VI International Conference on Adaptive Modeling and Simulation
ADMOS 2013
J. P. Moitinho de Almeida, P. Díez, C. Tiago and N. Parés (Eds)

FE ADAPTIVE ANALYSIS OF MULTI-REGIONS MODELS

LUIZ F. MARTHA[*], JOAQUIM B. CAVALCANTE-NETO[†], WILLIAM M. LIRA[‡], ANTONIO O. MIRANDA[††], AND RAFAEL A. SOUSA[*]

[*] Departament of Civil Engineering
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro - RJ, Brazil
e-mail: lfm@tecgraf.puc-rio.br, rfldesousa@hotmail.com

[†] Department of Computing
Federal University of Ceara
Fortaleza - CE - Brazil
email: joaquim@lia.ufc.br

[‡] Laboratory of Scientific Computing and Visualization (LCCV)
Technology Center, Federal University of Alagoas
Maceio - AL - Brazil
email: william@lccv.ufal.br

[††] Department of Civil Engineering
University of Brasilia (UnB)
Brasilia - DF - Brazil
email: acmiranda@unb.br

Key words: 3D Adaptive Analysis, Finite Element Method, Mesh Generation, Multi-regions.

Abstract. This work presents a methodology for adaptive generation of 3D finite element meshes using geometric modeling with multi-regions and parametric surfaces, considering a geometric model described by curves, surfaces, and volumes. The adaptive strategy adopted in this methodology is based on independent refinements of these entities. From an initial model, new sizes of elements obtained from numerical error analysis and from geometric restrictions are stored in a global background structure, a recursive spatial composition represented by an octree. Based on this background structure, the model curves are initially refined using a binary partition algorithm. The discretization of curves is then used as input for the refinement of adjacent surfaces. The surface discretization also employs the background octree-based refinement, which is coupled to an advancing front technique for the generation of an unstructured triangulation. Surface meshes are finally used as input for the refinement of adjacent volumetric domains. In all stages of the adaptive strategy, the refinement of curves, surface meshes, and solid meshes is based on estimated numerical errors associated with the mesh of the previous step in the adaptive process. In addition, curve and surface refinement takes into account curvature information. An example is presented in order to validate the methodology proposed in this work.

1 INTRODUCTION

In numerical simulations using the Finite Element Method (FEM), two important aspects to be considered are the automatic generation of the model's finite element mesh and the definition of the level of refinement associated with this mesh. For the first aspect, there are a variety of algorithms with different techniques to generate planar, surface, and volumetric meshes. The second aspect, i.e., the level of refinement, is usually defined manually by a specialist based on his/her own experience. However, this refinement should consider the fact that the density of the generated elements varies according to the gradient of the obtained solution, which is initially unknown. In this context, this work presents a methodology to automate the refinement process of three-dimensional meshes based on adaptive technique procedures. In this work, the proposed methodology is applied to stress analysis of solid structures using a displacement-based finite element formulation. However, since the methodology essentially treats geometric modeling and mesh generation aspects of the problem, it could be used in other types of 3D finite element simulation.

Adaptive procedures try to automatically refine and coarsen a mesh, relocate its nodes, or adjust its cells to improve response accuracy. Usually, the computation begins with a trial solution obtained from a coarse mesh. The discretization error of this solution is estimated. If it fails to satisfy a prescribed accuracy metric, adjustments are made to achieve the desired solution with minimal effort. Common procedures are [1,2]: local/global refinement and/or coarsening of a mesh (h-refinement), relocating or moving a mesh (r-refinement), and locally varying the polynomial degree of elements (p-refinement).

Some strategies have been proposed to efficiently automate the 3D mesh refinement process. These strategies can be divided in two approaches: local and global refinement. In local refinement, the process uses an initial mesh and locally, using a set of elements, refines or coarsens elements in the mesh. Most works in the literature are based on this approach, as described ahead. Kallinderis and Vijayant [3] and Muthukrishnan [4] present an adaptive grid scheme based on the division/deletion of tetrahedral cells. Golias and Tsiboukis [5] and Golias and Dutton [6] employ a set of topological Delaunay transformations of tetrahedral elements and a technique for node reposition. Lee and Lo [7, 8] approach mesh refinement by inserting additional nodes at the midpoint of the longest or quasi-longest line segment of the mesh that bisects the original edges to generate new elements. In Merrouche [9], the mesh adaptation is achieved by a 3D bisection method. De Cougny and Shephard [10] present an adaptive scheme based on subdivision patterns (for refinement), edge collapsing (for coarsening), and mesh optimization (following refinement and coarsening). Lee et al. [9] only increase the order and density of 3D finite element meshes. Lee and Xu [11] generate a surface mesh for the mid-surface of the thin-walled structure, controlling element size, and convert the surface mesh to a 3D solid mesh by extrusion. More recently, Zhang et al. [12, 13] generates tetrahedral and hexahedral meshing in multi-material domains using grid-based method that employs an octree structure, refining meshes also locally.

In global refinement, on the other hand, at each refinement step, the entire mesh is deleted and another is generated based on new sizes of elements obtained from a discretization error estimation analysis. This process is used by Kettil et al. [14] only in regions with complicated parts. Hughes et al. [15] refine the structured meshes with NURBS surfaces. Our work presents a methodology that employs the global h-refinement approach.

A previous paper [16] proposed a two-dimensional self-adaptive strategy that was able to perform simulations involving automatic generation of meshes and adaptive methods. Other works have considered the same problem through different approaches, such as the study by Mark Shephard's team [17], from the Rensselaer Polytechnic Institute. Cavalcante-Neto [18] proposed a technique for the generation of volumetric meshes of tetrahedral elements for arbitrary region domains. Combining this technique with the implementation of 3D error estimators, the authors defined a prototype of an environment for adaptive generation in three dimensions. However, in this previous work, the complete process was not performed automatically, i.e., the mesh was generated independently from the error estimation and had to be manually combined. Moreover, despite treating multi-regions, it was not very efficient, and parametric surfaces, which are used in several types of simulations, were not considered.

This paper aims to present a methodology for adaptive generation of three-dimensional finite element meshes, using geometric modeling with multi-regions and parametric surfaces. Basically, the whole mesh adaptive process involves three steps: (1) analysis of a finite element (FE) model with discretization error estimation; (2) construction of a background structure to store new FE sizes that take into account the estimated discretization error and curve and surface curvature; and (3) hierarchical refinement of a FE model that is represented geometrically by curves, surfaces, and volume regions. This process may be repeated until a desired maximum allowed error metric is achieved. The methodology described herein covers only the last two steps, since discretization error estimation can be computed through different processes [1]. In this work, discretization error estimation is based on a standard technique used in the literature [19], in which the error is evaluated through the difference between stress field computed using conventional FE procedures and stress field obtained by means of more accurate recovery procedures (e.g. ZZ, SPR, or REP) [19-22].

The paper is organized as follows. Next section explains the proposed adaptive refinement strategy. Section 3 describes all the steps required to generate the background data structure that is used to define FE sizes in the adaptive process. The following section presents the hierarchical refinement of curves, surfaces, and volumes. An example of adaptive refinement is presented in Section 5. Finally, in Section 6, there is a conclusion.

2 ADAPTIVE REFINEMENT STRATEGY

The three-dimensional geometric model has a topological description of the vertices, curves, surfaces, and regions, as well as an associated geometric description, which consists of the coordinates of the vertices and the mathematical representation of the curves and surfaces. The geometric model can contain many regions. In this environment, the attributes of the simulation, such as the properties of the materials, loads and restrictions, are associated with the geometric entities. In this framework, the entities of finite element mesh (nodes and elements) automatically receive the attributes of the geometric entities that are related to. Using this approach, it is possible to create new meshes without losing the attributes.

Figure 1 illustrates the automatic adaptive strategy of the proposed refinement process. The input data are the initial volumetric mesh of problem in question and the geometric entities (curves and surfaces), and as well as their associated attributes. Initially, this mesh is numerically analyzed, the information required to initiate the adaptive procedure. Such information basically consists of numerical discretization errors associated with each volumetric element of the mesh. From these errors, the need for adaptive refinement is verified. If the results converge, the adaptive process is concluded with a final discretization.

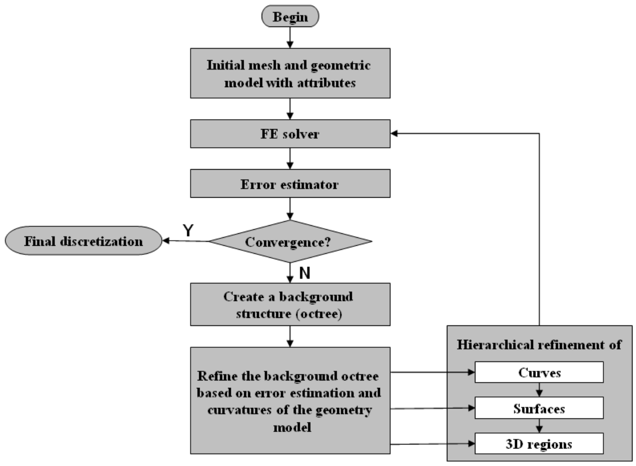

Figure 1: The proposed adaptive refinement process.

If convergence is not reached, the sizes of the new elements are computed based on the estimated discretization error. All the resized data are stored in an auxiliary background structure. Although many background structures are published in the literature, as reviewed by Quadros [23], the present work uses a background octree structure, which has the advantage of not only allowing fast search procedures down to internal leaves but also representing the desired size of the elements defined by the size of the internal leaves. For these reasons, an octree is used to support the discretization of curves, surface meshes, and the volumetric mesh.

In addition to discretization error estimation, curve and surface discretization is also required, especially when the curves and surfaces present high curvatures. In such locations, the meshes should be locally refined. Therefore, new element size data, based on the geometric information of the curves and surfaces, are computed and stored in the background octree. After this procedure, the background octree is internally finalized to provide a better transition between regions with elements of highly varying sizes.

Using the size information from the background octree, the next step consists of a three level hierarchical approach to create a new volumetric mesh. First, the curves are refined based on the size of the elements stored in the octree structure. This refinement subdivides the curves into segments with sizes consistent with those of the discretization error analysis and geometric criteria. After refining the curves, the meshes associated with each of the model's surfaces are discretized using an advancing front scheme in parametric space. This meshing scheme starts by subdividing curves on the boundary of each surface. Geometric curvature information is considered in surface refinement because the background octree takes this information into account. The last stage of the adaptive refinement process is related to the discretization of the domains of the model's regions. Such discretization uses a 3D advancing front technique that starts from the triangulated meshes associated with the boundary surfaces of each 3D region, also considering the sizes of the elements provided by the background octree. As can be seen, this adaptive meshing methodology supports multi-regions in a consistent manner, considering curve and surface curvature information in addition to the estimated discretization error. Finally, a new discretization error analysis is performed to assess the quality of the results. If convergence is not obtained, the whole adaptive process is repeated as described above. The next sections summarize the proposed refinement strategy.

3 THE BACKGROUND OCTREE

An octree is a tree data structure based on a cell with eight children. Each cell of an octree represents a cube in the physical space. Each child represents one octant of its parent. On the leaves of the tree are the computational cells of the grid. In this work, the background octree has two main objectives. The first is to develop local guidelines used to define the discretization of curves and surfaces. The second is to define the sizes of tetrahedral elements to be generated during the advancing front procedure. The octree generation includes four steps. Figure 2 depicts the external appearance of the background octree of a hypothetical model (Figure 2-a). In the first step (Figure 2-b), the octree is initialized based on the input mesh data, which are the new element sizes obtained in the discretization error analysis. The second step (Figure 2-c) refines the octree based on the geometric curvatures of curves. In the third step (Figure 2-d), the octree is refined based on the geometric curvatures of surfaces. Finally, in the last step (Figure 2-e), the octree is refined in order to obtain a better transition between the sizes of the elements generated in the advancing front surface refinement.

3.1 Background octree refinement based on error analysis

Initially, a bounding cube is created based on the maximum range of the three Cartesian coordinates of the input model. This cube is the octree's root cell. In the first step of the octree refinement (Figure 2-b), each discretization error result of an element is used to determine the local depth of the subdivision. A characteristic size of each element is calculated for the estimated error and the octree cell containing the element's central point is determined. If the size of the cell edge is larger than the calculated characteristic size, then this cell is subdivided into eight smaller cells. This process is repeated recursively and finishes when the size of the cell is smaller than the given size. This process is repeated for every element of the current FE mesh. The characteristic size is calculated considering an equilateral tetrahedron with the same volume of each element.

The background octree works as a density function to guide the adaptive process. It could be replaced by other functions. It has the advantage of also allowing fast search procedures down to internal leaves. It could also have a different orientation to better adapt to models that are not parallel to Cartesian coordinates. However, the bounding cube parallel to the Cartesian coordinates is easier and faster to implement and usually gives very good results.

3.2 Octree refinement based on curve curvature

In some cases, when only the discretization error is considered in the adaptive process the new generated mesh (in the following step of the process) does not respect the actual geometry of the model's curves. This behavior occurs when parts of a curve, for example, present high curvatures in a region where the discretization error is low. In these situations, it is necessary to refine the background octree based on the curvatures of the curves to preserve the original geometric characteristics of the model.

The methodology used to refine the curves of the model based on their curvatures is a one-dimensional version of the procedure applied to discretize the background octree. The refinement of each curve employs a recursive spatial numbering technique similar to a binary tree data structure [24].

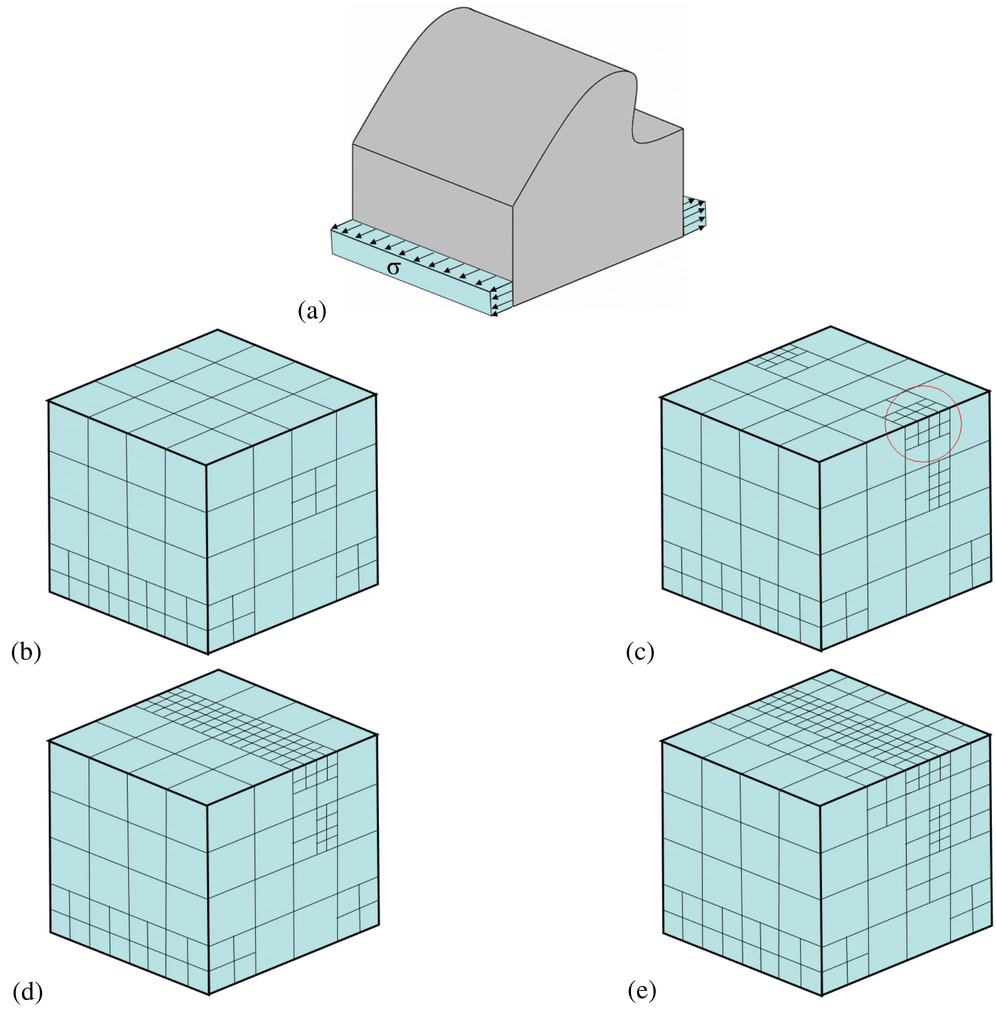

Figure 2: A hypothetical model to explain the steps of octree construction: (a) Model geometry and loading; (b) External appearance of background octree based on discretization error analysis; (c) Refinement after considering curve curvatures; (d) Refinement after considering surface curvatures; (e) Refinement after considering maximum cell size at boundary cells and maximum difference of one level between adjacent cells.

The main purpose is to generate a discretization on a curve according to its curvatures. The curvatures are calculated for specific curve segments. At first, the whole length of the curve is considered as the segment to be tested. If the curvature of the segment is lower than the maximum allowed curvature, the process is interrupted. Otherwise, the segment is recursively subdivided in two segments, and each one is tested in the same way, until the maximum curvature criterion is satisfied. At the end of this process (Figure 2-c), all the curve segment sizes and their middle points are transported to the background octree, using the same procedure explained in Section 3.1.

3.3 Octree refinement based on surface curvature

After the background octree is refined considering curve curvatures, the octree refinement is increased based on surface curvatures. This step (Figure 2-d) captures high curvatures of surfaces, computes the required element sizes and their locations, and passes this information to the background octree. The reason to perform this step is the same as the previous stage: to represent the original geometry of the model.

As in Section 3.2, the methodology applied to refine the surfaces of the model based on their curvatures is a two-dimensional version of the procedure used to discretize the background octree: a background quadtree, which is created similarly to the one presented by Miranda and Martha [25]. This way of computing the curvatures has shown to be efficient and robust, and this is the main reason for its adoption. The background quadtree generation follows some steps:

- Quadtree initialization based on given boundary edges;
- Refinement to force maximum cell size;
- Refinement to provide minimum size disparity for adjacent cells;
- Refinement to force minimum curvature difference between adjacent cells: this stage is explained ahead.

As described in detail by Miranda and Martha [25], the first step has some modifications in relation to the original 3D algorithm [26]. The second and third steps have not changed. The fourth step was added to take high surface curvatures into account.

The fourth step of the quadtree generation refines this auxiliary structure to force a minimum curvature difference between adjacent cells. Initially, the algorithm stores in each cell gradient vectors of the quadtree evaluated at the center of the cell. Then, it computes a vector normal to the surface of each cell. Finally, the algorithm obtains the cosine of the angle θ between the normal vectors of the two adjacent cells and compares it to a minimum value, $\cos\theta_{min}$. This kind of comparison is similar to comparing the angle between the normal vectors and the maximum angle. If $\cos\theta$ is smaller than $\cos\theta_{min}$, then a new cell size, H_{new}, is obtained from the current size, H_{old}, as $H_{new} = (H_{old}/\cos\theta_{min})\cdot\cos\theta$. This new size is used to locally refine the adjacent cells of the quadtree. This process is repeated recursively for every cell. The new element sizes stored in the auxiliary surface quadtree are transferred to the global background octree. At the end of this step, the background octree is refined considering the geometric curvatures of all of the model's surfaces.

3.4 Octree final refinement

The previous step can leave large octree cells in the interior of a 3D region. In the first step of this final stage, the octree is refined to guarantee that no cell in the interior is bigger than the largest cell on the boundary. This will avoid excessively large elements in the domain interior. The octree is subsequently processed to force a single difference level between neighboring cells (Figure 2-e). This leads to a natural transition between regions with different degrees of refinement. This refinement is performed by traversing the octree and examining the difference in tree depth between adjacent cells. If the difference is larger than one level, the adequate cells are refined until the criterion is satisfied.

4 HIERARCHICAL REFINEMENT

After the construction of the background octree, considering the new element sizes based on the discretization error and on curvatures of the geometric model, the final step is to re-generate the mesh of the whole model. As mentioned previously, it is assumed that the three-dimensional geometric model has a topological description of the vertices, curves, surfaces, and regions, as well as an associated geometric description, which consists of the coordinates of the vertices and the mathematical representation of the curves and surfaces. The geometric model can contain many regions. Mesh re-generation employs a hierarchical refinement of (1) curves, (2) surfaces, (3) and regions.

The methodology used to refine the model's curves is similar to the one mentioned in Section 3.2. At the beginning of the process, a curve is defined by its mathematical geometric description and by two nodes (initial and final points). Then, the curve length and middle node are obtained. From the middle node, one can determine the cell in the background octree where this node is located. A comparison is then made to verify whether the segment size is smaller than that of the corresponding cell. If the criterion is satisfied, the curve refinement process ends considering the nodes generated so far. Otherwise, the new node is inserted on the curve, this curve is subdivided in two partitions, and each one is tested in the same way, until the criterion is met.

Surface mesh generation is based on the algorithm presented by Miranda and Martha [25]. This algorithm is applied to the generation of triangular meshes on each surface with arbitrary geometry, using its parametric description. The parametric description is used because it is common and efficient, since the surface mesh is generated using two-dimensional triangulation techniques. However, additional length and angle corrections are needed to consider metric distortions between parametric and 3D Cartesian spaces. With this procedure, generated triangles present good shape in 3D space.

3D mesh generation in each closed region of the model is based on a technique presented by Cavalcante-Neto [26] and is used to obtain tetrahedral elements in arbitrary domains. Similarly to the procedure applied to generate surface meshes, this one is based on an advancing front technique coupled to a recursive spatial decomposition technique (octree). Originally, the algorithm employed an independent background octree in each 3D region to control the distribution of the node points generated in the interior. In the adaptive methodology proposed here, the global background octree is used for this purpose.

The algorithm was designed to meet four specific requirements: to avoid producing elements with poor aspect ratios; to generate meshes conforming to existing triangular meshes at the boundary of a domain; and to generate meshes exhibiting good transitions between regions of different element sizes. The input to the algorithm is a triangular surface mesh, which describes the domain to be meshed. This mesh is obtained from the surface meshes on the boundary of a 3D region to be meshed. The algorithm steps are as follows:

- A two-pass advancing front procedure is applied to generate elements. In the first pass, elements are generated based on geometric criteria, producing well-shaped elements. The background octree presented in Section 3 is used to control the sizes of the elements and the position of the interior nodes. The octree determines an ideal position for an optimal node to form a new element. This ideal position defines a search region where an optimal node for the new element may be located. This region is a sector of a sphere whose center is the ideal position and whose radius is proportional to the octree cell size. If one or more existing nodes are inside this region, they are ranked based on a solid angle criterion, in order to get the best node for the new element. However, if no existing node is found, a new node is inserted at the ideal position and an element is generated using this node. In the second pass, elements are generated based only on the criterion that they have valid topology. Here, any existing node that forms a valid new element can be used, regardless of whether it is close to the ideal position or not. However, the same quality criterion is used and the node that forms the best solid angle is chosen for the generation of the new element.

- If the advancing front procedure cannot progress, a back-tracking strategy [27] is employed to delete some elements, and the procedure is restarted. It consists basically of back tracking a few steps in the mesh generation and deleting faces that hinder the front from converging. This creates better regions where valid elements can be then generated. It is possible that the process of finding better regions may fail, for instance, if faces to be removed are part of the original boundary. When this occurs, other elements are deleted instead and the procedure is restarted. If a mesh still cannot be generated for this region, the algorithm fails and terminates. In principle, it is possible to create a boundary input mesh that forces the failure of the volume mesh generation. Such failure, however, has not yet been observed in “non-contrived” input, i.e., in any realistic input boundary meshes in many examples tested so far.
- Once a valid mesh is created, the quality of the element shapes is improved by using the standard Laplacian smoothing technique and locally deleting poorly shaped elements and those adjacent to them. The boundary contraction is then restarted.

After the generation of volumetric elements in all regions of the model as exposed above, a new error analysis is performed to assess the quality of the results. If convergence is not obtained, the whole adaptive process is repeated as described in the previous sections. The next section provides some examples of the proposed adaptive refinement process.

5 EXAMPLE

This section presents an example of adaptive 3D finite element mesh that was generated using the proposed adaptive methodology. It is important to emphasize that this paper does not aim to compute the performance of mesh generators (surface and volume) or assess the quality of the elements generated, since these tasks were covered in previous works [25, 26].

The adaptive strategy proposed in this paper results from the application of unstructured mesh generation techniques in surfaces and regions, combined with numerical errors associated with discretization. Numerical error estimators are implemented based on procedures developed for two-dimensional models [28] extended to three dimensions. These error estimators are supported by error estimation techniques widely adopted in the literature, called Superconvergent Path Recovery technique (SPR) [20,21] and Recovery by Equilibrium in Patches (REP) [22]. The numerical error estimators were implemented in a finite element numerical analysis program [29]. It is worth stressing that these error estimation techniques can be easily and directly replaced by any other technique that is more recent or efficient, as this is supported by the object-oriented organization of the analysis program. In the present example, SPR is employed for error estimation.

The presented example is a model of a bike suspension rocker [30], shown in Figure 3, that is composed by four cylinders that are connected by a central body. Uniformly distributed forces, of unitary intensity in the y direction are applied to the internal faces of the two top cylinders. Displacement restrictions are applied to the internal face of the lower cylinder.

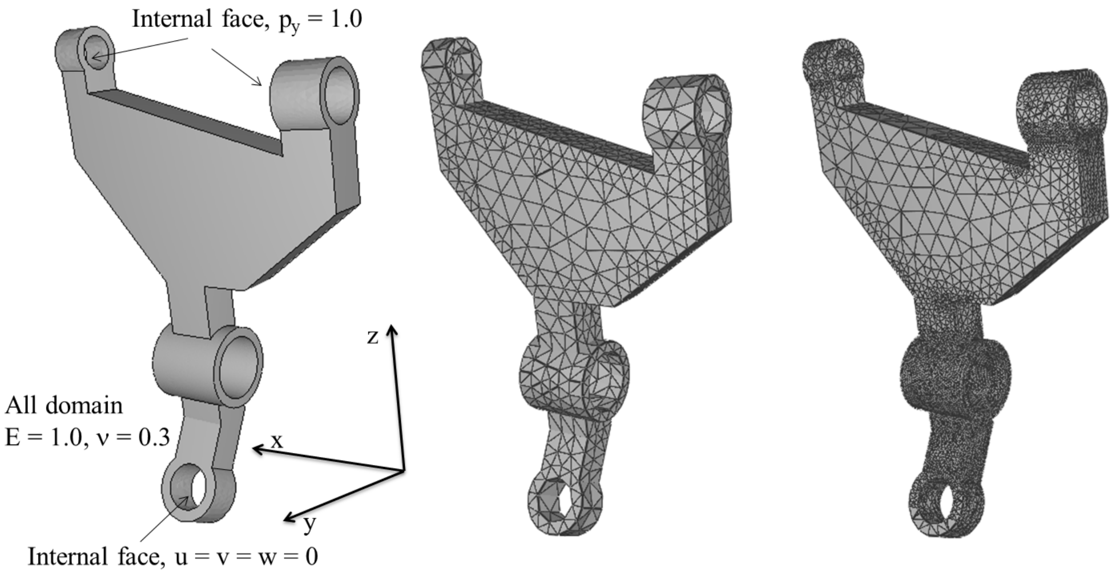

Figure 3: Example: bike suspension rocker model and finite element mesh refinement.

The adaptive refinement of this example (see Figure 3) was carried out until the target relative error (3%) was reached. The mesh is refined in the intermediate cylinder and only part of the lower cylinder, where there are concentrations of stress. The number of linear tetrahedral elements in the initial mesh is 6223 and the number of nodes is 10303. In the final mesh, the number of elements is 186238 and the number of nodes is 273633. This example demonstrates the importance of considering the curvatures of the supporting surface in the adaptive refinement, which is another characteristic of the present methodology.

6 CONCLUSIONS

This paper described a methodology for adaptive generation of three-dimensional finite element meshes, using geometric modeling with multi-regions and parametric surfaces. The mesh adaptive process involves three steps: (1) FE analysis with error estimation; (2) construction of a structure to store the new sizes of the FE; and (3) refinement of the FE model. The approach adopted is the global refinement of the whole model in each adaptive refinement, using a background octree structure. After the construction of the octree, the new model is geometrically re-discretized employing a hierarchical curve, surface and volume refinement. Some important characteristics of the proposed methodology are:

- The ability to refine and coarsen in regions of high and low response gradients.
- The use of only one background octree for all regions of the model, allowing a smooth transition between regions and elements.
- The hierarchical refinement of curves, surfaces, and volumes.

- The consideration of curvatures of curves and surfaces in the adaptive refinement.

A recent article [31] published by the authors compares results of convergence rates of the proposed methodology with results presented in the literature. It was demonstrated that the current methodology converges faster to a lower relative error, because the 3D mesh generator used has more freedom to create new elements based on desired element sizes. This characteristic generates a desired mesh with the application of only a few steps of the adaptive refinement. In contrast, using a local refinement strategy, (local) element manipulations restrict the shape quality of new elements.

Obviously, the current 3D mesh generation takes more time to create new elements, because the whole FE model must be created at each step. In this work, most of meshes were generated in less than one minute of clock time. However, in models that require a large number of elements (one million or more), time consumption can increases exponentially. While in many problems this is not an issue, it can be a limitation of the current approach that can be solved in two manners: (1) decomposing the domains into sub-domains and applying the mesh generator to each sub-domain; or (2) using a parallel 3D mesh generator. The latter option is our future work and is currently under development.

REFERENCES

[1] Mackerle, J. Error estimates and adaptive finite element methods: A bibliography (1990-2000). *Eng. Comput.* (2001) **18**:802-914.

[2] McRae, D.S. R-refinement grid adaptation algorithms and issues. *Comput. Meth. Appl. Mech. Eng.* (2000) **189**:1161-1182.

[3] Kallinderis, Y. and Vijayant, P. Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes. *AIAA Journal* (1993) **43**(8):1440-1447.

[4] Muthukrishnan, N., Shiakolas, P.S., Nambiar, R.V., and L, L.K. Simple algorithm for adaptive refinement of three dimensional finite element tetrahedral meshes. *AIAA Journal* (1995) **33**(5):928-932.

[5] Golias, N.A. and Tsiboukis, T.D. An approach to refining three-dimensional tetrahedral meshes based on delaunay transformations. *Int. J. Num. Meth. Engng*. (1994) **37**:793-801.

[6] Golias, N.A. and Dutton, R.W. Delaunay triangulation and 3D adaptive mesh generation. *Finite Elem. Anal. Des.* (1997) **25**:331-341.

[7] Lee, C.K. and Lo, S.H. Automatic adaptive refinement finite element procedure for 3D stress analysis. *Finite Elem. Anal. Des.* (1997) **25**:135-166.

[8] Lee, C.K. and Lo, S.H. A full 3D finite element analysis using adaptive refinement and pcg solver with back interpolation. *Comput. Meth. Appl. Mech. Eng.* (1999) **170**:39-64.

[9] Merrouche, A., Selman, A., and Knopf-Lenoir, C. 3D adaptive mesh refinement. *Commun. Numer. Methods Eng.* (1998) **14**(5):397-407.

[10] De Cougny, H.L. and Shephard, M.S. Parallel refinement and coarsening of tetrahedral meshes. *Int. J. Num. Meth. Engng*. (1999) **46**:1101-1125.

[11] Lee, C.K. and Xu, Q.X. A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures. *Int. J. Num. Meth. Engng*. (2005) **62**:1519-1558.

[12] Qian, J. and Zhang, Y. Automatic unstructured all-hexahedral mesh generation from b-reps for non-manifold cad assemblies. *Eng. Comput.* (2012) **28**:345-359.

[13] Zhang, Y., Hughes, T.J., and Bajaj, C.L. An automatic 3D mesh generation method for

domains with multiple materials. *Comput. Meth. Appl. Mech. Eng.* (2010) **199**:405-415.
[14] Kettil, P., Ekevid, T., and Wiberg, N.E. Towards fully mesh adaptive FE-simulations in 3D using multi-grid solver. *Computers and Structures* (2003) **81**:735-746.
[15] Hughes, T.J., Cottrell, J.A., and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Meth. Appl. Mech. Eng.* (2005) **194**:4135-4195.
[16] Cavalcante-Neto, J.B., Martha, L.F., Menezes, I.F.M., and Paulino, G.H. A methodology for self-adaptive finite method analysis using an object oriented approach. *Fourth World Congress on Computational Mechanics* (1998).
[17] Baehmann, P.L. and Shephard, M.S. Adaptive multiple-level h-refinement in automated finite element analysis. Engineering with Computers (1989) **5**:235-247.
[18] Cavalcante-Neto, J.B. Mesh generation and error estimative for finite element 3D models with cracks. *PhD dissertation, Pontifical University Catholic of Rio de Janeiro*.
[19] Zienkiewicz, O.C. and Taylor, R.L. *The finite element method*. McGraw Hill (1991).
[20] Zienkiewicz, O.C. and Zhu, J.Z. The superconvergent patch recovery and a posterior error estimates. Part 1: The recovery technique. *Int. J. Num. Meth. Engng*. (1992) **33**:1331-1364.
[21] Zienkiewicz, O.C. and Zhu, J.Z. The superconvergent patch recovery and a posterior error estimates. Part 2: Error estimates and adaptivity. *Int. J. Num. Meth. Engng*. (1992) **33**:1365-1382.
[22] Boroomand, B. and Zienkiewicz, O.C. Recovery by equilibrium in patches (REP). *Int. J. Num. Meth. Engng*. (1997) **40**:137-164.
[23] Quadros, W.R., Vyas, V., Brewer, M., Owen, S.J., and K, S. A computational framework for automating generation of sizing function in assembly meshing via disconnected skeletons. *Eng. Comput.* (2010) **26**(3):231-247.
[24] Knuth, D. *The art of computer programming.* Addison-Wesley (1997).
[25] Miranda, A.C.O. and Martha, L.F. Mesh generation on high-curvature surfaces based on a background quadtree structure. *11th International Meshing Roundtable* (2002) 333-341.
[26] Cavalcante-Neto, J.B., Wawrzynek, P.A., Carvalho, M.T.M., Martha, L.F., and Ingraffea, A.R. An algorithm for three-dimensional mesh generation for arbitrary regions with cracks. *Eng. Comput.* (2001) **17**(1):75-91.
[27] Cavalcante-Neto, J.B., Martha, L.F., Wawrzynek, P.A., and Ingraffea, A.R. A back-tracking procedure for optimization of simplex meshes. *Commun. Numer. Methods Eng.* (2005) **21**(12):711-722.
[28] Paulino, G.H., Menezes, I.F.M., Cavalcante Neto, J.B., and Martha, L.F. A methodology for adaptive finite element analysis: Towards an integrated computational environment. *Comput. Mech.* (1999) **23**:361-388.
[29] Martha, L.F. and Parente Jr., E. An object-oriented framework for finite element programming. *Fifth World Congress on Computational Mechanics* (2002).
[30] Cuillière, J., Francois, V., and Drouet, J. Automatic 3D mesh generation of multiple domains for topology optimization methods. *21th International Meshing Roundtable* (2012) 243-249.
[31] Miranda, A.C.O., Lira, W.W.M., Cavalcante-Neto, J.B., Sousa, R.A., and Martha, L.F., A 3D adaptive mesh generation approach using geometric modeling with multi-regions and parametric surfaces. *J. Comput. Inf. Sci. Eng.* (2013), accepted for publication.