The extended Delaunay tessellation

Nestor Calvo
International Center for Computational Methods in Engineering (CIMEC-INTEC), Universidad Nacional del Litoral and CONICET, Santa Fe, Argentina

Sergio R. Idelsohn
International Center for Computational Methods in Engineering (CIMEC-INTEC), Universidad Nacional del Litoral and CONICET, Santa Fe, Argentina
International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Barcelona, Spain

Eugenio Oñate
International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Barcelona, Spain

Received May 2002
Revised December 2002
Accepted January 2003

Keywords *Mesh generation, Delaunay/Voronoï tessellations*

Abstract *The extended Delaunay tessellation (EDT) is presented in this paper as the unique partition of a node set into polyhedral regions defined by nodes lying on the nearby Voronoï spheres. Until recently, all the FEM mesh generators were limited to the generation of tetrahedral or hexahedral elements (or triangular and quadrangular in 2D problems). The reason for this limitation was the lack of any acceptable shape function to be used in other kind of geometrical elements. Nowadays, there are several acceptable shape functions for a very large class of polyhedra. These new shape functions, together with the EDT, gives an optimal combination and a powerful tool to solve a large variety of physical problems by numerical methods. The domain partition into polyhedra presented here does not introduce any new node nor change any node position. This makes this process suitable for Lagrangian problems and meshless methods in which only the connectivity information is used and there is no need for any expensive smoothing process.*

1. Introduction

Several numerical methods in computational mechanics, as well as other volume integration methods need to subdivide the total domain into sub-domains called "elements". This is the case for the finite element method (FEM) and the finite volume method (FVM).

Once defined the total domain by the boundary surfaces, two standard algorithms may be used to perform the partition:

(1) divide the total domain into elements by an advancing front technique (AFT) (George, 1991; Lohner, 1996), or

Engineering Computations
Vol. 20 No. 5/6, 2003
pp. 583-600

0264-4401
DOI 10.1108/02644400310488763

(2) introduce a point distribution in the domain and then perform the partition via a Delaunay tessellation (DT) (George, 1991).

Both methods give, as a first step, unacceptable partitions to be used in a numerical method. For this reason, the first step needs an optimization of the element shape (sometimes called smoothing or "cosmetic" process), which is an iterative problem. For this reason, both methods are unbounded in computing time: the number of operations to obtain an acceptable partition to be used in a numerical solution may not be written as a function of the number of elements or nodes.

Both methods have advantages and disadvantages, which may be summarized as follows.

- In the AFT, the boundary surfaces are easily described, but the method is very expensive compared with the DT. For this reason, it is unacceptable to be used in problems with permanent partition update.
- The big advantage of the DT is that it is inexpensive and bounded in computing time. Unfortunately, it needs a lot of "cosmetic" in order to be used in a numerical method to solve a 3D physical problem. For this reason, the computing time may become as large as in the AFT.

Why the problem of a domain partition is until now an open question? The main reason is that in several physical problems, it is necessary to perform a domain partition at each time step. This is for instance the case for transient problems with moving boundaries or in Lagrangian formulations in fluid mechanics. In these cases the use of a bounded method is mandatory, the number of operations to achieve an acceptable partition must be fixed as a function of the number of elements or nodes. Both the methods, AFT and DT, need a process of cosmetics which increases without a fixed limit in the computing time.

In this paper, a method based on the DT will be described. The problem will be: given a node distribution, find an optimal partition, regarding computing time and element shape, to be successfully used in a numerical method.

It must be noted that obtaining a node distribution in a domain is not a difficult task and it is a bounded operation. The goal of this paper will be to obtain from an arbitrary node distribution a partition of the total domain. This partition will be optimal in order to be used in a numerical method as the FEM and the number of operations to obtain this partition must be bounded with n^{β}, where n is the total number of nodes and β must be much less than 2. Furthermore, the partition must be unique for a given node distribution.

2. The DT

In order to better understand the new procedure, classical definitions will be introduced for three entities: Voronoï diagrams, DT and Voronoï spheres.

Let a set of distinct nodes be: $\mathbf{N} = \{n_1,\ n_2,\ n_3, \ldots, n_n\}$ in $\mathbb{R}^3$.

(a) The *Voronoï diagram* of the set **N** is a partition of $\mathbb{R}^3$ into regions V_i (closed and convex, or unbounded), where each region V_i is associated with a node n_i, such that any point in V_i is closer to n_i (nearest neighbor) than to any other node n_k. See Figure 1 for a 2D representation. There is a single Voronoï diagram for each set **N**.

(b) A *Voronoï sphere* within the set **N** is any sphere, defined by four or more nodes without any node inside. Such spheres are also known as empty circumspheres.

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

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

The DT of a set of nodes is non-unique. For the same node distribution, different tetrahedrations are possible. Therefore, a partition based on the DT is sensitive to geometric perturbations of the node positions. On the other hand, its dual, the Voronoï diagram, is unique. Thus, it makes more sense to define a partition based on the unique Voronoï diagram than on DT. In Figure 2, two critical cases of Delaunay instabilities are represented. One is the case of four nodes on the same circle and the other is the case of a node close to a boundary. In both the cases, the Voronoï diagram remains almost unchanged.

Furthermore, in 3D problems the DT may generate several tetrahedra of zero or almost zero volume, which introduces large inaccuracies into the shape function derivatives to be used in a numerical method. This is the reason why a DT must be improved iteratively in order to obtain an acceptable partition to be used in a numerical method. The time to obtain an acceptable partition via a DT is then an unbounded operation, thus not satisfying the requirement expressed in the introduction.

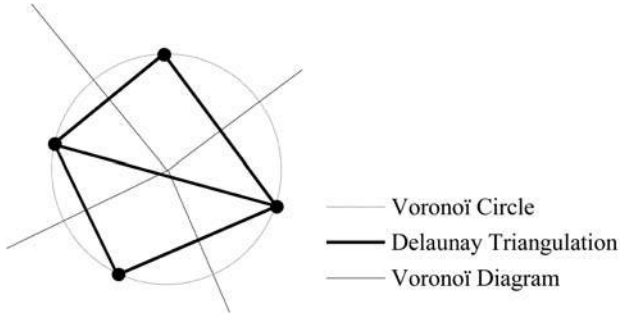

Figure 1.
Voronoï diagram, Voronoï circle and Delaunay triangulation for a four node distribution in 2D

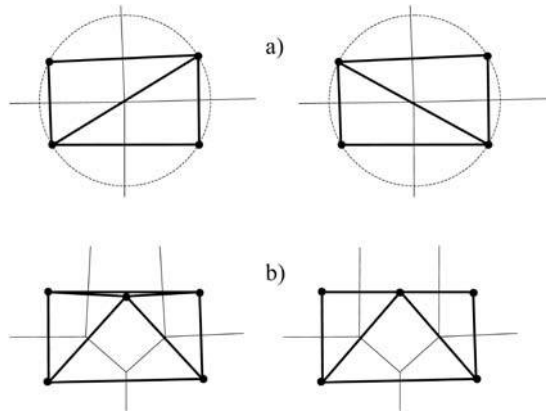

Figure 2.
Instabilities on the DT. (a) Four nodes on the same circle; (b) node close to a boundary

3. The extended DT

From the point of view of the application of a domain partition to the solution of a numerical method, the best partition is which the elements have:

(1) all its nodes on the same empty sphere (this is the concept of optimal distance between the nodes), and

(2) the polyhedral elements fill the sphere as much as possible (this is the concept of optimal angle between faces).

The standard DT satisfies both previous statements only for 2D problems. Unfortunately, the second statement is not necessarily satisfied in a 3D partition.

The drawbacks appear in the so-called "degenerated case", which is the case where more than four nodes (or more than three nodes in a 2D problem) are on the same empty sphere. For instance, in 2D, when four nodes are on the same circumference, two different triangulations satisfy the Delaunay criterion. However, the most dangerous case appears only in 3D. For instance, when five nodes are on the same sphere, five tetrahedra may be defined satisfying the Delaunay criterion, but some of them may have zero or almost zero volumes, called slivers (see Figure 3).

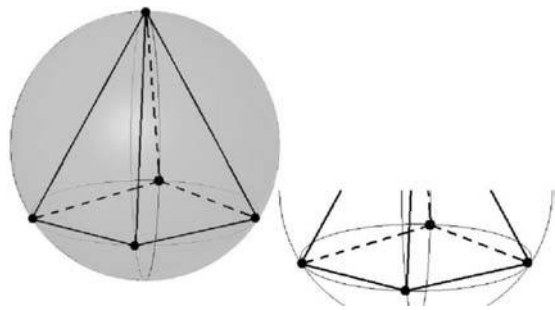

Figure 3.
Five nodes on the same sphere and possible zero or almost zero volume tetrahedron (sliver) on the right

In order to overcome the drawbacks referred, a generalization of the DT will be defined:

Definition. The extended DT (EDT) within the set $\mathbf{N}$ is the unique partition of the convex hull Ω of all the nodes into regions Ω_i such that $\Omega = \cup \Omega_i$, where each Ω_i is the polyhedron defined by all the nodes laying on the same Voronoï sphere.The main difference between the traditional DT and the EDT is that, in the latter, all the nodes belonging to the same Voronoï sphere define a unique polyhedron. With this definition, the domain Ω will be divided into tetrahedra and other polyhedra, which are unique for a given node distribution, satisfying one of the goals required in the Introduction.

Figure 4, for instance, is a 2D polygon partition with a triangle, a quadrangle and a pentagon. Figure 5 is a polyhedron with all the nodes on the same sphere, which may appear in a 3D problem.

In order to avoid numerical problems, which may hide polyhedra with more than four nodes, the polyhedra are defined by all the nodes of the same sphere and nearby spheres. The proximity of the spheres is governed by a parameter δ (see Appendix 1).

The parameter δ avoids the possibility of having near zero volume tetrahedra. When δ is large, the number of polyhedra with more than four nodes will increase, and the number of tetrahedra with near zero volume will decrease, and vice versa.

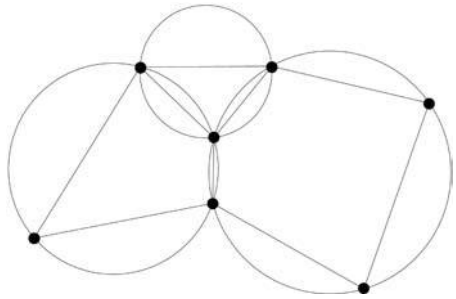

Figure 4.
Two-dimensional partition in polygons. The triangle, the quadrangle and the pentagon, each are inscribed on a circle

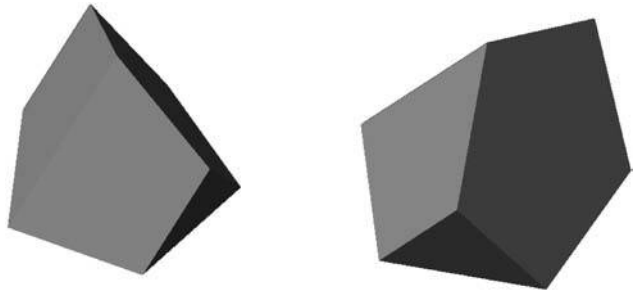

Figure 5.
Eight-node polyhedron. All nodes are on the same sphere

Then, the idea of the EDT is to avoid the tetrahedral partition when more than four nodes are in the same empty sphere (or near the same). When this is the case, the best element to be used in a numerical method is the polyhedron formed with all these nodes.

The EDT allows the existence of a domain partition which:

(1) is unique for a node distribution,

(2) is formed without polyhedra with near zero volume, and

(3) is obtained in a bounded time of order n.

Then, it satisfies all the goals stated earlier.

It must be noted that until quite recently, all the mesh generators were limited to the generation of tetrahedral or hexahedral elements (or triangular and quadrangular in 2D problems). The reason for this limitation was the lack of any acceptable shape function to be used in other kind of geometrical elements. Nowadays, several acceptable shape functions for any kind of polyhedron are easily generated (Belikov and Semenov, 1998; Idelsohn *et al.*, 2002 or Appendix 2). These new shape functions, together with the EDT presented in this paper, gives an optimal combination and a powerful tool to solve a large variety of physical problems by numerical methods.

4. The boundary surface

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

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

All nodes defining an empty circumsphere with a radius $r(\mathbf{x})$ *larger than* α $h(\mathbf{x})$, *are considered as boundary nodes*.

In this criterion, α is a parameter close to, but greater than one. Note that this criterion is coincident with the Alpha Shape concept (Akkiraju *et al.*, 1995; Edelsbrunner and Mucke, 1994).

Once a decision has been made concerning which of the nodes are on the boundaries, the boundary surface must be defined. It is well known that, in 3D problems, the surface fitting a number of nodes is not unique. For instance, four boundary nodes on the same sphere may define two different boundary surfaces, one concave and the other convex.

In order to avoid this undefined boundary, the boundary surfaces will be defined with:

All the polyhedral surfaces having all their nodes on the boundary and belonging to just one polyhedron are boundary surfaces.

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

Nevertheless, it must be noted that in the criterion proposed earlier, the error in the boundary surface definition is bounded and proportional to h. This is the standard error of the boundary surface definition in a numerical method for a given node distribution.

5. Element quality indicator

In order to evaluate the partition obtained with the EDT, compared with other partition as, for instance, DT, a polyhedron quality indicator will be defined.

A wrong mesh is a mesh which is not acceptable to be used in a numerical method. The main drawback of a Delaunay mesh is the presence of zero or almost zero volume polyhedra. In fact, the problem of almost zero volume is the existence of shape functions with infinite or almost infinite gradients. These infinite gradients deteriorate the numerical solution.

Classical definition of mesh quality will not be used here. The reason is that in this paper a node distribution is considered for which an optimal partition must be found without moving or removing nodes. The presence of nodes very close to each other in the node distribution is considered as a necessity of the physical problem to represent correctly a gradient in the solution and not as a wrong partition.

For instance, tetrahedra currently called *spires* and *wedges* (see Figure 6) will be considered as elements of good quality because they are the best elements for a node distribution with the nearest nodes along one direction and more separated nodes in the other. These elements represent correctly the gradient "expected" in each direction to solve a particular physical problem. On the other hand, *slivers* or *splinters* as well as *caps*, will be considered as wrong elements because they introduce shape functions with infinite or almost infinite gradients, which are not in agreement with the gradients expected for this node distribution.

In order to obtain a parameter to define which is a good or a wrong element to be used in a numerical method, but considering that the node distribution is fixed and introduced for physical needs, the gradient ratio will be defined as:

$$\gamma = \frac{\text{maximum gradient of the shape functions}}{\text{maximum gradient expected}} \tag{1}$$

The maximum gradient expected for the node distribution will be defined in each polyhedron as $1/h_{\min}$, where $h_{\min}$ is the shortest distance between the two nodes belonging to the polyhedron.

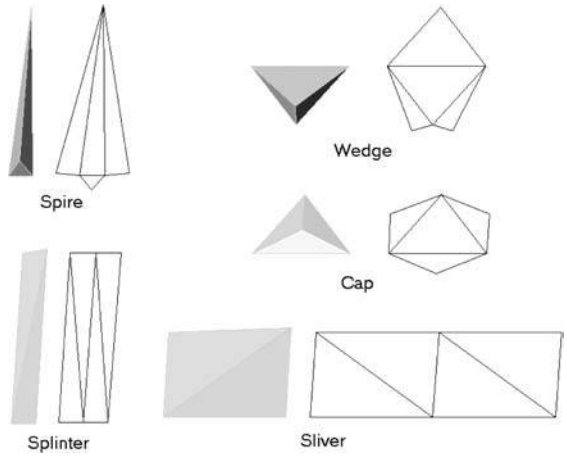

Figure 6.
Perspective unfolding and naming of some bad-shaped tetrahedra

The maximum gradient of the shape functions may be evaluated directly using the largest gradient modulus from the shape functions of the polyhedron $|\nabla N_p|_{\max}$ (see Appendix 2 for the definition of the shape functions N_p).

The quality of a polyhedron is then numerically defined as:

$$\gamma = \frac{1}{h_{\min}|\nabla N_p|_{\max}} \tag{2}$$

It must be noted that in the EDT algorithm there is no smoothing process in which the element qualities must be evaluated many times. The gradient ratio γ is only defined here in order to verify the quality of the resulting mesh.

The ideal element will therefore have a large gradient ratio. This will have both: the required distribution on the Voronoï sphere and an acceptable shape function for numerical computations. On the other hand, bad elements have all their nodes on the same empty sphere, but the small γ value is indicating high shape function gradients capable to destroy the numerical solution.

Taking into account that the computer precision nowadays is of order 10^{-16}, and also the results from the numerical test performed in the next section, a gradient ratio $\gamma > 10^{-2}$ is recommended in order to accept a mesh for numerical computations.

6. Numerical test

In order to check if a domain partition is acceptable or not to be used in a numerical method, a physical problem must be solved using a defined partition.

A cube of unit side, with an internal exponential source, has been used to validate the EDT.

The problem to be solved is the classical Poisson equation:

$$\nabla^2 u = f(x, y, z) \tag{3}$$

With the internal source:

$$f(x,y,z) = (-2kyz(1-y)(1-z) + (kyz(1-y)(1-z)(1-2x)^2)^2 - 2kzx(1-z)(1-x) + (kzx(1-z)(1-x)(1-2y)^2)^2 - 2kxy(1-x)(1-y) + (kxy(1-x)(1-y)(1-2z)^2)^2) \times (-e^{kxyz(1-x)(1-y)(1-z)}/(1-e^{k/64})) \quad (4)$$

The boundary condition is the unknown function set u, equal to zero on all the boundaries.

This problem has the following analytical solution:

$$u(x,y,z) = (1 - e^{kxyz(1-x)(1-y)(1-z)})/(1-e^{k/64}) \quad (5)$$

Several node distributions have been tested with 125 (5^3), 729 (9^3), 4,913 (17^3), and 35,937 (33^3) nodes, with structured and non-structured node distributions. In all cases, the numerical solution was obtained using the linear finite element shape functions for tetrahedral elements and polyhedral shape functions (as defined in the Appendix 2) for polyhedral elements.

For the structured node distributions, the following procedure was used to generate the nodes. Initially, all the nodes are in a regular cubic array with a constant distance h between the neighbor nodes. Then, each internal node has been randomly displaced at a distance ρh (with $\rho \ll 1$) in order to have an arbitrary, but structured, node distribution. Surface and edge nodes are perturbed, but remaining in the surface or the edge. Corner nodes were not perturbed. In this paper, the parameter ρ was fixed to 10^{-6}.

The 3D non-structured node distribution was generated using the GID pre/post-processing code (GID, 2002) with a constant h distribution. GID generates the nodes using an AFT, which guarantees that the minimal distance between the two nearby nodes lies between $0.707h$ and $1.414h$.

6.1 EDT versus DT

It must be noted that in the 2D problems, both node distributions generated as described before, structured and non-structured, will give a Delaunay partition with near-constant area triangles, which is optimal for a numerical solution. Nevertheless, this is not the case in the 3D problems in which, even for a constant h node distribution, many zero or near-zero volume tetrahedra (slivers) will be obtained on a standard Delaunay partition (Edelsbrunner and Damrong, 2001). Figure 7 shows, for instance, the presence of slivers on a structured eight-node distribution. Slivers may introduce large numerical errors in the solution of the unknown functions and their derivatives, which may completely destroy the solution.

In order to show this behavior and to show that the EDT eliminates this problem, the following tests were performed: for a fixed-node partition (e.g. 17^3 nodes) the δ parameter (introduced in Section 3 and in Appendix 1) was swept

from 0 to 10^{-1}. With $\delta = 0$ (i.e. Voronoï spheres are never joined) the standard DT is obtained. Larger values of δ give the EDT.

Figure 8 shows the error in L_2 norm for the derivative of the solution of the 3D problem stated in equations (1) and (2). This has been done both for structured and non-structured distributions against the δ parameter. It can be shown that in both cases the errors are very large ($\sim 10^{1}$) for $\delta > 10^{-6}$ and very small ($\sim 10^{-2}$) for $\delta > 10^{-5}$. Larger δ do not change the results. Hereafter, the EDT will be considered for $\delta > 10^{-4}$.

This example is very important because it is showing that, for a given node distribution, a tetrahedration using the standard Delaunay concept do not work. Mesh generators currently use edge-face swapping or another cosmetic algorithms to overcome the presence of wrong elements. All those operations are unbounded in computing time. The idea of joining similar spheres, even for a very small δ parameter, solves this problem in a very simple way. The wrong tetrahedra are automatically joined to form polyhedra with optimal shapes.

The results of Figure 8 also show that δ must be large compared with the computer precision (e.g. $\sim 10^{-5}$ for a computer precision of order 10^{-16}) but also means that δ is not a parameter to be adjusted in each example as results do not change by setting δ larger than 10^{-4}. In fact, all our examples were carried out with a fixed $\delta = 10^{-2}$.

All the polyhedra obtained with that fixed value of δ had qualities $\gamma > 10^{-2}$, thus acceptable gradients and low error.

6.2 Good and wrong elements

In order to show the performance of the EDT to generate good elements, the gradient ratio of the node distribution of the previous example was evaluated.

In Figure 9, for a fixed δ, the heights of the columns represent the number of elements having the same gradient ratio.

The importance of Figure 9 is to show that, by joining similar spheres, the best polyhedra are automatically built. For the standard DT (and also for small δ values), there are bad tetrahedra for both the structured and the

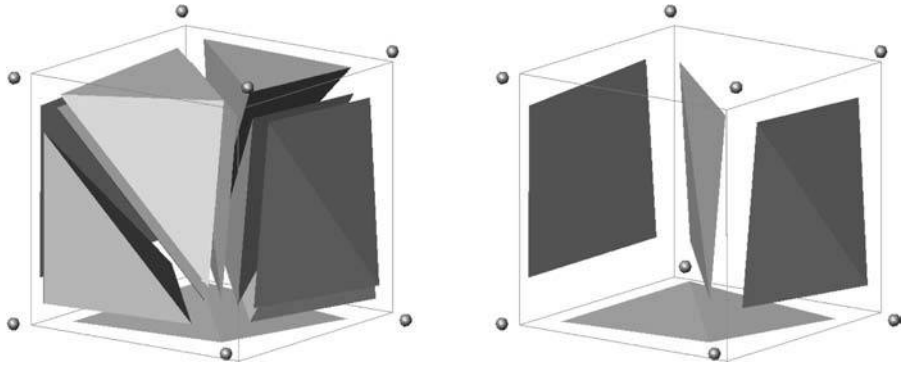

Figure 7.
Presence of slivers in a Delaunay partition of a perturbed cube. Left: tetrahedra produced by the Delaunay partition. Right: slivers isolated

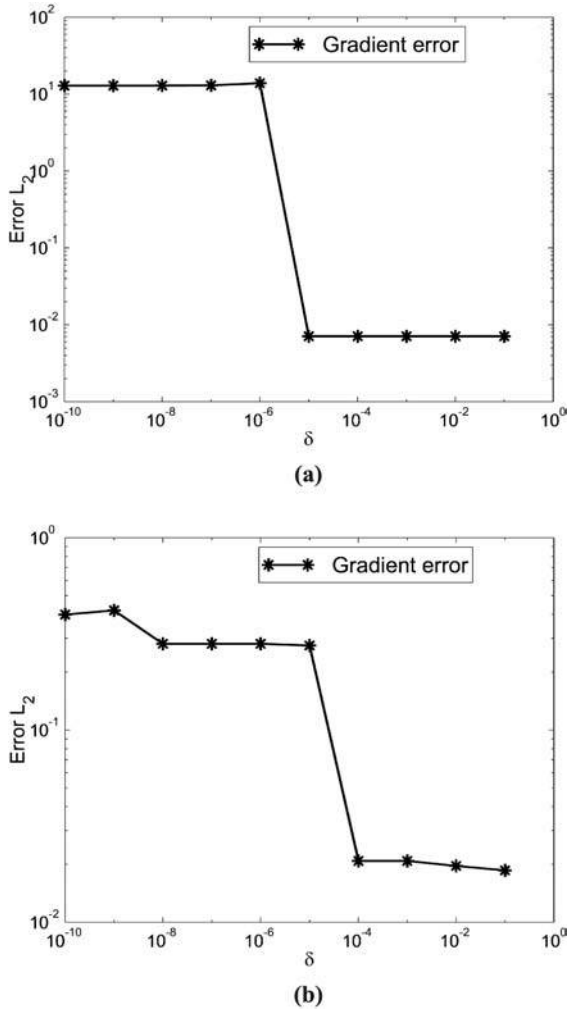

Figure 8.
Cube with exponential source. Error of the derivative in L^2. (a) Structured node distribution; (b) non-structured node distribution

non-structured node distribution. Increasing δ, the slivers disappear and, for the structured case, all the elements become "automatically" hexahedra (cubes), which is the optimal tessellation for this node distribution. For the non-structured case, EDT insure a mesh without any sliver when δ is set to any value larger than 10^{-6}.

6.3 Convergence rate

Figure 10 shows the convergence of the above-defined example, when the number of nodes is increased from 5^3 to 33^3. The upper plots show the error in L^2-norm, both for the function and its derivatives. All the graphics shows

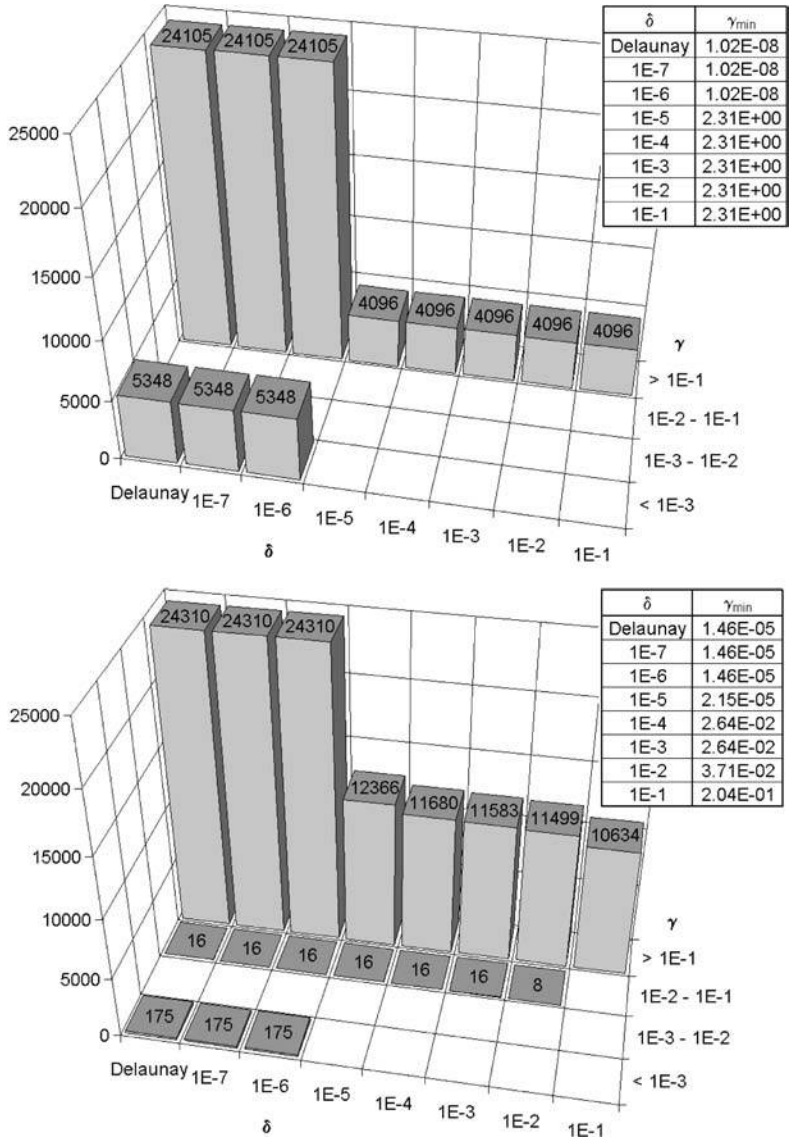

Figure 9.
Number of polyhedra by volume ratio for different δ. Top: structured node distribution. Bottom: non-structured distribution made by GID

an excellent convergence rate. It must be noted that for all the non-structured node distributions tested (and also the structured ones for $\rho = 10^{-6}$), the FEM with elements generated using a DT gave totally wrong results, and even at times ill-conditioned matrices were frequently found during the stiffness matrix evaluation.

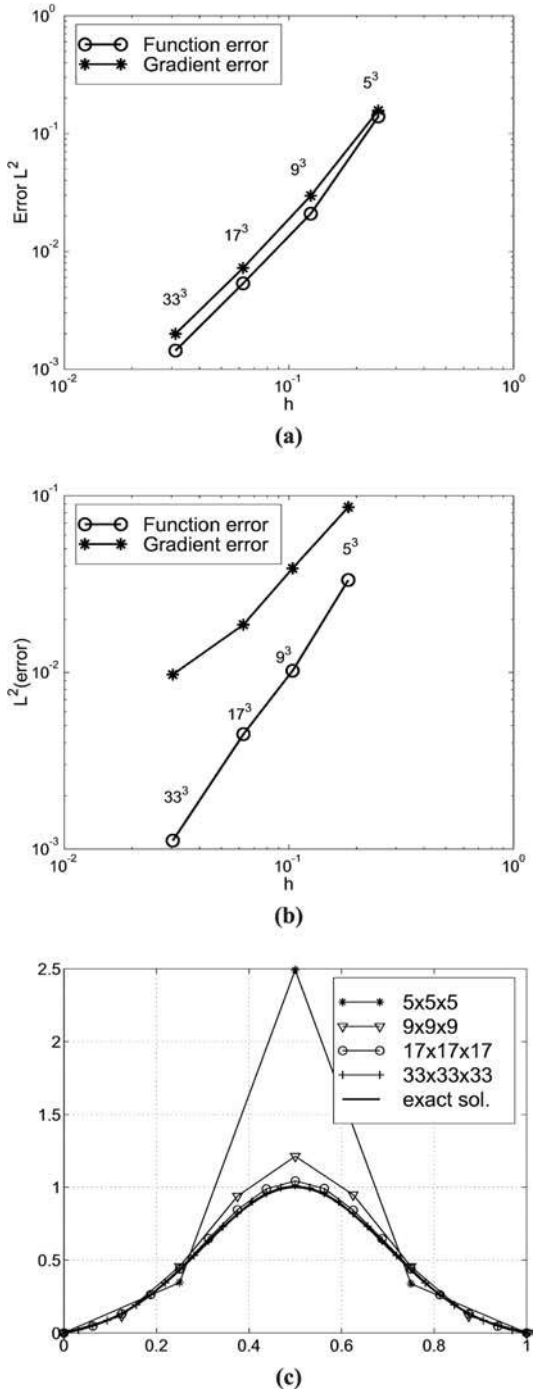

Figure 10.
Cube with exponential source. Convergence of the numerical solution and its gradient for different partitions. (a) Structured node distribution; (b) non-structured node distribution; (c) center-line solutions obtained with structured node distribution (the same results were obtained using the non-structured node distribution)

6.4 Computing times

In order to validate that the number of operations in the evaluation of the EDT partition is of order n, the computing time in a standard PC (Intel PIII 800 MHz) was analyzed. Figure 11 shows the time in seconds for both the structured and the non-structured cases.

A regression of the obtained results shows that the computing time is approximately:

$$t(\mathrm{s}) = 0.000325\, n^{1.08} \quad \text{for structured meshes, and}$$

$$t(\mathrm{s}) = 0.000283\, n^{1.10} \quad \text{for non-structured,}$$

showing that the convergence exponent is even better than the 1.333 bound expected in a DT.

6.5 Three-dimensional arbitrary geometry mesh

In order to show the performance of the method with an arbitrary 3D geometry, a volume representing the vocal A was generated with distributions of $n \cong 10^3$ and 10^5 nodes, using the GID node generator with a constant h. Figure 12 (left) shows, for instance, the boundary node distribution for $n = 8{,}452$ nodes.

Using the EDT algorithm described in Section 3 with the boundary surface definition described in Section 4 and an alpha shape parameter $\alpha = 1.3$, polyhedral partitions were found with external boundary surfaces represented in Figure 12.

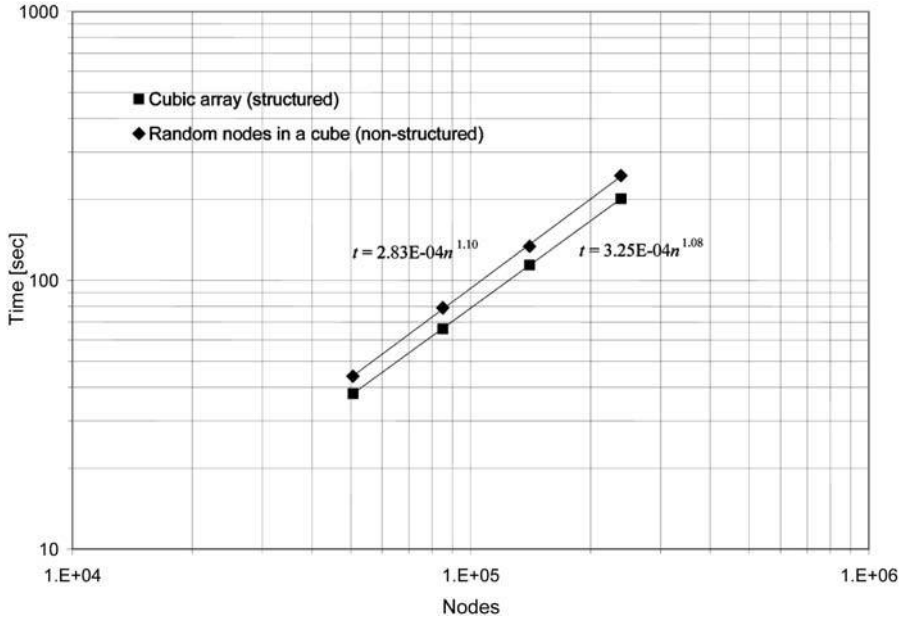

Figure 11.
Time versus number of nodes in a standard PC

The following are the main characteristics polyhedral mesh made for the larger node set, compared with the tetrahedral mesh obtained via the standard DT (Table I).

From a total of 106,947 nodes, the standard DT generates 599,934 tetrahedra, but 349 of them are slivers. On the other hand, EDT generates 430,262 tetrahedra and 67,162 polyhedra with more than four nodes. None of them have gradient ratio smaller than 10^{-2}. Another interesting conclusion to take out from this example is that in the EDT mesh, 86.5 percent of the elements are tetrahedral and then, in these elements the definition of the shape functions (Appendix II) will be coincident with classical linear shape functions of the FEMs.

Finally, Figure 13 shows the slivers distribution in the mesh. Elements having a γ parameter smaller than 10^{-2} have been plotted in black. Zero bad polyhedra have been found in the partition given by the EDT.

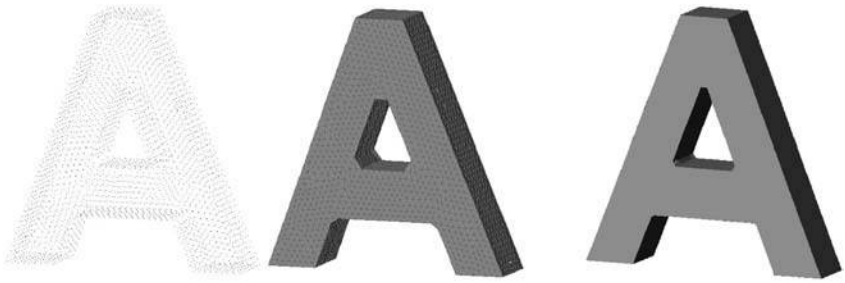

Figure 12.
Three-dimensional arbitrary geometry. Left: boundary nodes from the point distribution generated by GID with constant h and 8,452 nodes. Center: boundary surface for 8,452 nodes. Right: boundary surface for 106,947 nodes

	EDT	DT
Nodes	106,947	106,947
Polyhedra	497,424	599,934
Tetrahedra	430,262	599,934
Slivers	0	349
$\gamma_{\min}$	1.84e-2	2.34e-16

Table I.

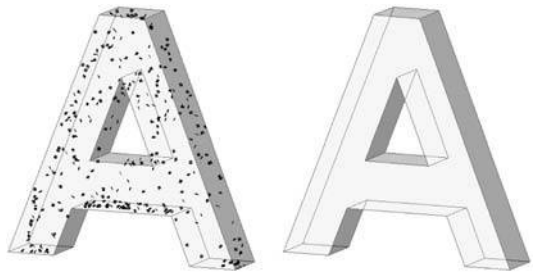

Figure 13.
Wrong elements in the 106,947 nodes tessellation. Left: 349 bad tetrahedral (slivers) in the DT. Right: zero wrong polyhedra in the EDT

7. Conclusions

For a given set of nodes with arbitrary 3D distribution and non-structured and variable distance between nodes, the method proposed gives a polyhedral mesh and a boundary surface with the following characteristics.

(1) The solution is unique for a given set of parameters δ and α. The solution is not sensitive to small variation of those parameters.

(2) Each polyhedron has all its nodes on the same empty sphere (optimal distance between nodes).

(3) Each polyhedron has an acceptable positive volume to be used in a numerical method. (The maximum gradient of the shape functions is of the same order of the expected gradient for a given node distribution.)

(4) The boundary surface obtained may be concaves or convexes, and the correct definition depends on the local node distance between nodes $h(x)$.

(5) The computing time to achieve the mesh is bounded and of order n.

(6) The large majority of the polyhedral elements generated are tetrahedra (around 85 percent in an arbitrary node distribution). This allows the use of standard lineal finite element shape functions in all of them.

References

Akkiraju, N., Edelsbrunner, H., Facello, M., Fu, P., Mücke, E. and Varela, C. (1995), "Alpha shapes: definition and software", *Proceedings of the 1st International Computational Geometry Software Workshop*, pp. 63-6, url: http://www.geom.umn.edu/software/cglist/GeomDir/shapes95def/

Belikov, V. and Semenov, A. "Non-Sibsonian interpolation on arbitrary system of points in Euclidean space and adaptive generating isolines algorithm", *Numerical Grid Generation in Computational Field Simulation, Proc. of the 6th Intl Conf.*, July 1998, Greenwich Univ.

Edelsbrunner, H. and Damrong, G. (2001), "An experimental study of sliver exudation", *Proceedings 10th International Meshing Roundtable*, 7-10 October, Sandia National Laboratories, pp. 307-16.

Edelsbrunner, H. and Mucke, E.P. (1994), "Three-dimensional alpha shapes", *ACM Transactions on Graphics*, Vol. 13, pp. 43-72.

George, P.L. (1991), *Automatic Mesh Generation, Applications to Finite Methods*, ISBN 0-471-93097-0, Wiley, New York.

GID (2002), "The personal pre and post processor", CIMNE, Barcelona, url: http://gid.cimne.upc.es

Idelsohn, S., Oñate, E., Calvo, N. and Del Pin, F. (2002), "The meshless finite element method", *International Journal for Numerical Methods in Engineering* (in press).

Lohner, R. (1996), "Progress in grid generation via the advancing front technique", *Engineering with Computers*, Vol. 12, pp. 186-210.

Appendix 1. Criterion to join polyhedra

Consider two Voronoï spheres having nearby centers. See Figure A1 for a two-dimensional reference.

As both Voronoï spheres are empty, they must satisfy the following relationship:

$$|r_2 - r_1| \leq \|c_1 - c_2\| \tag{6}$$

where r are the radii and c the centers of the spheres.

Thus, two spheres are similar when their centers satisfy:

$$\|c_1 - c_2\| < \delta r_{\text{rms}}, \tag{7}$$

where δ is a small non-dimensional value and r_{rms} is the root mean square radius.

Two polyhedra will be joined if they belong to similar spheres.

The algorithm finds all the four-node empty spheres, and then the polyhedra are successively joined using the above criterion. It must be noted that when all the nodes of a polyhedron belongs to another polyhedron, only the last one is considered.

Appendix 2. Shape functions for arbitrary 3D polyhedra

For any point within a polyhedron **P**, there is a Voronoï cell **V**(x) associated with the variable point x in the Voronoï tessellation of the set $\mathbf{P} \cup \{x\}$.

Figure A2 shows that every node $n_p \in \mathbf{P}$ has a corresponding face $\mathbf{F}_p$ of **V**, which is normal to the segment $\{x, n_p\}$ by its midpoint. This is because **V** is the set of points closer to x than any other point.

Defining the functions:

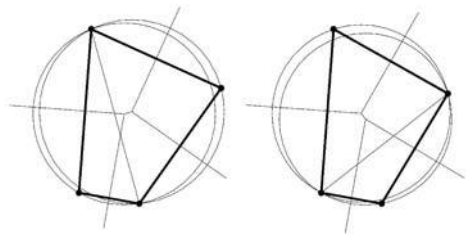

Figure A1.
Four nodes in near-degenerate position showing the empty circumcircles, the Voronoï diagram and the corresponding discontinuous Delaunay triangulation

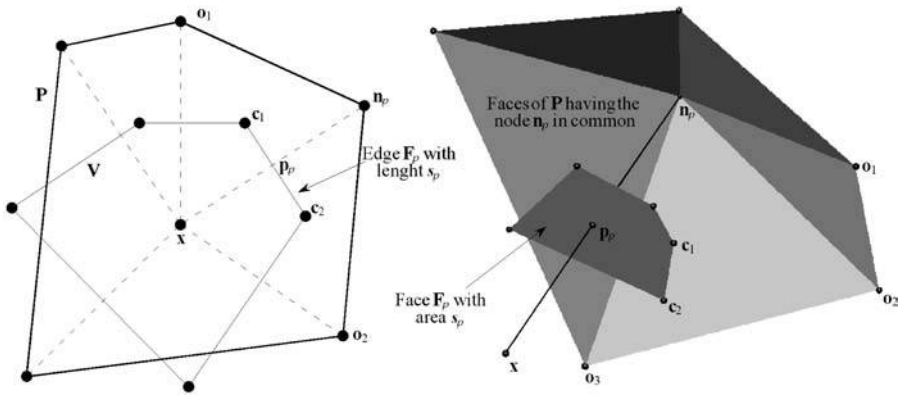

Figure A2.
Elements defining 2D and 3D shape functions

$$\phi_p(x) = s_p / \|n_p - x\| = s_p / h_p, \tag{8}$$

as the quotient of the Lebesgue measure s_p of $\mathbf{F}_p$ and the distance h_p between the point x and the node n_p. The shape functions are:

$$N_p = \phi_p \Big/ \sum_q \phi_q, \tag{9}$$

These functions automatically satisfy the partition of unity property:

$$\sum_p N_p = 1. \tag{10}$$

This article has been cited by:

1. Amirtham Rajagopal, Markus Kraus, Paul Steinmann. 2018. Hyperelastic analysis based on a polygonal finite element method. *Mechanics of Advanced Materials and Structures* **25**:11, 930-942. [Crossref]
2. Abdiel Ramon Leon Bal, Ulrich Hoppe, Thai Son Dang, Klaus Hackl, Günther Meschke. 2018. Hypoplastic particle finite element model for cutting tool-soil interaction simulations: Numerical analysis and experimental validation. *Underground Space* **3**:1, 61-71. [Crossref]
3. S. de Azevedo, M. F. González, C. Cintas, V. Ramallo, M. Quinto-Sánchez, F. Márquez, T. Hünemeier, C. Paschetta, A. Ruderman, P. Navarro, B. A. Pazos, C. C. Silva de Cerqueira, O. Velan, F. Ramírez-Rozzi, N. Calvo, H. G. Castro, R. R. Paz, R. González-José. 2017. Nasal airflow simulations suggest convergent adaptation in Neanderthals and modern humans. *Proceedings of the National Academy of Sciences* **114**:47, 12442-12447. [Crossref]
4. P.B. Ryzhakov, J. Marti, S.R. Idelsohn, E. Oñate. 2017. Fast fluid–structure interaction simulations using a displacement-based finite element model equipped with an explicit streamline integration prediction. *Computer Methods in Applied Mechanics and Engineering* **315**, 1080-1097. [Crossref]
5. M Mejia-Ugalde, A Dominguez-Gonzalez, M Trejo-Hernandez, LA Morales-Hernandez, RA Osornio-Rios, JP Benitez-Rangel. 2016. Triangulation intersection approach from Poisson's equation applied to automatic tool selection in computer numerical control mill-lathe. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **230**:4, 722-731. [Crossref]
6. P. Ryzhakov, R. Rossi, A. Viña, E. Oñate. 2013. Modelling and simulation of the sea-landing of aerial vehicles using the Particle Finite Element Method. *Ocean Engineering* **66**, 92-100. [Crossref]
7. Markus Kraus, Amirtham Rajagopal, Paul Steinmann. 2013. Investigations on the polygonal finite element method: Constrained adaptive Delaunay tessellation and conformal interpolants. *Computers & Structures* **120**, 33-46. [Crossref]
8. Juan Carlos Cante, M. Dolores Riera, Juan Oliver, Jose Manuel Prado, Alvaro Isturiz, C. Gonzalez. 2011. Flow regime analyses during the filling stage in powder metallurgy processes: experimental study and numerical modelling. *Granular Matter* **13**:1, 79-92. [Crossref]
9. Josep Maria Carbonell, Eugenio Oñate, Benjamín Suárez. 2010. Modeling of Ground Excavation with the Particle Finite-Element Method. *Journal of Engineering Mechanics* **136**:4, 455-463. [Crossref]
10. Gerardo V. Mazzetta, Adrián P. Cisilino, R. Ernesto Blanco, Néstor Calvo. 2009. Cranial mechanics and functional interpretation of the horned carnivorous dinosaur Carnotaurus sastrei. *Journal of Vertebrate Paleontology* **29**:3, 822-830. [Crossref]
11. Sergio Ortiz, Damian Siedlecki, Laura Remon, Susana Marcos. 2009. Three-dimensional ray tracing on Delaunay-based reconstructed surfaces. *Applied Optics* **48**:20, 3886. [Crossref]
12. Alexandru Constantiniu, Paul Steinmann, Tom Bobach, Gerald Farin, Georg Umlauf. 2008. The Adaptive Delaunay Tessellation: a neighborhood covering meshing technique. *Computational Mechanics* **42**:5, 655-669. [Crossref]
13. Dimitris Vartziotis, Theodoros Athanasiadis, Iraklis Goudas, Joachim Wipper. 2008. Mesh smoothing using the Geometric Element Transformation Method. *Computer Methods in Applied Mechanics and Engineering* **197**:45-48, 3760-3767. [Crossref]
14. Bo Tang, Junfeng Li, Tianshu Wang. 2008. The least square particle finite element method for simulating large amplitude sloshing flows. *Acta Mechanica Sinica* **24**:3, 317-323. [Crossref]

15. E. Oñate, J. Rojek, M. Chiumenti, S.R. Idelsohn, F. Del Pin, R. Aubry. 2006. Advances in stabilized finite element and particle methods for bulk forming processes. *Computer Methods in Applied Mechanics and Engineering* **195**:48-49, 6750-6777. [Crossref]
16. E. Oñate, J. García, S.R. Idelsohn, F. Del Pin. 2006. Finite calculus formulations for finite element analysis of incompressible flows. Eulerian, ALE and Lagrangian approaches. *Computer Methods in Applied Mechanics and Engineering* **195**:23-24, 3001-3037. [Crossref]