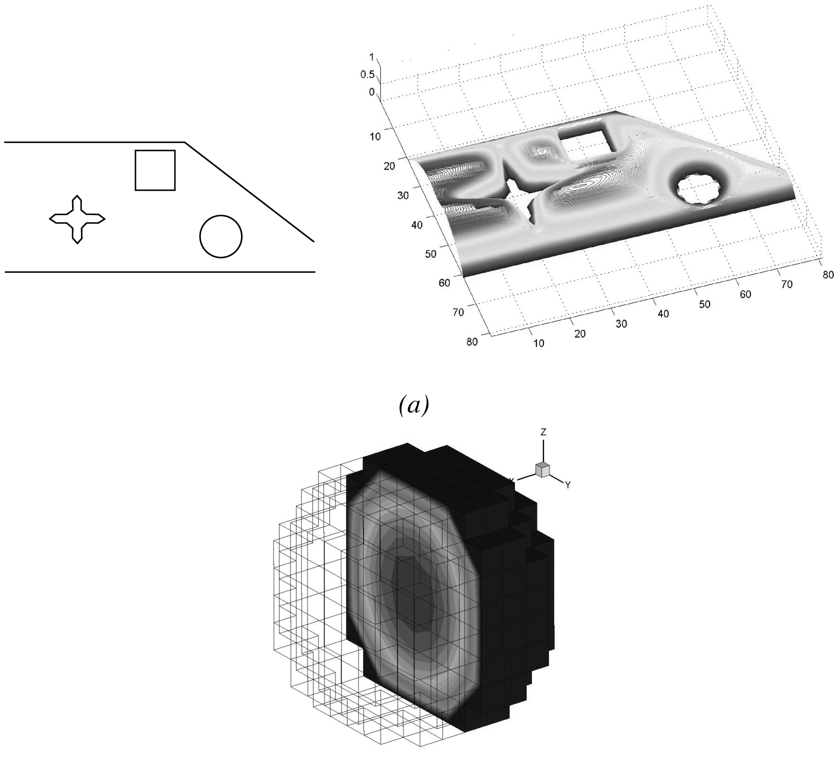


Figure 9. *Definition of the boundary of a given domain as a set of inequalities*



(a)

(b)

Figure 10. *Example of arbitrary essential boundary and resultant R-function. (a) A two-dimensional example, (b) R-function vanishing at the boundary of a sphere in 3D*

For non-homogeneous boundary conditions of the type

$$\mathbf{u}\big|_{\Gamma_u} = \varphi_0 \qquad [19]$$

a function φ such that *trace*(φ)=φ$_0$ can be added to the solution for the homogeneous problem. Assuming that φ$_0$ is approximated by piece-wise linear functions along the boundary, the function φ can be obtained by means of

$$\varphi = \frac{\sum_{i=1}^{m} \varphi_i \prod_{j=1, j\neq i}^{m} \omega_j}{\sum_{i=1}^{m} \prod_{j=1, j\neq i}^{m} \omega_j} \qquad [20]$$

where φ$_i$ represents the prescribed function at each portion of the boundary and ω$_j$ represents an *R*-function that vanishes along the boundary. *m* represents the number of segments in the boundary.

As can be seen in Figure 10, R-functions behave like an approximated distance function and this fact can also be exploited to impose interface conditions in piece-wise homogeneous domains, for instance, through the Partition of Unity method.

4.2. *Natural (Neumann) boundary conditions*

The imposition of inhomogeneous natural boundary conditions deserves a special treatment when meshes not conforming to the domain are employed (see Figure 11). For cells cutting the boundary with non-homogeneous boundary conditions, the term $\int_{\Gamma_t} \delta \mathbf{v} \cdot \bar{\mathbf{t}} d\Gamma$ in Equation [6] should be integrated separately and added to the right-hand side (or force) vector. For this we firstly approximate the boundary facet (segment in 2d) by an appropriate set of triangles (or portion of the planar straight line graph in 2d –slash line in Figure 11–) and perform a three-point Gauss numerical integration on each triangle.

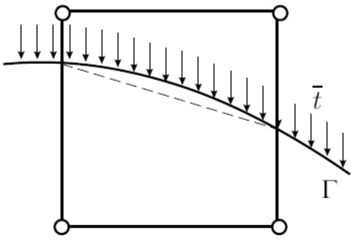


Figure 11. *A cell cut by the natural boundary*

The integration of the weak form at cells with homogeneous natural boundary conditions is straightforward, as proven in (Belytschko *et al.*, 2003). It is only necessary to integrate the weak form only in the portion of the cell within the domain.

5. Numerical results

5.1. *Bi-material beam under bending*

To show the performance of the proposed method we consider here a two-dimensional example of refinement. This example is composed by a bi-material beam subjected to bending. In this case, D = 4 and L = 10, P = 1000. Two distinct regions are considered in the beam, with Young's modulus E1 = 10000 and E2 = 1000000 respectively, for the right and left parts .

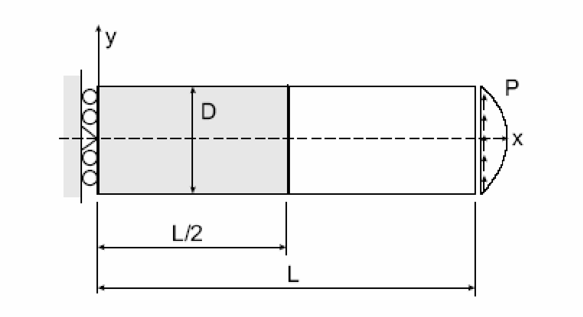


Figure 12. *Geometry of the beam*

In this case, non-uniform quadtree representations were chosen, as shown in Figure 13. Note that the proposed discretisation was non-conforming to the boundary. A sequence of refined meshes was implemented and analysed in order to check the convergence of the method.

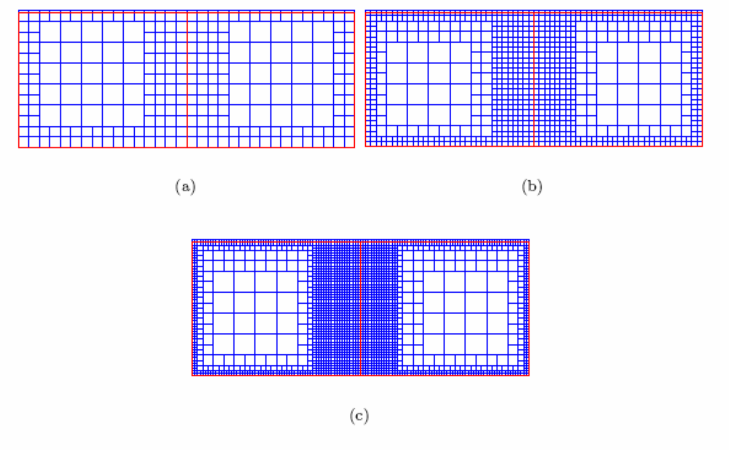


Figure 13. *Quadtree models for the beam problem*

The convergence for this problem is shown in Figure 14, where it can be noticed that it is similar to that of the standard NEM procedure.

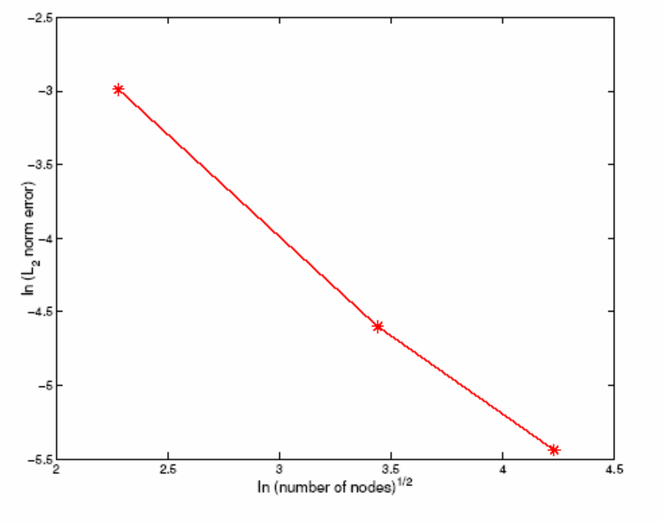


Figure 14. *Convergence plot for the beam problem*

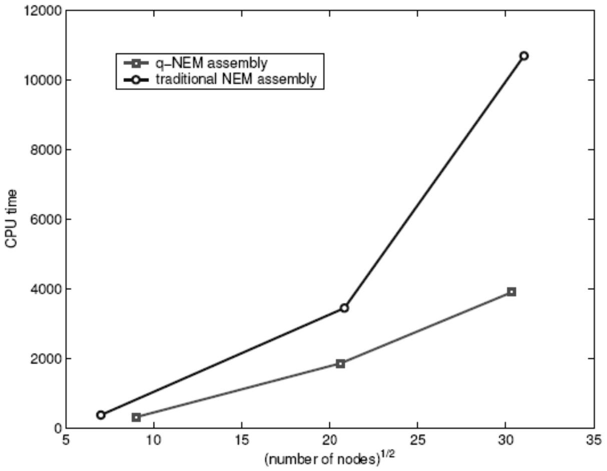


Figure 15. *Computer time for the standard and quadtree versions of the method*

In Figure 15 the computer savings of the proposed method are shown. For a deeper insight in this topic, the interested reader is referred to (Laguardia *et al.*, 2005).

5.2. *Three-dimensional hollow cylinder under pressure*

To prove the accuracy of the proposed method in three-dimensional problems we study the problem of a hollow cylinder under internal pressure, which is classic in the literature. The analytical solution to this problem is:

$$\sigma_\rho = \frac{R_i^2 p}{R_e^2 - R_i^2}\left(1 - \frac{R_e^2}{\rho^2}\right) \qquad [21a]$$

$$\sigma_\theta = \frac{R_i^2 p}{R_e^2 - R_i^2}\left(1 + \frac{R_e^2}{\rho^2}\right) \qquad [21b]$$

$$\sigma_\mathbf{z} = \nu(\sigma_\rho + \sigma_\theta) \qquad [21c]$$

$$\varepsilon_\rho = \frac{1}{E}(\sigma_\rho - \nu\sigma_\theta - \nu\sigma_z) \qquad [22a]$$

$$\varepsilon_\theta = \frac{1}{E}(\sigma_\theta - \nu\sigma_\rho - \nu\sigma_z) \qquad [22b]$$

$$\varepsilon_z = 0 \qquad [22c]$$

$$u_\rho = \frac{R_i^2 p\rho}{E(R_e^2 - R_i^2)}\left[1 - \nu + \frac{R_e^2}{\rho^2}(1+\nu)\right] \qquad [23a]$$

$$u_\theta = u_z = 0 \qquad [23b]$$

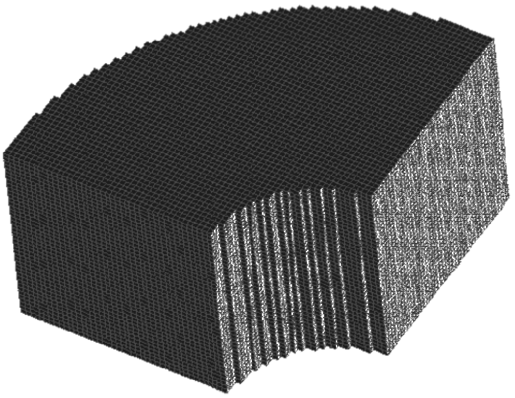


Figure 16. *Octree model for the cylinder under pressure problem*

Where R_i and R_e represent, respectively, the inner and outer radii of the cylinder and p the applied pressure. The problem is in fact two-dimensional and reproduces a state of plane strain. In our case, both ends have been fixed in the axial direction to take into account the plane strain assumption. Owing to the symmetry, only one quarter of the cylinder has been modelled, as shown in Figure 16.

The obtained accuracy for this problem is entirely similar to that of the standard NEM. The rate of convergence for this particular problem is shown in Figure 17, together with the obtained CPU times for standard and octree-based NEM models for this problem, in Figure 18.

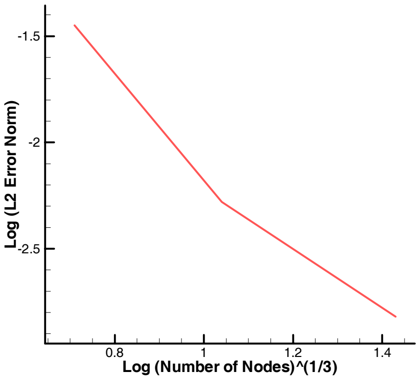


Figure 17. *L_2 error norm for the hollow cylinder problem*

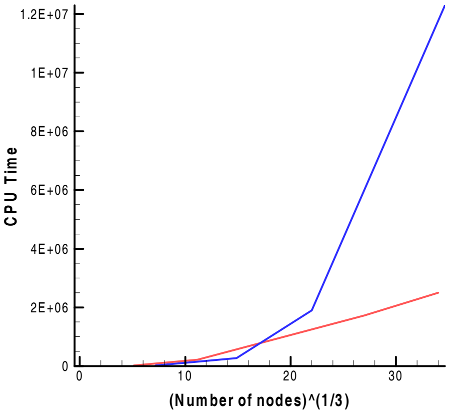


Figure 18. *CPU times obtained for the standard (dark line) and octree-based (light line) model for the cylinder problem*

5.3. *A biomechanical analysis of a pig's heart*

As an example of the application of the proposed method to real biomechanical problems we consider the analysis of the stress in a pig's heart under internal pressure. The geometry of the heart (see Figure 19) was provided by (Plaza, 2005) in the form of a CAD file. After processing, it was converted to a simplified STL format (Figure 20).

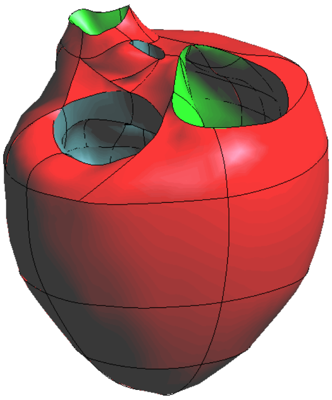


Figure 19. *View of the pig's heart*

The geometry was processed in order to generate an octree description of the heart, shown in Figure 21.

As boundary conditions we considered the heart clamped at auriculoventricular and semilunar valves. Ventricles are under pressure. In the left ventricle we applied 0.8 *kPa* and 0.16 *kPa* for the right one.

Although the actual structure of the heart tissue is fibrous and highly anisotropic, we considered, as a first approximation, a linear elastic, homogeneous and isotropic material. A deep discussion about these assumptions and a comparison with some more elaborated models can be found in (Plaza, 2005). In this case, Young's modulus is estimated to be around 15*Kpa*, being Poisson's ratio 0.3.

In Figure 22 a contour plot of the displacements of the heart walls is shown. The obtained displacements are slightly higer than the corresponding ones for a neo-hookean model, see (Plaza, 2005). In the before-mentioned work, maximum displacements of 2.25 mm were predicted for a linear elastic material, vs. 1.79 mm for a neo-hookean model. In our case, maximum displacements of 2.45 were obtained.

Despite the obvious limitations of the model, mainly from the material's point of view, results are in qualitatively good agreement with other, previous, more sophisticated models employing finite element techniques. Note that the proposed technique does not impose any limitations on the constitutive equations employed for the particular material being considered.

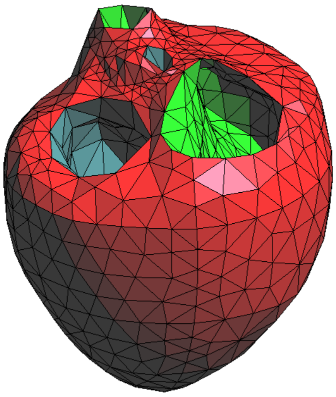


Figure 20. *STL description of the geometry*

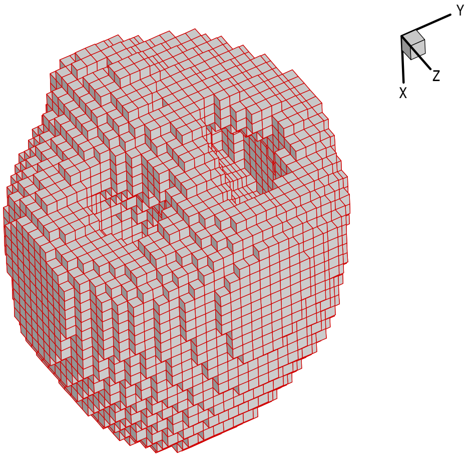


Figure 21. *Octree model of the heart*

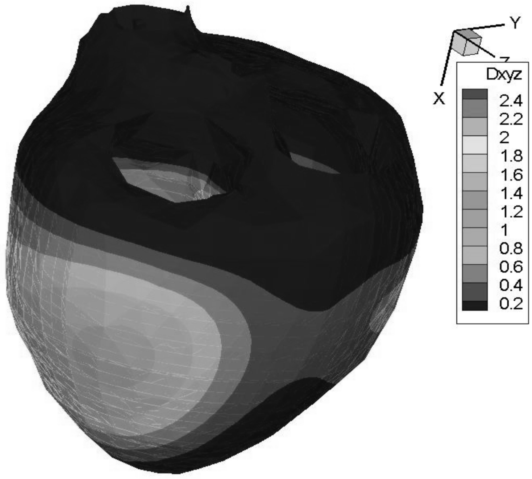


Figure 22. *Module of the displacements at the heart's walls*

6. Conclusions

In this paper we present a novel technique that is able to deal with extremely complex geometries arising from image data, either in voxel or STL formats. It is based on the employ of not necessarily uniform octree meshes that are conforming, in spite of the presence of hanging nodes. This is achieved through the use of natural neighbour interpolation, which renders the approximations continuous.

The method employs *R*-functions to enforce essential boundary conditions and principles arising from PU-methods to deal with holes and natural boundaries. It has been implemented so as to be able to store in memory many of the necessary values, thus having important savings in computer costs.

Accuracy of the method has demonstrated to be similar to that of the standard NEM, thus slightly better than that of the FEM.

Acknowledgements

The authors gratefully acknowledge the fruitful discussions with Dr. J. F. Rodriguez about the mechanical behaviour of the heart and his help, together with E. Heidenreich, in developing the geometry of the heart.

The work here presented has been partially supported by the Spanish Ministry of Education and Science through contract CICYT-DPI2005-08727-C02-01. This support is gratefully acknowledged.

7. References

Babuška I., Banerjee U., Osborn J.E., "Survey of meshless and generalized finite element methods: A unified approach", *Acta Numerica*, 2003, p. 1-125.

Belytschko T., Parimi Ch., Moës N., Sukumar N., Usui S., "Structured extended finite element methods for solids defined by implicit surfaces", *International Journal for Numerical Methods in Engineering*, 56, 2003, p. 609-635.

Cueto E., Sukumar N., Calvo B., Martinez M. A., Cegoñnino J., Doblare M., "Overview and recent advances in Natural Neighbour Galerkin methods", *Archives of Computational Methods in Engineering*, 10, (4), 2003, p. 307-384.

Klaas O., Shephard M. S., "Authomatic generation of octree based three-dimensional discretizations for partition of unity methods", *Computational Mechanics*, 25, 2000, p. 296-304.

Laguardia J. J., Cueto E., Doblaré M., "A natural neighbour Galerkin method with quadtree structure", *International Journal for Numerical Methods in Engineering*, 63, 2005, p. 789-812.

Lorensen W. E., Cline H. E., "Marching cubes: a high resolution 3d surface reconstruction algorithm", *Computer Graphics*, 21, 1987, p. 163-169.

Nagashima T., Ishihara Y., Niiyima K., Makinouchi A., "Development of Stress analysis method based on Voxel-type X-FEM", *Proceedings of the V World Congress on Compotatinal Mechanics, Viena*, Mang, Rammerstorfer and Eberhardsteiner, Eds. IACM, 2002.

Park S. W., Linsen L., Kreylos O., Owens J. D., Hamann B., "Discrete Sibson interpolation", *IEEE Transactions on Visualization and Computer Graphics*, to appear, 2005.

Plaza E., Simulation of a pig's heart by the finite element method, M.Sc. Thesis, University of Zaragoza, 2005, J. F. Rodriguez, advisor, in spanish.

Rvachev V. L., Sheiko T. I., "R-functions in boundary value problems in mechanics", *Applied Mechanics reviews*, 48, 1995, p. 151-188.

Samet H., "The quadtree and related hierarchical data structures", *Computing Surveys*, 16, (2), 1984, p. 187-260.

Shapiro V., Tsukanov I., "Meshfree simulation of deforming domains", *Computer-Aided Design*, 31, 1999, p. 459-471.

Sibson R., "A Vector Identity for the Dirichlet Tesselation", *Mathematical Proceedings of the Cambridge Philosophical Society*, 87, 1980, p. 151-155.

Sibson R., "A brief description of natural neighbour interpolation", *Interpreting Multivariate Data*, V. Barnett (Editor), John Wiley, 1981, p. 21-36.

Sukumar N., Moran B., Belytschko T., "The Natural Element Method in Solid Mechanics", *International Journal for Numerical Methods in Engineering*, 43, (5), 1998, p. 839-887.

van Rietbergen B., Weinans H., Huiskes R., "Computational strategies for iterative solutions of large fem applications employing voxel data", *International Journal for Numerical Methods in Engineering*, 39, 1996, p. 2473-2767.