Implementation of a Refinement/Derefinement Algorithm for Tetrahedral Meshes

J. M. González-Yuste*, R. Montenegro, J. M. Escobar, G. Montero and E. Rodríguez

*University Institute of Intelligent Systems and Numerical Applications in Engineering
University of Las Palmas de Gran Canaria

Keywords: 3-D triangulations, finite element, adaptive meshes, object oriented method, adaptive refinement/derefinement, data structures.

To solve problems using finite element method it is necessary to obtain a good domain discretization. Once the problem geometry has been approached by an initial mesh, it must be able to adapt itself to the singularities of the numerical solution. The refinement process introduces new elements in zones of the mesh where the solution needs to be improved. The derefinement removes any added elements by refinement where the numerical solution presents low variations. With refinement/derefinement is possible to build a partition that achieves the numerical solution with the desired precision. These techniques are specially useful in non-steady problems like those arising in environmental modelling, where the position of the singularities in the mesh depends on time.

In this work we present an implementation of a 3-D local refinement/derefinement algorithm. The refinement algorithm [1] is based on the subdivision of a tetrahedron into eight subtetrahedra [2]. The element in the mesh t_i will have associated the error indicator η_i, and it will be refined if $\eta_i \geq \gamma\eta_{max}$, being η_{max} the maximum value of the error indicator and γ the refinement parameter ($\gamma \in [0, 1]$). The derefinement algorithm is defined as the inverse of the refinement one. It can remove nodes if the difference between the numerical and the interpolated solution is lower than ϵ. The value of the derefinement parameter ϵ depends on the required precision.

The implementation of the algorithm has been carried out with *C++*. Using the *C++* classes, the elements of the mesh (nodes, edges, faces and tetrahedra) could be modelled. First of all, a very simple class was designed, called *Element*, with only few properties, but common to all of them. From this class, two classes will be inherited: *Divisible* and *Node*. *Divisible* will maintain genealogical references of the elements, that is, relationships among father elements and their sons. The other main classes, *Edge*, *Face* and *Tetrahedron* will be inherited from *Divisible*. There will be few classes more, mainly dedicated to neighbourhood relationship and memory management. With this organisation, any relation among elements can be easily established, and the implementation of the algorithm can be programmed.

In both refinement and derefinement implementations, the processes will have to pass over the data structure. On the first hand, for refinement, a procedure will mark the edges of tetrahedra that must be refined. For each marked made, the mesh conformity must be ensured in the adjacent tetrahedra according to the algorithm. On the other hand, for derefinement, the process will work inversely: it begins with all nodes marked for derefining. If any node could not be eliminated then a recursive process will unmark this node and those two nodes of its surrounding edges. When this process finishes, conformity must be ensured again, and new nodes would be unmarked if it were necessary.

Finally, several applications are presented in order to show the efficiency of the algorithm. This refinement algorithm has been successfully applied in [3] and [4] in steady problems. In addition, the new refinement/derefinement procedure allows to simulate time dependent problems.

References

[1] J.M. González-Yuste, R. Montenegro, J.M. Escobar, G. Montero and E. Rodríguez, *"Local Refinement Triangulations Using Object-Oriented Methods"*, Advances in Engineering Software, 2003, in press.

[2] R. Löhner and J.D. Baum, *"Adaptive h-refinement on 3D unstructured grids for transient method"*, Int. J. Num. Meth. Fluids, 8, 1135-1149, 1988.

[3] J.M. Escobar, R. Montenegro, G. Montero, E. Rodríguez and J.M. González-Yuste, *"Improvement of mesh quality by combining smoothing techniques and local refinement"*, Proceedings of the Ninth International Conference on Civil and Structural Engineering Computing, Civil-Comp Press, CD-ROM: Paper 21, 1-16, 2003.

[4] G. Montero, E. Rodríguez, R. Montenegro, J.M. Escobar and J.M. González-Yuste, *"Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model"*, Advances in Engineering Software, 2004, in press.