

FEA-driven Geometric Modelling for Meshless Methods

Prof Dr. Eng. Oscar Ruiz, Student Assistant Miguel Granados, Prof Dr. Math. Carlos Cadavid

CAD CAM CAE Laboratory
EAFIT University, Medellín, COLOMBIA
Phone : +57-4-261-9500 / Fax : +57-4-266-4284
E-mail : {oruiz,mgranad1,ccadavid}@eafit.edu.co

Abstract: Optimized Boolean Operations against orthogonal Fixed Grids (**FG**) for 2-manifold construction in quasi-meshless methods for Finite Element Analysis are presented. A Piecewise Linear (PL) or Boundary Representation (B-Rep) **B** is assumed to be the boundary of a solid $S \subset \mathbf{R}^3$. On the other hand, \mathbf{R}^3 is partitioned into a 3-dimensional array of cubic, uniform cells $C_{i,j,k}$. Cells $C_{i,j,k}$ with $C_{i,j,k} \cap S \neq \Phi$ and $C_{i,j,k} \cap S \neq C_{i,j,k}$ are particularly important for FG applications. These are the cells $C_{i,j,k}$ intersecting **B**, which happen to be Neither Inside nor Outside (NIO) of **B**. The boundary $\partial(C_{i,j,k} \cap S)$ of $C_{i,j,k} \cap S$ must be calculated from $\partial C_{i,j,k}$ and **B** for a large number of cells $C_{i,j,k}$, which makes the normal boolean operations unpractical. The article illustrates with examples the immersion of B-Rep models in Fixed Grids, visits the downstream results of the stress-strain calculations using FG and explains how this approach is used in Product Design Optimization.

Key words: meshless methods; geometric modelling; orthogonal boolean operations; Fixed Grid, finite element analysis.

Glossary

B	A PL 2-manifold without border (a 2D object).
S	S is the union of B and its interior, hence $\mathbf{B} = \partial S$ (S is a 3D object).
F	Face of B , F is PL.
$C_{i,j,k}$	Cubic, <i>i</i> -th, <i>j</i> -th, <i>k</i> -th cell in the X , Y , and Z directions respectively, with faces parallel to the XY , XZ and YZ planes ($C_{i,j,k}$ is a 3D object).
FG	The collection of the $C_{i,j,k}$, with $i, j, k \in [1..N]$
$NIO_{i,j,k}$	The portion of S confined to $C_{i,j,k}$, i.e., $NIO_{i,j,k} = C_{i,j,k} \cap S$ (a 3D object).
$B_{i,j,k}$	The boundary of $NIO_{i,j,k}$, i.e., $B_{i,j,k} = \partial NIO_{i,j,k}$ (a 2D object).
H	Face of $C_{i,j,k}$, H is PL, $\mathbf{H} \subset \partial C_{i,j,k}$.
$L(\lambda)$	Parametric half-ray $L(\lambda) = P_0 + \lambda \cdot v$, $\lambda \geq 0$.
Event λ	Parameter value of half – ray $L(\lambda)$. For convenience, it has associated the point $p=L(\lambda) \in \mathbf{R}^3$ and the faces F or H such that $p \in F$ or $p \in H$.

I, O, NIO	Characteristic of a $C_{i,j,k}$ cell, which is Inside, Outside, or Neither Inside nor Outside, respectively, of S .
Γ_H	Planar closed counterclockwise oriented Jordan curves lying on a face H ($\mathbf{H} \subset \partial C_{i,j,k}$).
H_{XZ}^+ / H_{XZ}^-	Plane in cell $C_{i,j,k}$ perpendicular to Y , with largest / smallest y coordinate. H_{ZY}^+ , H_{ZY}^- , H_{XY}^+ and H_{XY}^- are defined in analogous way.
genus, genera	The number of through holes (singular, plural) in S .

1- Introduction

This article discusses the construction of a valid Boundary Representation for the part of a solid **S** confined to each cell $C_{i,j,k}$, hence of $\partial(C_{i,j,k} \cap S)$, for cell arrangements containing large numbers of cells. The algorithm exploits the convexity of $C_{i,j,k}$, and a pre-condition on the size of $C_{i,j,k}$, which precludes disconnected borders in $\partial(C_{i,j,k} \cap S)$. As a result, the algorithm is not the repetition of a general Boolean Intersection one.

Fixed Grid methods for Finite Element Analysis require the immersion or representation of a Boundary Representation **B** of the object to analyze, **S**, using a Fixed Grid or orthogonal array of regular equally sized cells $C_{i,j,k}$. Each cell in the array is labeled as being inside, outside or neither inside nor outside (I, O, NIO respectively) of **S**. For NIO cells a certain estimation of their volume inside / outside **S** is required in order to set up structural, thermal, or other equations of interest. This estimation is best served by calculating a Boundary Representation of $NIO_{i,j,k} = C_{i,j,k} \cap S$. Traditional methods using solid modelling boolean operations are inconvenient in this domain, since the number of solid – solid intersections is $O(N^3)$, where **N** is the number of cells per **X**, **Y**, or **Z** axis of the Fixed Grid. Therefore, 2-manifold construction techniques suited for orthogonally placed solids (the cells) against flat faced polyhedra (the solid to analyze), which accelerate the representation of an arbitrary solid by using I, O and NIO cells are presented in this paper. The algorithms and results of such techniques are presented, along with applications of the representations obtained in FEA scenarios. The implementation of the algorithms uses a

paradigm with exact arithmetic through the CGAL (Computational Geometry Algorithm Library).

2- Literature Review

Meshless methods for Finite Element Analysis are used as an alternative to the subdivision of the solid S into elements of varying size, position and orientation such as tetrahedra, blocks, etc. (see [1,2]). In the Fixed Grid meshless method, the subdivision of the solid S is substituted by a regular, anchored subdivision of R^3 . Although the subdivision of R^3 might be irregular (e.g. using octrees), or be floating, it must be noticed that a regular subdivision of R^3 reduces the computational intricacies of the algorithms and the special operations that are to be expected when different cell sizes, positions and shapes are present.

In the Fixed Grid method, the solid S is immersed in the Fixed Grid $FG = \{ C_{i,j,k} \}$. This implies that each cell $C_{i,j,k}$ must be classified respect to S as Inside (I), Outside (O) or Neither Inside nor Outside (NIO). For FEA calculations NIO cells require a full representation of the geometrical and topological structure of each $NIO_{i,j,k}$. If Boundary Representation is used, the $B_{i,j,k} = \partial(C_{i,j,k} \cap S)$ is required. Boolean operations like calculating $C_{i,j,k} \cap S$ are an elementary matter in current geometric modelling. However, the number of repetitions required by the immersion of S in FG (minimal order 10^6), makes optimization compulsory.

The present work addresses the construction of topologically and geometrically valid Boundary Representations (see [3]) for the $NIO_{i,j,k}$. The optimization proposed exploits the fact that the $C_{i,j,k}$ are convex, connected, and have a **PL connected** boundary. Also, a small enough cell size for $C_{i,j,k}$ is requested, for guaranteeing the $B_{i,j,k}$ s to have some connectedness properties. More specifically, each $B_{i,j,k}$ is not only connected (i.e. consists of a single shell), but also each one of their **FACES** has connected border (i.e. is bounded by a single LOOP). The $B_{i,j,k}$, however, are not required to be convex.

The software presented in this article was programmed in C++, on top of CGAL (Computational Geometry Algorithm Library [5], [6]), using the exact arithmetic paradigm of CGAL. It pre-processes Boundary representations of Objects and constraints for a Fixed-based FEA solver ([1,2,7]).

In the literature reviewed, Structural Optimization (SO) is a follow up on the application of both mesh-based and meshless FEA methods ([8,9]). SO implies the change (usually thinning) of the solid S as per iso-stress implicit surfaces ([10]) with particular values. The Genetic Algorithms assume the application of parameters for judging the individuals of the specie to which S belongs. The topology of the resulting B-Rep B is, however, capital in the manufacturing of S ([11]), among other aspects. Therefore, the efficient construction of topologically and geometrically correct B-Reps, both for S and the $B_{i,j,k}$ NIO elements is central to the development of FG methods. This is the rationale of the present article.

In section 3 the methodology for construction of B-Reps $B_{i,j,k}$ is explained, while section 4 presents results in the realm of Fixed Grid FEA calculations. Section 5 concludes the article.

3- Methodology

The following discussion examines the required preconditions for the proposed algorithm, some important simplifications stemming from such preconditions, numerical or software characteristics, and the use of the prescribed algorithms.

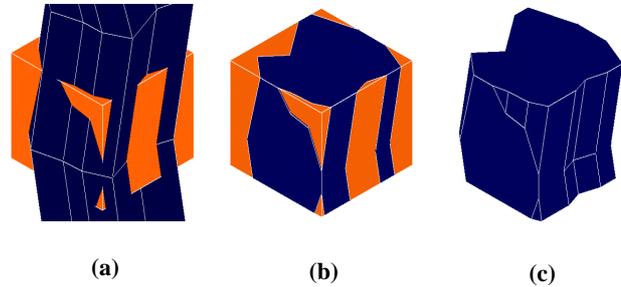


Figure 1 : (a) Piece of B-Rep B and Cell, (b) Traces of B on Cell, (c) Intersection of Cell and B-Rep B

3.1 – Pre-Conditions on Grid and B-Rep

Figure 1-(a) shows a solid set S and a particular cell $C_{i,j,k}$. The first goal is to calculate each loop Γ_H , i.e. the boundary of the intersection of $B_{i,j,k}$ with face H of $C_{i,j,k}$. All Γ_H are assumed to be connected ruling out the situation shown in Figure 1-(b), where the blue portion of the front-right face is disconnected. Figure 1-(c) displays the result $C_{i,j,k} \cap S$ for this (forbidden) case.

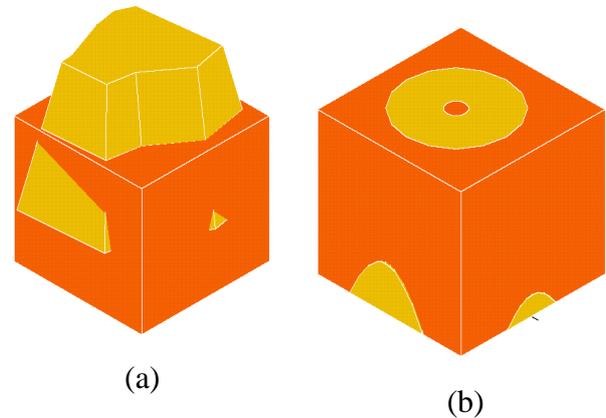


Figure 2 : Undesirable situations (unconnected NIO Face Boundaries) due to large cell size.

Likewise Figure 2 displays two undesirable cases. Figure 2-(a) shows that the loop Γ_H does not intersect the boundaries ∂H of some of the cell faces H . In the situation depicted in Figure 2-(b) the Γ_H loop is non-connected, as an internal hole is present. The natural way to ensure the restrictions here imposed is the reduction of the cell size to the point in which such illegal conditions disappear.

3.2 – Events on Half - Rays

Given the orthogonal character of the $C_{i,j,k}$ cells, a grid of half - rays parallel to the principal directions X, Y, Z is set up. Figure 3-a displays typical members of the families of half-

rays, originating on regular grids of points on the main planes XY, YZ, XZ and given by the formulae $L_Z(\lambda)=P_{XY} + \lambda .w$, $L_X(\lambda)=P_{ZY} + \lambda .u$, $L_Y(\lambda)=P_{XZ} + \lambda .v$, respectively. As an example, in the v direction, each point $p \in L_Y(\lambda)$ corresponds to an event, that is noted by λ , the parameter at which P occurs on $L_Y(\lambda)$. Later on, the events λ will be also related to non-orthogonal half-rays.

Notice also that the regular pattern of the FG (see Figure 3), allows for conducting the *point-in-polytop* inclusion test in a *per row* form, in contrast with the *per point* approach in [4], rendering a computational advantage.

The construction of each face of $B_{i,j,k}$ parallel to the planes XY, YZ, XZ , is conducted by an automaton (see Figure 3-b). In the case of $L_Z(\lambda)$ the automaton starts at $\lambda=0$ in the XZ in the point $L_Z(0)$ (without loss of generality) outside B , following $L_Y(\lambda)$. When the automaton registers an intersection of the ray with a face F of B , it will follow the events taking place on faces of $C_{i,j,k}$ parallel to the planes XY and YZ . These events involve faces of type H and F . The automaton proceeds building the loop on the faces H of $C_{i,j,k}$. When the original event is found, the loop is closed, and a face of $B_{i,j,k}$ of the type H is completed. The faces of the type B (non orthogonal) in $B_{i,j,k}$ are found by traversing the border of the 2-manifold formed by the orthogonal (parallel to the main planes) faces of $B_{i,j,k}$. The B-Rep may then be completed.

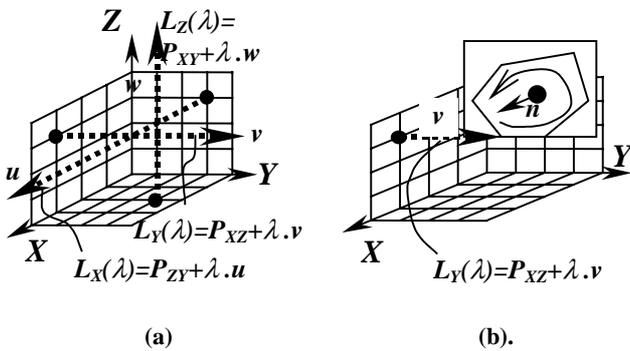


Figure 3 : (a) Grid and Half-Ray notation. (b) Automaton Iteration builds a face of $B_{i,j,k}$ parallel to the plane YX .

Figure 4 shows the follow up that the method exerts on a half-ray, originating in this case in the XZ plane. As the half-ray crosses the space subdivision of the cells $C_{i,j,k}$ (faces H), events of the type λ^* (grid events) are said to occur to it. As the half-ray hits a face F of B , events of type λ (B-Rep events) are said to occur. The point $p = L_Z(\lambda)$ and the faces (of types H or F) that each event compromises are associated to it. As the half-ray travels embedded on each face, the analytical form of $L_i(\lambda)$ obviously changes.

3.3 – Half-Ray Transitions

The algorithm proposed below requires a complete inventory of the possible events that could happen on a half – ray, in the

presence of a Fixed Grid FG , and a Boundary representation B . Table 1 shows the possible transitions that a half - ray may experience when its present status is OUTSIDE the B-Rep.

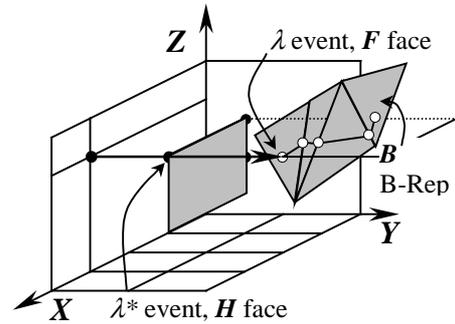


Figure 4 : Grid and B-Rep Events

Dim. of Simplex whose interior is hit by the half-ray	Intersection dimension	Figure
2	0	
1	0	
0	0	

Table 1 : OUTSIDE – INSIDE Transitions.

Table 1 classifies the transition from OUTSIDE to INSIDE (the opposite transition is symmetric), according to the dimensionality of the simplex whose interior is hit by the half-ray. The convention that the interior of a 0-simplex is the simplex itself has being used.

Table 2 displays the OUTSIDE – OUTSIDE transition, occurring when the half-ray has tangentially hit a 0-simplex, while staying outside the B-Rep.

Dim. of Simplex whose interior is hit by the half-ray	Intersection dimension	Figure
0	0	

Table 2 : OUTSIDE – OUTSIDE Transition.

The OUTSIDE – OUTSIDE case is conceptually different from the last one in Table 1. However, the calculations are the same in both cases. The B B-Rep forms a solid angle in the apex of the conic region (Figure 5), thus dividing a sphere in two solid angles (internal, external). The decision as to whether the half-ray enters or only touches the solid is the same as the decision as to whether the half-ray intersection points with the sphere lie on the same or different regions of the sphere. In Figure 5-(b), left, both intersection points hit in the same region (OUTSIDE). The half-ray, therefore, does not change status. In Figure 5-(b), right, the intersection points fall in different regions (OUTSIDE, INSIDE), producing an OUT-IN (or IN-OUT) transition.

Table 3 displays all possible OUTSIDE – BORDER transitions. Again, the transitions BORDER-OUTSIDE, INSIDE – BORDER or BORDER - INSIDE, are similar. These transitions produce a simplex of higher dimensionality (1-simplex) as the normal cases in Table 1.

Dim. of Simplex whose interior is hit by the half-ray	Intersection dimension	Figure
0	1	
1	1	

Table 3 : OUTSIDE – BORDER Transitions.

Dims. of (i) Initial Embedding, (ii) Entering Topology (iii) Final Embedding:	Figure
2 1 2	
2 0 1	
1 0 2	
1 0 1	

Table 4 : BORDER - BORDER Transitions

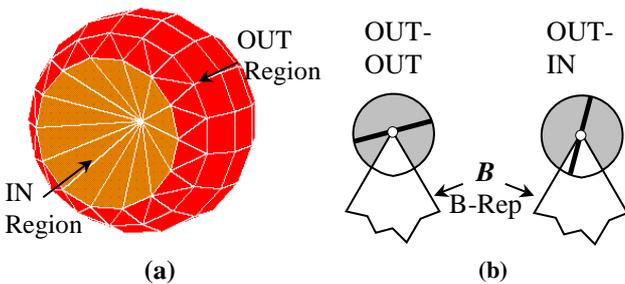


Figure 5 : (a) Solid angle neighborhood of a vertex. (b) Subdivision of the solid angle neighborhood.

Table 4 displays all possible BORDER – BORDER transitions. To classify them, dimensionalities of the three simplexes involved are recognised: (i) the simplex in which the half-ray is initially embedded, (ii) the simplex hit by the half-ray in the transition, and (iii) the terminal simplex of the half-ray transition. Strictly speaking, the transitions in Table 4 may belong to INSIDE-INSIDE ones. However, for the purposes of programming, it is more convenient to refer to them as exceptions, and to set the algorithms to act as facing exceptional situations.

3.4 – Face Construction for $NIO_{i,j,k}$

The boundaries $B_{i,j,k}$ are formed by two types of faces: (i) the

ones lying on faces H of $C_{i,j,k}$, and (ii) the ones lying on faces F of the B-Rep B . Faces of type (i) are parallel to the coordinate planes, and are calculated first, by the automaton. The faces of type (ii) are then calculated to complement the ones in (i), closing the manifold $B_{i,j,k}$.

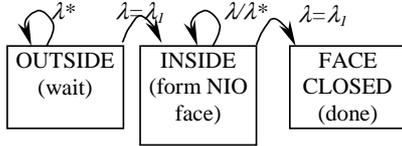


Figure 6 : Automaton for building an orthogonal (H) face of a $B_{i,j,k}$.

3.4.1 - Orthogonal Face Construction

The construction of type H faces follows the automaton illustrated in Figure 6: the half-ray starts OUTSIDE of B . While only events of the type λ^* (Fixed Grid divisions) occur, the half-ray stays outside of B . As the first transition of type λ is found, an H face of the NIO cell has to be formed. The automaton finds successive λ or λ^* events, which determine successive edges of the face under construction.

In the presence of the situation depicted in Figure 7 this automaton would act as follows: as the transition $\lambda = \lambda_1$ (OUTSIDE - INSIDE) is recorded, the automaton starts building a face normal to vector n , lying on plane Π_{XY} (in this case), with counterclockwise sense (since we assume only outer loops in the faces of $B_{i,j,k}$). The automaton finds successive λ events against faces F_{i1}, F_{i2}, F_{i3} , of the B-Rep or λ^* events against planes $H^+_{XZ}, H^-_{XZ}, H^+_{ZY}, H^-_{ZY}$. When λ_1 is found again, the face is closed.

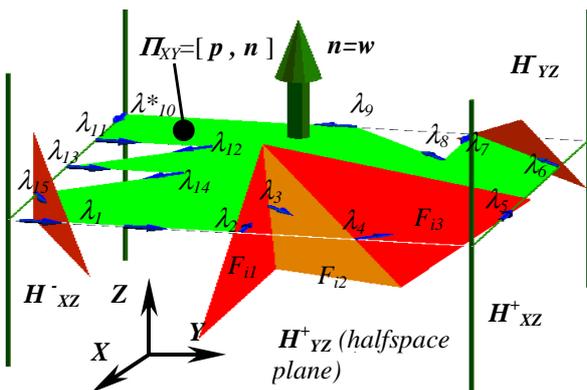


Figure 7 : Events Determining a Face Boundary

3.4.2 - Non- Orthogonal Face Construction

Figure 8 displays a NIO cell, which is mainly filled up in the upper part. The automaton (Figure 6) for the orthogonal faces of $B_{i,j,k}$ delivers faces in the form of sequences of events. For instance $F^+_{XY} = [\lambda_1, \lambda_2, \dots, \lambda_9]$ (Figure 7). These faces lie on planes $H^+_{XZ}, H^-_{XZ}, H^+_{ZY}, H^-_{ZY}, H^+_{XY}$ and H^-_{XY} . However, these faces do not close the 2-manifold $B_{i,j,k}$.

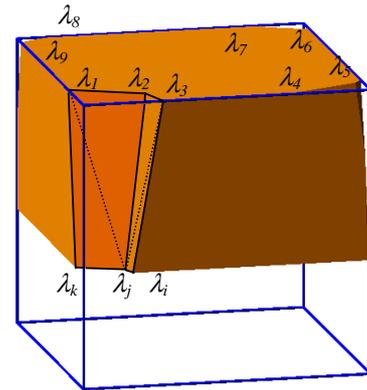


Figure 8 : Triangulation of Cell B-Rep outside the orthogonal face $F^+_{XY} = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_9]$.

The portion corresponding to $F = [\lambda_1, \lambda_k, \lambda_j, \lambda_i, \lambda_3, \lambda_2]$ is still missing, and it is not a planar face (Figure 8). On the other hand, the information of which of these faces are present is retrieved from the B-Rep B . With this B-Rep information, the sequence F is (for this example) divided into $F_1 = [\lambda_1, \lambda_k, \lambda_j, \lambda_2]$ and $F_2 = [\lambda_j, \lambda_i, \lambda_3, \lambda_2]$, which correspond to planar, non-triangular faces. A triangulation then renders $[\lambda_1, \lambda_k, \lambda_j], [\lambda_1, \lambda_j, \lambda_2], [\lambda_j, \lambda_i, \lambda_3]$ and $[\lambda_j, \lambda_3, \lambda_2]$ as the non-orthogonal faces.

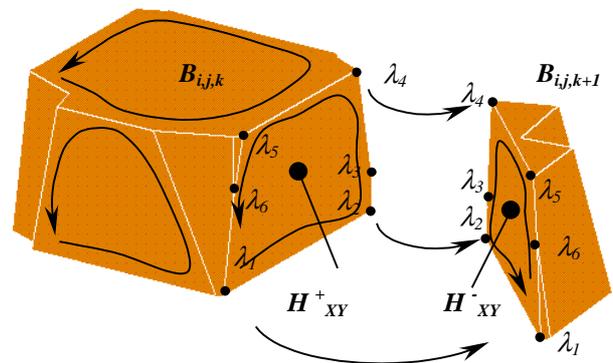


Figure 9 : Borrowing of loops contained in orthogonal between neighboring cells.

3.6.3 – Face Sharing between $NIO_{i,j,k}$ B-Reps.

Figure 9 shows that faces lying on planes $H^+_{XZ}, H^-_{XZ}, H^+_{ZY}, H^-_{ZY}, H^+_{XY}$ and H^-_{XY} need to be calculated only once in consecutive $B_{i,j,k}$ B-Reps, since they are the inversion of each other in neighboring $C_{i,j,k}$ cells. $B_{i,j,k}$ and $B_{i,j,k+1}$ share the orthogonal faces $H^+_{XY} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6]$ and $H^-_{XY} = [\lambda_6, \lambda_5, \lambda_4, \lambda_3, \lambda_2, \lambda_1]$.

4- Results

Figure 10 shows the immersion of the Vertebra data set in the Fixed Grid, using the methodology just presented. The B-Rep input data set was produced in VRML format in the

DigitLAB software ([12,13]), from a 3D slice contact digitization of a cow vertebra.

The file was converted to OOF format, appropriate for the CGAL library ([14,15]). All algorithms described here were programmed on CGAL. The test of inclusion in solid angles (Figure 5) is already implemented in CGAL. The arrows and icons appearing in Figure 10 correspond to force and cinematic constraints, respectively, specified on the vertebra.

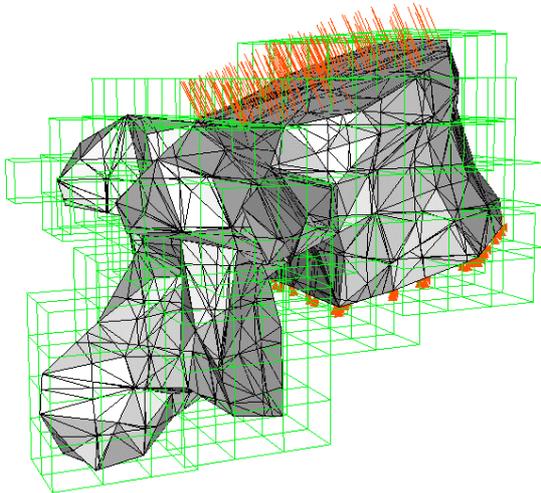


Figure 10 : Immersion of the Vertebra Model in a Fixed Grid

Figure 11 presents the corresponding immersion of a piston data set. Again, constraints such as force, pressure and position were added. The linear system for the computational mechanics problem was solved with the software X-BLUE (see [1,2]). One of the resulting vector fields (deformation) is displayed in Figure 12. The results of the X-BLUE solution of the system were found to be equal to the ones produced by the ANSYS FEA software. As this article refers to the geometric and topological aspects of the modelling for FEA Meshless Methods, the intricacies of the numerical solution and the extrapolation of boundary conditions for the Fixed Grid method may be found in [11,16].

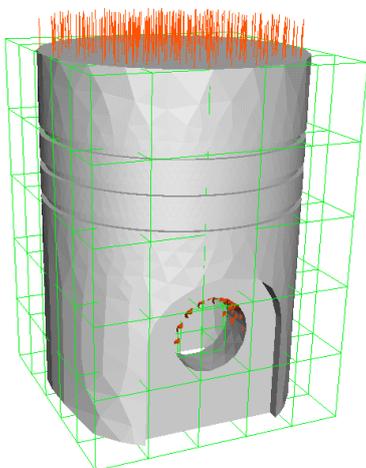


Figure 11 : Immersion of the Piston Model in a Fixed Grid

5- Applications in Product Design and Manufacturing

In [17, 18, 19, 20] the Computational Mechanics algorithms (Fixed Grid for FEA), lack a geometrical counterpart for building the immersion of S into the FG . This geometrical counterpart is the main subject of this paper. In the present section the application in Computational Mechanics of this immersion process is discussed.

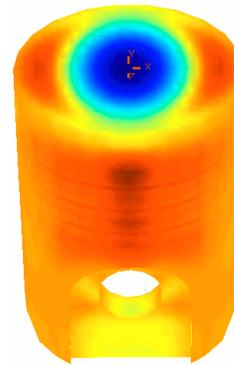


Figure 12 : Result of Deformation in the Piston data set, as calculated by the Fixed Grid method.

5.1 – Structural Optimization

Figure 13 shows a data flow diagram of the application of the immersion algorithm just discussed, in the context of Structural Optimization (presented in depth in sequel publications). A valid Boundary Representation of a solid, in triangular facet format is immersed in the Fixed Grid. In particular, the NIO cells are built. The B-Rep is accompanied by the kinematic and dynamic boundary conditions imposed by the problem at hand. The shell immersion gives as a result a set of equations of the type $A.x = B$ obtained by using the X-BLUE software (see [1, 2, 7, 8]). They are then solved rendering the behavior of S (stress, strain, pressure, temperature, velocity, etc.) under the prescribed loads or constraints.

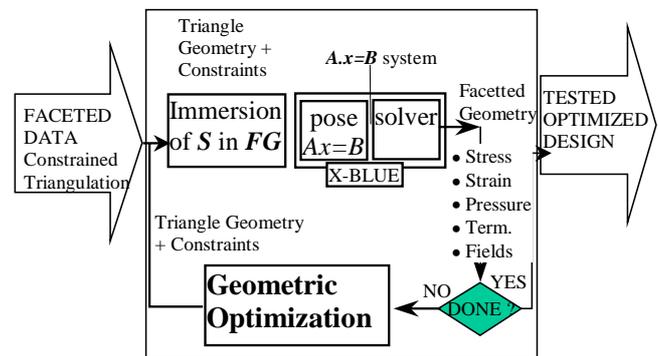


Figure 13 : Generic Optimized Analysis

Visualization and numerical evaluation steps allow the designer to define whether the object stands the tests. In negative case, or if a performance criterion is not yet satisfied, an optimization module takes control.

5.2 – Geometric Optimization

At the present time, the optimization step is performed as explained in [17] and displayed in Figure 14: a numerical performance criterion, $f_p()$ is proposed by the designer. This grading function depends on the geometric features of the current model (volume, weight, area, inertia, etc), and its performance under working conditions, expressed by the scalar fields (strain, stress, temperature, pressure, etc.). After an individual B_i is examined with this criterion and found unsatisfactory, a new individual, B_{i+1} , is generated. These generated individuals are again immersed in the Fixed Grid, analyzed (by X-BLUE or other FEA software) and evaluated under the performance criterion. The procedure for generating new objects varies. Some alternatives are: (i) small geometric perturbations on B_i are introduced, such that the topology of B_i is respected. The objects B_i and B_{i+1} differ in that one of them may be thinner than the other in particular neighborhoods. However, topological properties (such as genera) are conserved. (ii) topological perturbations are generated by lego-like elementary changes. This is the case presented in [17] where the authors use Genetic Algorithms. The topology of the objects B_i and B_{i+1} may differ significantly (for example, they may have different genera). (iii) large geometrical perturbations are generated leading to significant topological changes. Although this approach traverses rapidly towards optimization, it generates at each step large numbers of topologically unfeasible objects, which must be detected and corrected. Due to its algorithmic difficulty, it is still in experimental stages. In any of (i), (ii) or (iii), steering from the user is advisable.

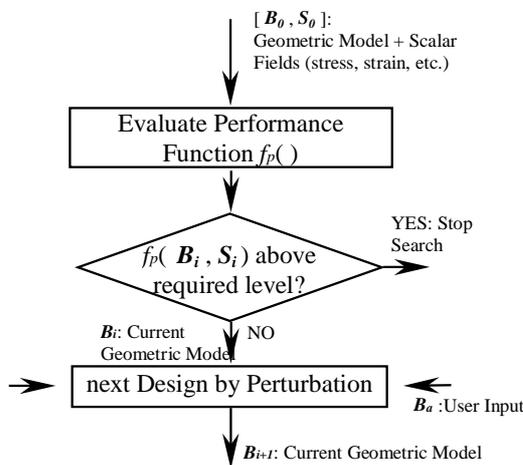


Figure 14 - Geometric Optimization

The prediction of the behavior of a piece under working conditions has applications, not only in the obvious realm of product design, but also in the area of virtual and augmented reality, as the differential equations solved correspond to the modelling of the situation, under the laws of fluid mechanics, chemistry, solid mechanics, thermodynamics, etc. If the relevant differential equations can be solved for a certain scenario, then the phenomenon might be displayed in Virtual and Augmented Reality environments.

6- Conclusions

Geometric Modelling, Data Export and Import, and Meshing occupy more than 98% of the resources used in Finite Element Analysis. Numerical solution of the $A.x=B$ linear system, visualisation, etc. occupy the remaining 2%. The actual numerical solution of the differential equations via Linear or Non-Linear solvers appear to be possible only after considerable struggle with the geometric aspects of FEA. Therefore, any effort in lowering such problems would positively impact the industrial practice. Immersion of B-Reps into Fixed Grids collaborates in these efforts, because it avoids the back-and-forth geometrical exchange, as the immersion may be performed by simply clipping NIO elements that are non-operative. On the other hand, since NIO elements present properties proportional to the occupied volume, the solving algorithm tolerates inaccuracies in the geometrical representation. These considerations make research on Fixed Grid methods attractive for the early stages of industrial product design.

The aspects of geometric and topological modelling of Boundary Representations of the $NIO_{i,j,k}$ portions of a solid S immersed in a Fixed Grid FG are discussed, and their results are presented in this paper. The operation of intersecting the solid S with the cell $C_{i,j,k}$ of the Fixed Grid and calculating its B-Rep $NIO_{i,j,k}$ are accelerated by (i) assuming a small enough cell size, which precludes unconnected boundaries (in faces or loops) for the $NIO_{i,j,k}$, (ii) taking advantage of the convexity of the $C_{i,j,k}$ cells, and (iii) assuming a uniform cell size across the whole Fixed Grid. The programming was aimed to extend the X-BLUE software (for meshless FEA) developed in the CAD CAM CAE Laboratory at EAFIT University. The initial B-Reps produced and quality-controlled by using the DigitLAB software (Digitization Lab. [12 , 13]), were developed in the same research unit.

Future improvements are planned by introducing Quadtree methods in order to lower the number of $C_{i,j,k}$ cells to be generated. The advantage of lowering the time and space complexity, however, must be enabled by a larger investment in the difficulty of the algorithm.

Acknowledgements

This research was performed with the support of the Colombian Council for Science and Technology (Colciencias) and EAFIT University. Other contributors to these results along its development are: the Fraunhofer Institute for Computer Graphics (Drs. Rix, Sakas, Stork, Darmstadt, Germany), the Max Planck Institute für Informatik (Drs. Mehlhorn, Kettner, Saarbücken, Germany, participants of the CGAL project), University of Vigo (Prof. Xoan Leiceaga) and University of Sidney (Prof. Steven). Special recognition is granted to Prof. Dr. Eng. Manuel García and his assistants Luis M. Ruiz and Miguel Henao of EAFIT University, for their contribution to the results exhibited in Figure 12. Further expansion in the solution modules of the X-BLUE software are the subject of cited past and future publications.

References

- [1] García, M.; Ruiz, O. and Steven, G. On-Line Fea For Teaching And Structural Optimization Using Fixed Grid NAFEMS (Intl Assoc. Eng. Analysis Community) World Congress 2001, Lake Como, Italy. April 24-29.
- [2] García, M.; Ruiz, O. Steven, G. Engineering Design Using Evolutionary Structural Optimization Based On Iso-Stress-Driven Smooth Geometry Removal. In NAFEMS (Intl Assoc. Eng. Analysis Community) World Congress 2001. Lake Como, Italy. April 24-29.
- [3] M. Mantyla. An Introduction to Solid Modelling. Computer Science Press, Rockville, Maryland, 1988.
- [4] Preparata F. and Shamos M. Computational Geometry: An Introduction. ISBN 3-540-961313. Springer-Verlag, 3rd Ed. October 1990.
- [5] Mehlhorn K. and Näher S., The LEDA Platform of Combinatorial and Geometric Computing, Cambridge University Press, 1999.
- [6] Kettner, L. Using Generic Programming for Designing a Data Structure for Polyhedral Surfaces. Lutz Kettner. In Computational Geometry - Theory and Applications 13: 65-90, 1999.
- [7] García M., Ruiz O., Querin O. Fixed Grid Finite Element Analysis for 3D Linear Elastic Structures. In WCCM VI World Congress on Computational Mechanics & APCOM'04, Beijing - China Sept. 5-10, 2004.
- [8] García, M. and Gonzalez, C. Shape Optimisation of Continuum Structures via Evolutionary Strategies using Fixed Grid Finite Element Method. In Structural and Multidisciplinary Optimization, v.26: 92-98, 2004.
- [9] Woon S.; Tong, L.; Querin O.; Steven G.; Knowledge-based algorithms in fixed-grid GA shape optimisation. In International Journal for Numerical Methods in Engineering (58) 4: 643-660, 2003.
- [10] Jules Bloomenthal: Polygonization of implicit surfaces. In Computer Aided Geometric Design 5(4): 341-355 1988.
- [11] García M., Querin O., Kim H. and Xie M., Fixed Grid Finite Element Analysis in Evolutionary Structural Optimisation. In Engineering Computations 17(4): 427-439 2000.
- [12] Ruiz O. and Karangelis G. Boundary Representation of Anatomical Features. Ruiz, O. Karangelis, G. In Computer Graphics Topics. ISSN 0936-2770, No. 6: 24-25, 2002.
- [13] Ruiz O., Cadavid C., Granados M., Vásquez E. and Peña S. 2D shape similarity as a complement for Voronoi-Delone methods in shape reconstruction. In Journal Computers & Graphics, Elsevier, ISSN. 0097-8493, 29 (1): 81-94, 2005.
- [14] Veltkamp R. Generic Programming in CGAL, the Computational Geometry Algorithms Library, In 6th Eurographics Workshop on Programming Paradigms in Graphics, pp 127-138. Budapest, Hungary, 8 September 1997.
- [15] Fabri A., Giezeman G., Kettner L., Schirra S. and Schönherr S., The CGAL Kernel: A Basis for Geometric Computation. In 1st ACM Workshop on Applied Computational Geometry, Editors M. C. Lin and D. Manocha. pp 91-202, Springer-Verlag Lecture Notes on Comput. Sci. (1148), 1996.
- [16] Li Q., Steven G. and Xie M., A simple checkerboard suppression algorithm for evolutionary structural optimisation. In Structural and Multidisciplinary Optimization. Springer-Verlag GmbH, ISSN: 1615-147X (Paper) 1615-1488 (Online) (22) :3: 230 - 239 October 2001.
- [17] García M. and Mejía J., .Optimal Reinforcing of a Plastic Reticular Structure Using Genetic Algorithms and Finite Element Analysis. In 19th Canadian Congress of Applied Mechanics, v.1 p.198 - 201, Calgary, Canada, 2003.
- [18] Ruiz O, García M. and Cadavid C., .Integration CAD / Fixed Grids. Technical Report, EAFIT University, December 2002.
- [19] Ruiz O, García M. and Cadavid C., .”Synthesis of PL 1-y 2-manifolds for CAGD” . Technical Report EAFIT University, December 2003
- [20] Ruiz O, García M. and Cadavid C., .”CAD-FixGrid Integration Tools for Reverse Engineering and Computational Mechanics”, Technical Report I of project 12161414887, contract No. 572-2003. p.9, EAFIT University Medellín, December 2004.