Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXVII, págs. 3059-3069 (artículo completo)
Alberto Cardona, Mario Storti, Carlos Zuppa. (Eds.)
San Luis, Argentina, 10-13 Noviembre 2008

AN H-ADAPTIVE NON-CONFORMAL 3-D UNSTRUCTURED MESH STRATEGY FOR UNSTEADY COMPRESSIBLE FLOWS.

Gustavo A. Ríos Rodriguez, Mario A. Storti and Norberto M. Nigro

Centro Internacional de Métodos Computacionales en Ingeniería CIMEC,
Universidad Nacional del Litoral, CONICET, Güemes 3450, 3000,
Santa Fe, Argentina , gusadrr@yahoo.com.ar, http://www.cimec.org.ar

Keywords: h-adaptivity, unstructured meshes, non-conformity, unsteady compressible flows, blast waves.

Abstract. An h-adaptive unstructured mesh refinement strategy to solve unsteady problems by the finite element method is described. Key features of the strategy are the non-conformity and both a prescribed updating frecuency and a maximum level of refinement for the adapted meshes. The 1-irregular node refinement constraint is extended to tetrahedral meshes to ensure a smooth grading in the elements size. A particular 1:8 partitioning sequence, which shows to keep bounded the quality decrease, is applied to tetrahedral elements. The type of element is not changed and no transition templates are used, therefore hanging nodes appear in the adapted mesh. The elements' refinement algorithm is described in some detail. The adaptivity strategy is implemented in C++ code using both the STL (Standard Template Library) and Boost Multiarray (http://www.boost.org/) libraries for the managment of the data structure. This code is used to solve the Taylor-Sedov spherical blast wave problem on an unstructured mesh of tetrahedra.

1 INTRODUCTION

Nowadays, the benefits of mesh adaptivity in the solution of computational fluid dynamics problems by the finite element method are well recognized. Mesh enrichment (*h*-adaptivity) and nodes relocation (*r*-adaptivity) are common procedures to achieve this. Not to modify the fluid dynamic solver is one of the advantages of the former procedure.

In this work, an *h*-adaptive element-based unstructured mesh strategy to solve unsteady compressible flows by the finite element method is briefly described. Then it is applied to solve a three dimensional blast wave problem. The refinement algorithm has been already used to solve steady 2-D and 3-D problems and an axisymetric unsteady compressible flow problem in Ríos Rodriguez et al. (2005, 2006). Here, it is the first time that it is used to solve a 3-D unsteady problem using tetrahedral elements. To keep bounded the quality decrease of the mesh is the main driving force of the refinement strategy. It has been shown in a previous work by Ríos Rodriguez et al. (2008), that the best choice (from both the quality and computing cost points of view) to regularly refine a tetrahedron in 8 sub-tetrahedra is to choose the shortest diagonal of the inner octaedron that arises in the regular 1:8 partitioning process. So, this is the element refinement pattern being used in this work.

Besides, the adaptivity strategy has to manage hanging nodes since no transition elements are used to match zones with different levels of refinement. Therefore the solution must be constrained on these nodes. The 1-irregular node constraint amongst neighbouring elements extended for tetrahedral elements and is applied to ensure a smooth transition in the elements size. The mesh adaptation stage is sequentially performed on a single processor of a Beowulf cluster (but not the server node) while the fluid dynamic problem solver (PETSc-FEM, Storti et al. (1999-2007)) runs on several nodes. The spherical blast wave problem of Taylor-Sedov is described and solved.

2 REFINEMENT ALGORITHM

For unsteady problems the mesh must be refined and derefined every few time steps to track the structures of the flow through the computational domain. Therefore adaptivity algorithms for this kind of problems must be computationally inexpensive. An element-based technique is used to refine the elements of the mesh (i.e., both the selection and refinement are applied in an element-wise fashion). Only a regular 1:8 refinement pattern is used for 3-D elements and the type of element is conserved. Regarding element quality, it is decided not to use transition templates such as those developed by Staten (1996) to match zones with different levels of refinement because they produce low quality elements and they involve a computational effort to be handled. As a consequence: a) the adapted mesh has hanging nodes b) the 1-irregular node refinement constraint must be applied to ensure a gradual transition in the element size, c) mesh quality is degraded up to a certain value, d) there is no need to use compatibility rules for refinement / coarsening, and finally e) the solution must be constrained on the hanging nodes.

2.1 Extension of the 1-irregular node refinement constraint

The 1-irregular node constraint says that *no more than 1 hanging node should be shared amongst neighbouring elements through the common edge (2-D and 3-D) or face (3-D) to which the node belongs*. These kind of mesh refinemement was first proposed by Babuska and Rheinboldt (1978) for 2-D elements and here it is extended and used on 3-D meshes. To this end, face and edge entities are managed within the data structure of the mesh.

It is said that an edge or face is *active* if not all the elements that share it are refined, otherwise

it is *inactive*. The *parent* of an edge is the edge from which it derives by insertion of a midpoint. The *parent* of a face is the face from which it derives by joining the midpoints (and centre point on quadrilateral faces) with edges. As a consequence, the faces and edges that appear *inside* refined elements have no *parents.*

Refined elements are *always* deactivated. Finally, edges has two *childs,* faces has four and elements can have either 4 (2-D) or 8 (3-D) *childs*. These geometrical entities are organized in a hierarchical manner, so we say that an entity belongs to a low level of refinement if it has a higher hierarchy, and conversely.

The refinement function that applies the 1-irregular refinement constraint is recursive. Given a list of elements to be refined `eles2ref`, created at the error estimate stage, the function can be described as follows:

1. Find the edges that belong to `eles2ref`. Call these ones `edges2ref_tmp`.
2. Find in the set `edges2ref_tmp` those that have not been refined yet. Call them `edges2ref`.
3. Search the parents of the `edges2ref`. If some of them are still *active*, put them in the list `ParentEdges2ref`.
4. Find the elements to which these `ParentEdges2ref` belong and build a new list `eles2refNEW`.
5. If `eles2refNEW` is not empty, call the function with `eles2refNEW` as an argument. Else start to refine the elements in `eles2refNEW`.

For simplicity reasons the procedure has been described only for the edges, but a similar one is applied to the faces. In fact, `eles2refNEW` is a unique list containing the elements to be refined, obtained from applying the procedure to both the edges and faces of the mesh. `edge` and `face` are C++ classes/structures, and the edge/face class objects are stored in STL `vector<...>` containers. For example, the `edge` structure declaration is as follows

```
#ifndef EDGE_CLASS_H
#define EDGE_CLASS_H
#include<cstdlib>
#include<vector>
class edge{
public:
edge();
~edge();
int middle_node; //middle node in the edge
std::vector<int> nodes_parent; //nodes of the edge
int parent;  //parent's edge number
std::vector<int> child;  //child edges numbers
int flag;  //property flag
int state; //alive = 1 , inactive = 0
};#endif // EDGE_CLASS_H ///:~
```

and a vector of edges `v_edges` is declared as `vector<edge> v_edges`.

2.2 Strategy for unsteady problems

The strategy for applying the refinement algorithm to unsteady problems is developed considering that the adapted mesh has to be updated every few time steps. The updating frequency is constant and prescribed by the user for the whole run. The adapted mesh is obtained from applying refinement iterations to a *base mesh*, as it shown in Fig.1. For each one of these *adaptivity steps* (except for the first one) elements are marked to be refined, based on the error estimate chosen for the problem and the computed solution on the last adapted mesh. Elements to be refined are always selected at the maximum level of refinement. Since no higher level of refinement than that prescribed by the user is desired, a *parents search* algorithm is used. If elements which do not belong to the base mesh are selected, it is required to search for the parents of these elements in the data structure. Then, the elements selected for refinement are replaced by their parents and the process is recursively repeated until none of the elements in the list of elements to be refined has parents. In this way, *preliminary* lists of elements to build the next adaptivity step are stored. This process is represented by blue arrows in the diagram of Fig.1. Then, the next adaptivity step begins to be built. Since the elements numbering scheme of the adapted meshes at the same level of refinement changes from one adaptivity step to the next, the numbering of the elements to be refined in the preliminary list needs to be updated. Once this list is updated, the refinement process can go on. The updating and refinement processes are shown in the diagram as green and red arrows, correspondingly. Finally, the problem solver is restarted using the most refined mesh. Only for the first adaptivity step, the problem solver is run a few time steps on the base mesh. Then, elements are marked and refined as many times as it is required by the error estimates or until the maximum level of refinement is reached. The strategy does not consider the coarsening of the *base mesh*. An initial state to restart the flow computation is obtained as the linear interpolation of the solution belonging to the last adapted mesh, at the nodes of the base mesh. The boundary conditions are also updated. To this end, geometrical entities of the mesh have a property flag attached to them. This flag is associated with a special property, such as a fixation, a periodic or a slip constraint, dynamical boundary conditions for compressible flows developed by Storti et al. (2008), an element-wise property or maybe a combination of properties. Those properties are inherited from parents to childs at the refinement stage by inheritance of the flag (Fig.2), and updated lists of geometrical entities with that properties are built in a post-refinement stage within the adaptivity step.

2.3 Error indication

Error estimates are used to identify the zones of the mesh that need to be refined in order for the error of the approximate solution to be smaller than a prescribed tolerance. The choice of the right error indicator depends on multiple factors, such as the relative cost to compute the error estimate in regards to that of the rest of the adaptivity process, the affordable error estimate precision and the mathematical character of the equations. Although mathematically rigurous error estimate methods have been devised, most of them are usefull for elliptic problems. Heuristic gradient-based methods are frecuently used in hyperbolic problems since they are computationally unexpensive, they let the zones to be refined be properly identified and because of the lack of a more rigurous theory development for this kind of equations. Despite of these advantages, success in using these methods depends on the user's experience to choose the right variable or combination of flow variables that better suit the problem being solved. For non-viscous compressible flows the gradient of the density and/or pressure fields are often used.

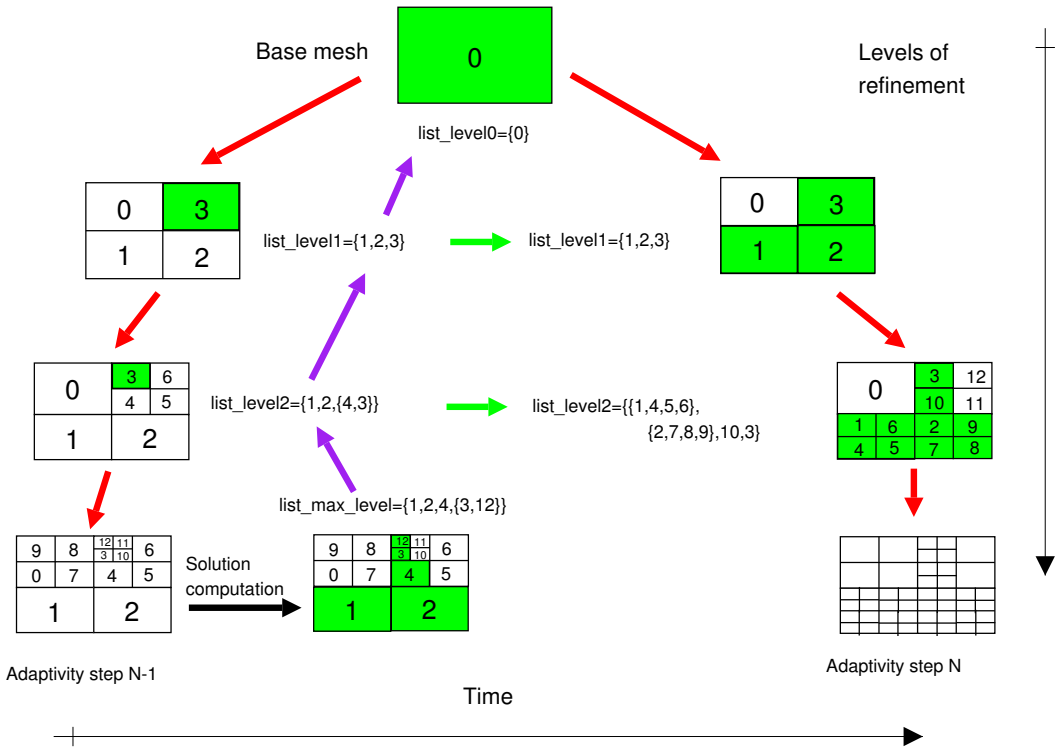


Figure 1: Adaptivity strategy scheme for unsteady problems.

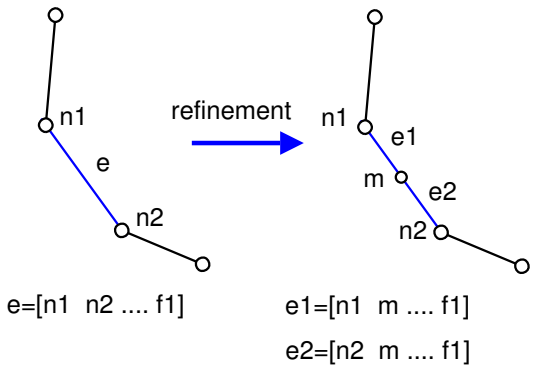


Figure 2: Property treatment for geometrical entities of the mesh - edge refinement.

Then, an element is marked to be refined when the element-wise computed gradient $|\nabla_e u|$ of a flow variable u times a measure of its size h_e (ie. the circumradius or the length of its longest edge) is greater than a prescribed tolerance value for the element error ε_e^*

$$|\nabla_e u| \cdot h_e \geq \varepsilon_e^* \tag{1}$$

3 THE TAYLOR-SEDOV SPHERICAL BLAST WAVE

The Taylor-Sedov shock solution describes what happens if a point-like explosion occurs in a uniform density gas. After some short time the explosion will have 'swept up' more than the mass of the material ejected in the explosion, and subsequently one expects to find a spherical shock wave propagating radially outward, with the shock front position a function of $R = R(E, \rho_0, t)$ where E is the energy of the explosion, $\rho_0 = 1.225 \text{kg/m}^3$ is the resting density of the ambient gas and t is the time since the explosion.

The shock wave comes to an end because the source of pressure also comes to an end, allowing a rarefaction wave to propagate forward, finally overtaking the shock. When this kind of phenomena takes place it is said that a blast wave happens. Blast waves are very common because the release of energy is of limited duration. They appear in processes such as solar flares release of energy (causing blast waves to form in the solar wind), interactions of jets with clouds, stellar explosions (supernova remnants develop a blast wave structure which they retain for much of their evolution), atmospheric nuclear explosion and depth charges.

Following Taylor's analysis of the problem, the spherical blast wave problem has a self-similar solution. By dimensional analysis and assuming that the solution has power law dependence on E, ρ_0 and t (otherwise it wouldn't be self-similar) it can be stated that

$$R \propto E^l \rho_0^m t^n \tag{2}$$

Then, equating powers of mass, length and time on both sides of Eq.(2) gives the power law indices $l = 1/5$, $m = -1/5$ and $n = 2/5$, so the expression for the spherical shock front position is now given by

$$R \propto E^{\frac{1}{5}} \rho_0^{\frac{-1}{5}} t^{\frac{2}{5}} \tag{3}$$

The constant of proportionality will depend on the equation of state of the gas. Although this can be computed by solving first the system of ODE's for the self-similar profiles of the flow variables, Hutchens (2000) gives the following equation for computing such constant

$$Const = \left[\frac{(N+3)(\gamma+1)}{4}\right]^{\frac{2}{(N+3)}} \left[\frac{(N+1)}{C_N}\right]^{\frac{1}{(N+3)}} \tag{4}$$

where $C_N = 4\pi$ and $N = 2$ for a spherical geometry and $\gamma = 1.4$ is the isentropic coefficient of the gas (air).

3.1 Problem setup

The Taylor-Sedov spherical blast wave problem is solved on an hemi-spherical domain using the adaptive strategy. The numerical solution is computed over an unstructured mesh of tetrahedra. The boundary conditions of the problem are

- Slip condition $\vec{V} \cdot \vec{n} = 0$ at the planar surface of the computational domain.

- Fixed pressure $p = p_0$ at the spherical surface of the computational domain, where $p_0 =$ 1atm is the ambient pressure.

For the last boundary condition to be correct, it is considered that the blast wave distance to the spherical boundary is sufficiently great.

The initial conditions of the problem are

- A small hemi-spherical region is at the center of the computational domain. This is the region of the explosion and its initial radius is $R_{ini} = 0.25$m.
- The state variables inside the radius of the explosion are $p_{blast} = 200\,p_0$, $\rho_{blast} = \rho_0$ and the fluid is at rest in the whole domain.

The three dimensional Euler equations are solved using the advection-diffusion module `adv_dif` of the PETSc-FEM multi-physics finite element solver, using 10 nodes on a cluster of PC. `adv_dif` is based on a Streamline Upwind Petrov-Galerkin stabilized finite element formulation together with a shock-capturing scheme.

The time step size is computed taking into account the CFL (Courant, Friedrichs and Lewy) condition and Courant number for the simulation is $Cou = 0.6$. The elements to be refined are selected computing the magnitude of the element-wise pressure gradient $\nabla p(T)$, choosing those whose value are

$$c_1 \cdot \max_{T_i \in \tau} (\nabla p(T_i)) \tag{5}$$

where constant $c_1 \simeq 0.1$ for the whole run.

Two levels of refinement are used and the frequency of adaption for the mesh is equal to 5 time steps. This means that every 5 time steps the mesh is adapted to the last computed solution. The final time for the simulation is $t_f \simeq 0.04$sec. The base mesh for the adaptive simulation has 140.000 elements and 26.400 nodes while the mesh for the final instants of the simulation has 1.350.000 elements and 285.000 nodes, approximately.

3.2 Results

The radial position of the shock front for the finite element solution is compared against that given by Eq.(3). It is seen that for time instant $t \simeq 0.02$sec, the computed position of the shock front using the finite element method agrees very well with that given by Eq.(3), that is

$$R_{cfd}(0.02) \simeq 5.6\text{m} \tag{6}$$

while

$$R_{TS}(0.02) = 5.4\text{m} \tag{7}$$

The exposed faces of the adapted mesh at time instant $t = 0.00885$sec. are shown in Fig.(3), while a cut of the mesh on a vertical plane of simmetry plus a Mach colourmap can be seen in Fig.(4). It can be clearly seen that the spherical shock front is captured by the mesh adaptivity strategy. Finally, the Mach number colourmap of Fig.(5) shows that a supersonic flow develops behind the travelling shock wave.

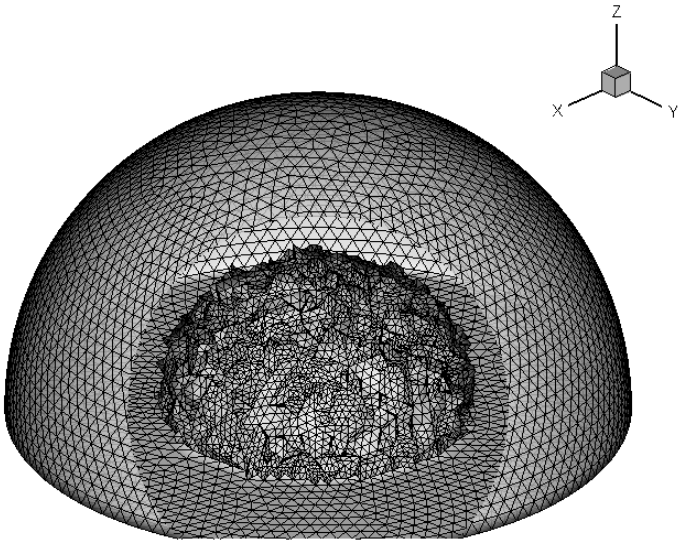


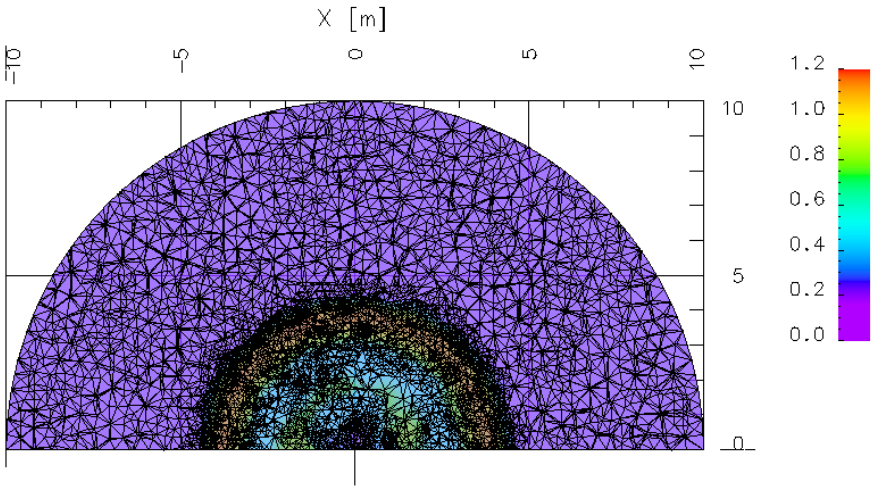
Figure 3: Exposed faces of the adapted mesh at $t = 0.008853$seconds.

4 CONCLUSIONS

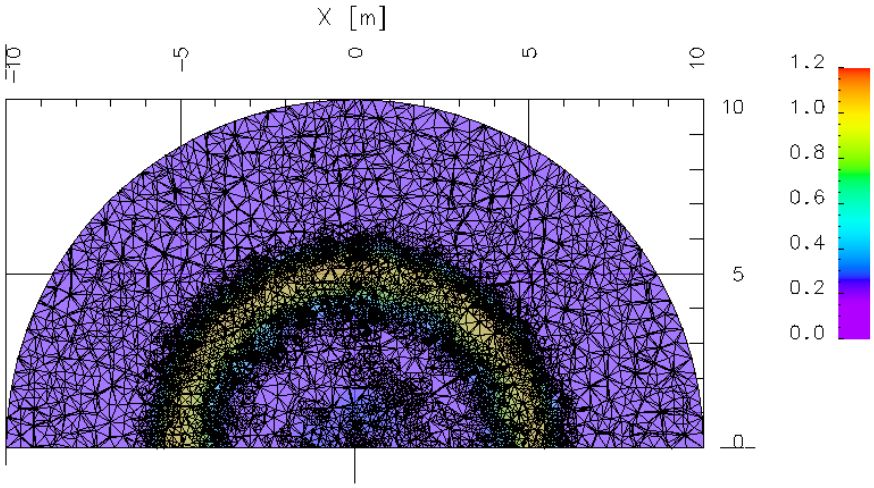
An h-adaptive strategy to solve unsteady compressible flow problems over unstructured finite element meshes has been presented. Here, it is tested for a tetrahedral mesh and it shows to be able to track the flow features through their displacement in the computational domain. The strategy, coupled with the advection-diffusion code of PETSc-FEM allows to accurately compute the shock wave position for the spherical blast wave problem. The capability of the adaptive code to handle different type of boundary conditions is also tested. Future work includes efficiency measurements of the code.

Acknowledgment

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grant PIP 5271/05), Universidad Nacional del Litoral (UNL, Argentina, grant CAI+D 2005-10-64) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT 12-14573/2003, PME 209/2003, PICT-1506/2006). Authors made extensive use of freely distributed software from the GNU Project and others, as Fedora GNU/Linux OS, MPI, PETSc, the GCC compiler collection, Octave, Open-DX, among many others. Also, many ideas from these packages have been inspirational to them.

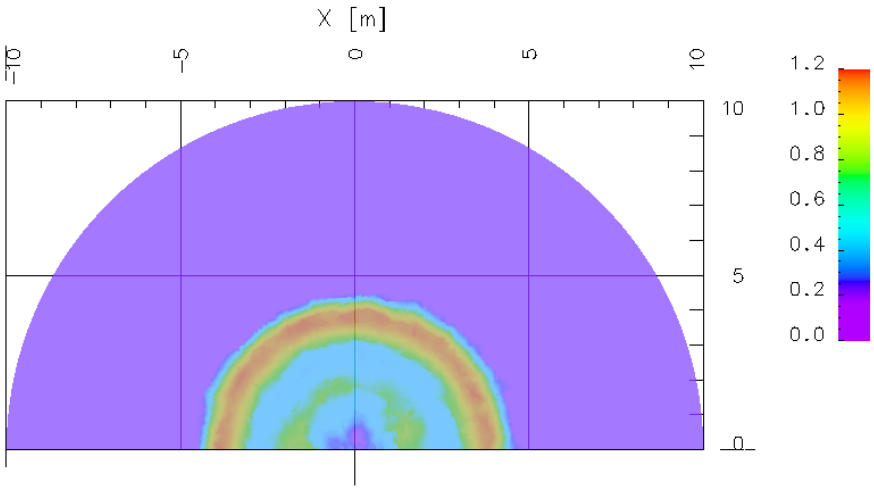


(a) t=0.008853sec

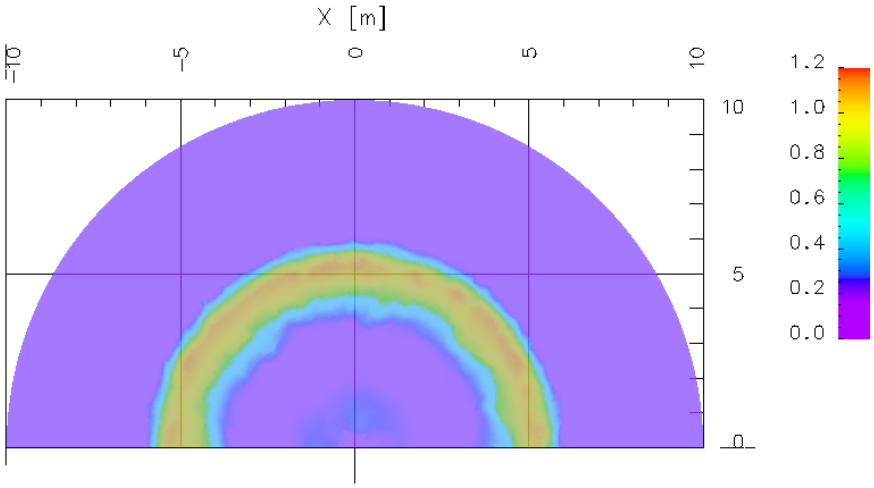


(b) t=0.0203sec

Figure 4: Adapted mesh plus Mach colormap for two time instants.



(a) t=0.008853sec



(b) t=0.0203sec

Figure 5: Mach colormap for two time instants.

REFERENCES

Babuska I. and Rheinboldt W. Error estimates for adaptive finite element computations. *SIAM J.Numer.Anal*, 15:736–754, 1978.

Hutchens G. Approximate near-field blast theory: A generalized approach. *Journal of Applied Physics*, 88(6):3654–3658, 2000.

Ríos Rodriguez G., López E., Nigro N., and Storti M. Refinamiento adaptativo homogéneo de mallas aplicable a problemas bi- y tridimensionales. 2005.

Ríos Rodriguez G., Storti M., and Nigro N. Refinamiento adaptativo en problemas no estacionarios. 2006.

Ríos Rodriguez G., Storti M., and Nigro N. An h-adaptive unstructured mesh refinement strategy for unsteady problems. *Latin American Applied Research*, 2008. In press.

Staten M. *Selective Refinement of Two and Three-Dimensional Finite Element Meshes.* Master's Thesis, Department of Civil Engineering, Brigham Young University, 1996.

Storti M., Nigro N., Paz R., Dalcin L., and Lopez E. *PETSc-FEM, A General Purpose, Parallel, Multi-Physics FEM Program*. CIMEC-CONICET-UNL, 1999-2007.

Storti M., Nigro N., Paz R., and Dalcín L. Dynamic boundary conditions in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 197:1219–1232, 2008.