AN H-ADAPTIVE SOLUTION OF THE SPHERICAL BLAST WAVE PROBLEM

G.A. Ríos Rodriguez, M.A. Storti, E.J. López and S.S. Sarraf

Centro Internacional de Métodos Computacionales en Ingeniería CIMEC
Universidad Nacional del Litoral, CONICET
Güemes 3450, 3000, Santa Fe, Argentina
gusadrr@yahoo.com.ar, mstorti@intec.unl.edu.ar

***Abstract*— Compressible flow problems usually exhibit features in the flow field that need a very refined mesh in order to achieve an acceptable accuracy in the numerical solution. Uniform refinement of the whole mesh quickly becomes prohibitive in three dimensions. In these situations, the use of an adaptive technique shows advantageous. Here, a h-refinement/coarsening strategy for unstructured finite element meshes is described and applied to solve the spherical blast wave problem driven by a point-like intense explosion. The resulting meshes are non-conformal. A 3-D refinement constraint is introduced to enforce a smooth size distribution among neighbor elements in the mesh and a gradient error indicator based on the flow variables is used to track discontinuities. It is shown that the adaption strategy allows to improve the accuracy and to reduce the spurious oscillations in the numerical solution, requiring just a small fraction of the overall computing time.**

***Keywords*— Mesh adaption, unstructured meshes, hanging nodes, spherical blast waves.**

I. INTRODUCTION

Transonic and supersonic flow problems are common candidates for being adaptively solved by the finite element method because discontinuities usually develop in a very thin region compared to some characteristic length of the problem domain. Adaption of the mesh allows to reduce the computational effort required to solve the fluid dynamic problem because of the introduction of smaller finite elements only where they are needed. A mesh enrichment procedure is used in this work to achieve this goal. The meshes to be adapted are unstructured and made up tetrahedra, although hexaedra can also be handled. Hanging nodes appear in the refined meshes because no transition elements are used to match the regions of the mesh with differente levels of refinement. A simple gradient error indicator based on the flow variables is used to drive the refinement of the elements in order to track the discontinuities through the flow field.

The adaptation algorithm is partially parallelized, i.e. the mesh refinement/unrefinement stage is sequentially performed while the solution of the fluid flow equations is computed in parallel on cluster of workstations. The three dimensional Euler equations are solved with the PETSc-FEM code (Storti et al., 1999-2009). This solver implements both the SUPG formulation introduced by Brooks and Hughes (1980, 1982) to stabilize the advection terms and shock capturing techniques for the treatment of strong shocks. The adaption of the mesh and the solution of the equations are coupled together for an efficient capturing of the continuously varying flow physics. The refinement algorithm is developed in C++ making an intensive use of the algorithms and containers provided by the STL(Sil, 1993-2006) library.

The spherical blast wave problem is solved. The results corresponding to the adapted solution are compared to those computed on a fixed mesh. Also, the set of ODEs derived from the Euler equations under the Taylor-Sedov self-similar assumptions are solved and used to validate the finite element solutions of the flow field behind the shock. Conclusions about the advantages of using the adaptive strategy are drawn based on the accuracy of the numerical solution and computational costs.

II. REFINEMENT SCHEMES AND REFINEMENT CONSTRAINTS

The adaption strategy introduced in this work is an extension of that presented in a series of previous works for steady and unsteady problems (Ríos Rodriguez et al., 2005, 2009).

It is well known that the condition number of the global stiffness matrix in the finite element method strongly depends on the geometrical quality of the elements in the mesh (Schewchuck, 2002). Also, the refinement procedures tend to diminish the quality of the mesh. To keep bounded this quality decrease, only 1:4 and 1:8 regular refinement patterns are applied to subdivide 2-D and 3-D elements, respectively. However,

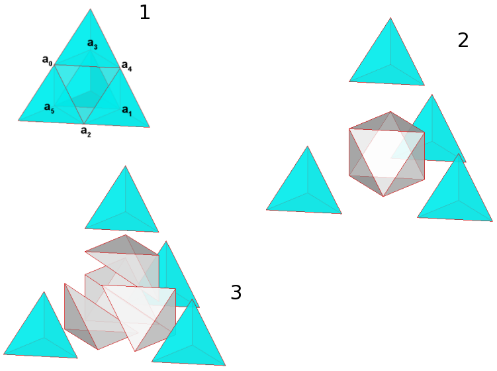


Figure 1: Tetrahedron refinement sequence.

since no regular 1:8 subdivision exists for tetrahedra a refinement scheme that shows a good tradeoff between the computational effort and the geometrical quality of the new elements in the mesh is chosen. In Ríos Rodriguez et al. (2009) it was shown through numerical experiments that refining the parent tetrahedron by joining the midpoints $(a_0, \ldots, a_5)$ of the edges with new ones and choosing the shortest diagonal of the inner octaedron (see sequence 1-2-3 in Fig.1) to get the four remaining tetrahedra is a good refinement strategy. In those experiments, the geometrical quality of the tetrahedra was measured with both geometric and algebraic quality metrics such as the dihedral angles and the mean ratio shape measure introduced by Liu and Joe (1994).

But besides high geometrical quality elements, a smooth transition in the size of the elements is also needed (Schewchuck, 2002). To this end, an extension to a refinement rule introduced by Babuska and Rheinboldt (1978) in the case of 2-D meshes is used. The rule of Babuska and Rheinboldt says that *no more than one hanging node should be shared among neighbour elements through the common edge to which the hanging node belongs.*

However, this constraint does not take into account special situations that appear in 3-D meshes, as it is shown in Fig.2 for tetrahedral elements. In three dimensions the neighborhood through edges and faces as well as the refinement of *orphan* edges [1] on triangular shared faces also have to be taken into account. Consider any two tetrahedra in a mesh that are neighbors through a common face. One of them has already been refined. Then, one of the elements that "touches" the shared face with an edge is marked to be refined (the green-coloured element in Fig.2.a). As a consecuence, an orphan edge (i.e.: the edge defined by the nodes $n1$ and $n2$) on the shared face also needs to be refined. It is seen in Fig.2.b) that if the two dimensional constraint were just considered, this situation would be accepted. But then a difference of more than one level of refinement would exist among nearby elements in the mesh. Besides, the state of the flow variables on that irregular node (marked as a red dot) would be constrained to the average of the state at nodes $n1$ and $n2$, which could also be constrained in the flow calculation (explained in the following section). Figure 2.c) shows that refining the element that shares the face such that at least one of its orphan edges has been refined copes with the problem. Obviously, this requires appropriate data structures to handle the downward and upward adjacencies among the mesh entities.

III. ADAPTION STRATEGY

The adaptive solution of the problem begins by solving the fluid flow equations on a conforming (no hanging nodes) unstructured mesh, hereafter called the *base* mesh. After a fixed number of time steps ($nsteps$), the regions of the base mesh that need to be refined are selected. In this work, the selection criterion is based on the magnitude of the element-wise computed gradient of the density in the following manner

$$c_1 \leq \frac{\parallel \nabla_i \rho \parallel \cdot h_i}{\max_i(\parallel \nabla_i \rho \parallel \cdot h_i)} \qquad (1)$$

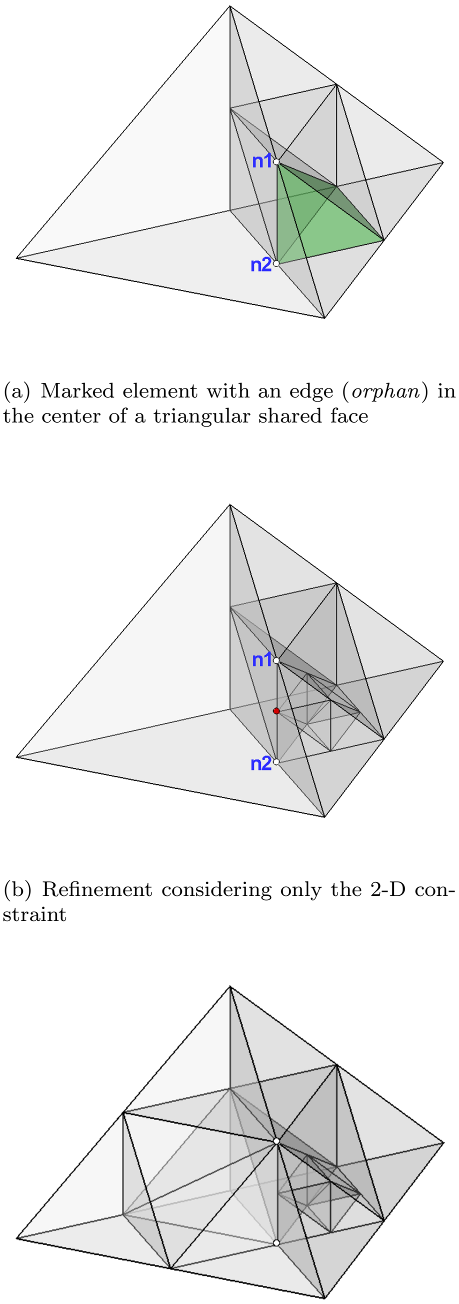
where c_1 is a constant set beforehand by the user, h_i is a measure for the size of the element and $\parallel \nabla_i \rho \parallel$ is the magnitude of the density gradient computed for that element.

Then a succession of nested nonconforming meshes is generated by applying the refinement patterns and refinement constraints already described in section II. until the maximum level of refinement allowed by the user is reached. It is worth to mention that although the adapted meshes introduce hanging nodes on the edges or faces of an element, constraining the solution at these nodes ensures its continuity among nearby elements through that edge or face. That is, the flow field variables on hanging nodes are computed as the average of the corresponding variables for the nodes that define the edges or faces (hexaedral elements) to which these hanging nodes belong.

As the base mesh in refined, the state computed by the solver is projected and the boundary conditions are updated as will be later explained in subsections Aand B. When the maximum level of refinement is reached, the flow computation is restarted using the projected solution as the initial state together with the proper boundary conditions.

After the solution is advanced $nsteps$ time steps, the selection criterion given by Eq.(1) is applied again and new elements are marked to be refined. The adaption strategy implicitly assumes that all the elements that are not selected for refinement should be coarsened. Also, since a maximum level of refinement is set to avoid high computational costs, only those elements that do not belong to the maximum level are finally included in the list of elements to be refined. On the

[1] Edges that appear in the refined mesh but are not obtained by the refinement of another edge are said to be *ophans* or with no parents.



(a) Marked element with an edge (*orphan*) in the center of a triangular shared face

(b) Refinement considering only the 2-D constraint

(c) Refinement of the neighbour through the shared triangular face (3-D constraint)

Figure 2: Refinement constraint for 3-D meshes - Orphan edge case.

other hand, an element is unrefined if and only if all its *brothers* (8 in 3-D and 4 in 2-D) are already marked to be unrefined. Then they are replaced by their parent element. This *search parent* procedure is recursively repeated on the data structure that stores the hierarchical relationship of the elements in the mesh until the base mesh level is reached. Finally, the list of elements to be refined in order to adapt the mesh is provided. It must be taken into account that although some elements should initially be unrefined, the subsequent application of the refinement constraint previously described may be does not allow it. Besides, it is emphasized that this strategy does not consider the coarsening of the base mesh.

Finally, a few words about the updating frequency (i.e. number of time steps *nsteps* until the mesh is updated again) are mentioned. The updating frequency of the mesh is set constant throughout the whole simulation. But, since the time step size is adjusted after every mesh adaption to satisfy the Courant-Friedrich-Lewy (CFL) condition for compressible flow problems (see Laney (1998)), the time simulated between two succesive adaptions of the mesh is not constant. This should prevent the shock waves from moving outside of the most refined regions of the mesh until they are adapted again.

On the other hand, a question arises about how to choose the appropriate frequency. In practice it is found that the adaption of the mesh, the boundary conditions and the projection stage of the state takes just a small fraction of the total computational time (approximately less than 5 per cent for unsteady three dimensional problems). This induce as to choose a high updating frequency for the mesh, maybe 5 or 10 time steps, for not compromising the overall performance of the adaptive solution procedure. If the time required by the adaption of the mesh were found to be a greater percentage of the overall computing one then a lower updating frequency should be chosen. In this case, however, a bigger cost would be transferred to the flow computation stage since the refined regions of the mesh would need to be "wider" to ensure the discontinuities will be kept inside these regions until the mesh is updated again. Choosing a higher frequency for adapting the mesh enables to use narrower refined regions around discontinuities so that the fluid flow problem is less expensive to solve.

A. Solution Projection

A new state for the recently adapted mesh has to be supplied as an initial condition to restart the flow computation. When the base mesh is adapted for the first time, a linear interpolation of the flow variables is used to compute the state on the new vertices. This is enough to guarantee conservation of the flow variables if linear finite element are used and if the new points are placed in the middle of the edges, barycenter of faces (quads) or elements (hexaedra).

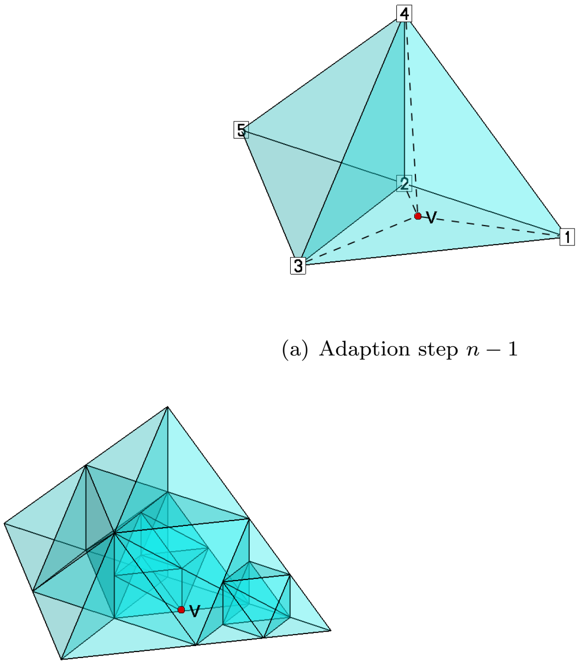
For the other adaption steps, the state is projected as follows : given a vertex V in the recently adapted mesh (adaption step n) it is required to find the element in the previously adapted mesh (adaption step $n-1$) that contains it. This is done through an approximate nearest neighbour search using the algorithms provided by the ANN library (Arya and Mount, 2006). The barycenter of the elements that belong to the $(n-1)$ adaption step are used as data points in ANN framework. The approximate search provides a list of k-nearest elements $l_V = \{e_1, e_2, \ldots, e_k\}$ to vertex V. These elements are candidates for containing V. Then, it is possible to find the element that truly contains V by computing the volume (3-D) or area (2-D) coordinates $N_j^{(e_i)}$, $j = 1, \ldots, ne$ for each one of the candidates at this vertex. Here ne is the number of nodes per element and j is the index of the local vertex in the element. If any of the volume coordinates $N_j^{(e_i)}$ is less than zero plus a certain tolerance, point V is outside of element e_i and the next element in l_V is considered. Only if all the volume coordinates for that element are greater than or equal to zero it can be stated that V is inside element e_i or over one of its faces or edges. In this case the traversal of l_V is interrupted and the state vector is computed as follows

$$\mathbf{U}_V^{(n)} = \sum_{j=1}^{Ne} N_j^{(e_i)}(\mathbf{x}_V) \cdot \mathbf{U}_j^{(n-1)} \tag{2}$$

where $\mathbf{U}_j^{(n-1)}$ is the state vector defined at local vertex j of the element e_i at adaption step $n-1$, $\mathbf{x}_V$ is the coordinates vector of vertex V and $\mathbf{U}_V^{(n)}$ is the state vector for vertex V at adaption step n. Dashed lines in Fig.3.a illustrate the volume coordinates at vertex V for adaption step $n-1$ while Fig.3.b shows the situation for the current adaption step.

B. Boudary Conditions and Mesh Entities Properties Managment

Boundary conditions and properties applied to mesh entities (namely edges, faces and elements) are handled in the adaption procedure by a property identifier associated to the entities of the base mesh. This identifier is inherited from a parent entity to its sons during the adaption procedure. The identifier is defined by the user and it can describe a set of features of different "nature" for an entity. For example: the identifier assigned to a face could mean that a slip boundary condition has to be enforced on that face and also that the face belongs to a curved surface which defines a particular section of the boundary. The user must supply a list of vertices which define the entities of the mesh that have a particular set of properties. Then, the identifier is only assigned to an entity provided certain conditions on the list of vertices are satisfied, e.g. if a set of properties is to be applied to faces, the condition might be that all the vertices of the faces should be in the list of vertices. After refinement, the entities with the same properties are identified in order to update the boundary conditions or the properties files supplied to the flow solver.



(a) Adaption step $n-1$

(b) Adaption step n

Figure 3: State projection - Vertex position detection.

IV. THE SPHERICAL BLAST WAVE PROBLEM

The blast wave problem was formerly and independently studied by Taylor (1946, 1950a,b) and Sedov (1959), and describes what happens if a very intense point-like explosion occurs in a uniform density gas. After a short lapse of time, it is expected to find a spherical shock wave travelling radially outward at supersonic speeds with a transonic flow behind it. This shock wave comes to an end because the source of pressure (.i.e. the release of energy) also comes to an end, in which case the rarefaction wave generated in the center of the explosion weakens the spherical shock until it becomes a pressure wave. When this kind of phenomena takes place it is said that a blast wave happens.

A. Self-similar Solutions

Taylor and Sedov analysis assumes a self-similar solution for the problem, which means that the solution profiles for the density ρ, velocity u and pressure p keep their shape in time and depend only on a single parameter ξ that is defined as the ratio of the radial coordinate r measured from the center of the explosion to the position of the spherical shock front R, so that

$0 \leq \xi \leq 1$. They formulate the following relationship between the physical variables and the self-similar profiles for the velocity $U(\xi)$, density $\Omega(\xi)$ and pressure $P(\xi)$

$$u = \dot{R}U(\xi), \quad \rho = \rho_0\Omega(\xi), \quad p = \rho_0\dot{R}^2P(\xi) \tag{3}$$

ρ_0 is the density of the ambient gas and $\dot{R}$ is the radial velocity of the shock wave. This solution holds as long as the mass swept up by the spherical shock front is much greater than the mass of the explosive material and as long as the shock wave can be considered strong. The equations for the self-similar solutions are derived from the Euler equations in radial coordinates

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\rho u\right) &= 0 \qquad (4)\\ \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + \frac{1}{\rho}\frac{\partial p}{\partial r} &= 0 \qquad (5)\\ \frac{\partial p}{\partial t} + u\frac{\partial p}{\partial r} - c_s^2\left(\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial r}\right) &= 0 \qquad (6)\end{aligned}$$

The latter can be reduced to a system of ordinary differential equations if it is further assumed that the density shows a power law dependence in space and time and the shock front position obeys to a power law in time. The ODE's are then numerically integrated with a fourth order Runge-Kutta method assuming the following boundary conditions inmediately behind the shock front (at $\xi = 1$)

$$U = \frac{2}{\gamma+1}, \quad \Omega = \frac{\gamma+1}{\gamma-1}, \quad P = \frac{2}{\gamma+1} \tag{7}$$

The self-similar profiles as a function of the similarity parameter are plotted in Fig.4. It is seen that the pressure in the center of the blast wave is almost half the maximum pressure inmediately behind the shock and it is fairly uniform within the blast wave. It can also be seen that most of the ambient gas mass processed by the shock wave is compressed within a thin spherical shell inmediately behind the shock which moves slightly slower than the shock itself ($u \simeq 0.83\dot{R}$ if $\gamma = 1.4$). Finally the velocity profile is almost linear in the blast wave.

A dimensional analysis shows that

$$R(t) \propto \left(\frac{E_x}{\rho_0}\right)^{1/5} t^{2/5} \tag{8}$$

where E_x is the energy released by the explosive and t is the time variable. The constant Q that allows to equate both sides of Eq.(8) can be computed by numerical integration of the total energy profile for a given time instant

$$E_x \quad = \quad \int_0^R \left(\frac{p}{\gamma-1} + \frac{\rho u^2}{2}\right) 4\pi r^2 dr \tag{9}$$

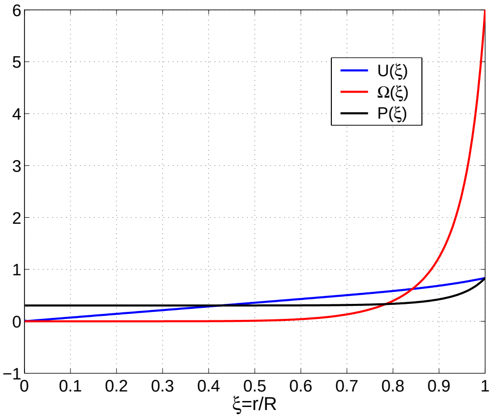


Figure 4: Self-similar profiles for the spherical blast wave problem ($\gamma = 1.4$).

Changing to variable ξ, substituting u, p and ρ from Eqs.(3) in Eq.(9), taking into account that $\dot{R} = \frac{2}{5}\frac{R}{t}$, then replacing E_x given by Eq.(9) into Eq.(8) and finally solving for Q it is found

$$Q \quad = \quad \left(\frac{16\pi}{25}\int_0^1 \left(\frac{P(\xi)}{\gamma-1} + \frac{\Omega(\xi)U(\xi)^2}{2}\right)\xi^2 d\xi\right)^{-1/5} \tag{10}$$

For $\gamma = 1.4$ the computed value for $Q \simeq 1.165$.

B. Finite Element Solutions

The problem is solved on a spherical domain of radius $R_{ext} = 5$m. Full 3-D Euler equations are solved although it there is spherical symmetry, since one the goals of the simulation, besides to analyze the improvements in the solution accuracy, is to evaluate how much of the overall computation time is needed to adapt the mesh and also to measure how the recursive refinement algorithm scales with the size of the problem, since it is expected that the number of elements to be refined increases with time because the area of the shock front also does it.

The initial conditions are given as follows: the ambient gas is at rest, at a constant pressure and density equal to $p_0 = 101325$Pa and $\rho_0 = 1.225$kg$/m^3$. It is assumed that the energy released by the explosive instantly raises the pressure to $p_{blast} = 10^5 \cdot p_0$ in a small spherical region of radius $R_{blast} \simeq 0.25$m. The initial explosion is not simulated in this work. The pressure fixation at the surface of the spherical domain is the only boundary condition prescribed. This condition is acceptable as long as the shock wave does not reach this boundary.

Tetrahedral elements are used to subdivide the problem domain, with elements of smaller size prescribed towards the center of the sphere. The resulting mesh has 421.000 tetrahedra and 76.500 vertices approximately. This mesh is used for both simulations, namely as the base mesh for the adaptive simulation and as the mesh for the non-adaptive one.

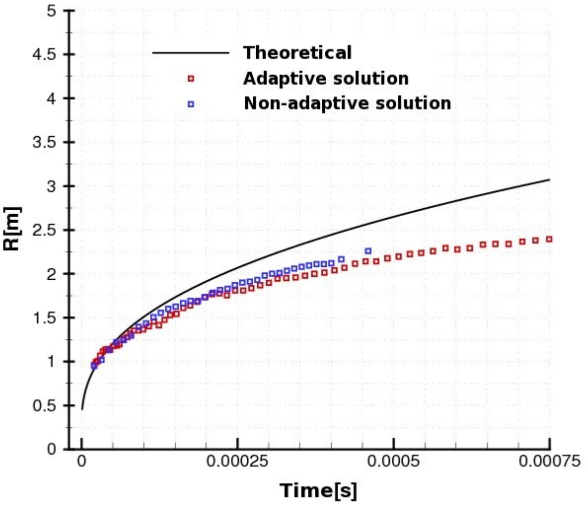


Figure 5: Shock wave position as a function of time.

The Euler equations are solved in parallel with 15 processors on a cluster of workstations and a Backward-Euler scheme is used for the time integration. It is recalled that the magnitude of the density gradient is chosen as an indicator for the adaptive simulation since the flow field generated by the blast wave is dominated by a strong shock and an expansion wave. A value of $c_1 \simeq 0.15$ in Eq.1 is used for the simulation. An updating frequency of 10 time steps is chosen and a maximum of 2 levels of refinement is prescribed. The final time for both simulations is equal to $t_f \simeq 0.001$s.

C. Simulation Results

In comparing the position of the shock front to that given by Eq.(8) it should be taken into account that the FEM solution profiles will just approximate those of the self-similar ones after a few time steps because the initial conditions for the flow variales are not those of the self-similar profiles of the theory. Bearing this in mind, Fig.(5) shows the shock wave position as a function of time for both the adapted and no-adapted simulations and the analytical one given by Eq.(8). It can be stated that although there is a good agreement for the first time instants, both simulations lug behind the analytical one.

Solutions along the radius for different time instants are shown in Figs.6 and 7, computed with the adapted (square symbols) and fixed (continuous line) meshes. Figure 6 shows that the pressure within the blast wave behave like that given by the self-similar solution, although higher values for the pressure at the shock front and behind it are reached if the mesh is adapted to the solution. The Mach number within the blast wave is depicted in Fig.7. It can be seen that it is in the transonic-subsonic regime, as it is stated by the theory. Both figures show that the entire flowfield is better resolved using the adaptive procedure because no spurious oscillations appear in the expansion region behind the shock wave and the latter is sharply defined. The shock wave travels roughly at an average speed of 3000m/s for the simulated final time t_f, i.e. it travels at mach number $M_S \simeq 9$ so the strong shock assumption of the self-similar solutions holds. Finally Fig.8 depicts a cut of the mesh on a plane of symmetry at $t = 0.645$ms while Fig.9 shows the pressure field and isolines for the same time instant. The mesh has approximately 2.34 million tetrahedra and 428000 vertices.

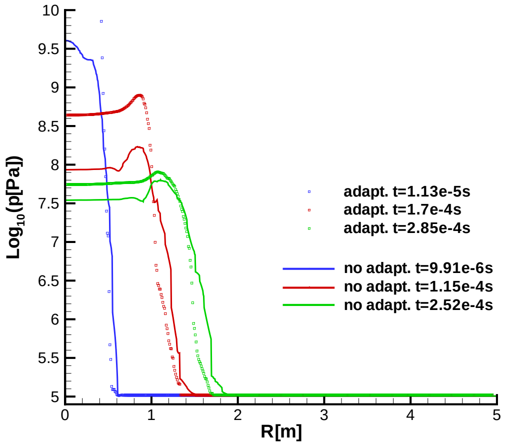


Figure 6: Time evolution of the pressure within the blast wave.

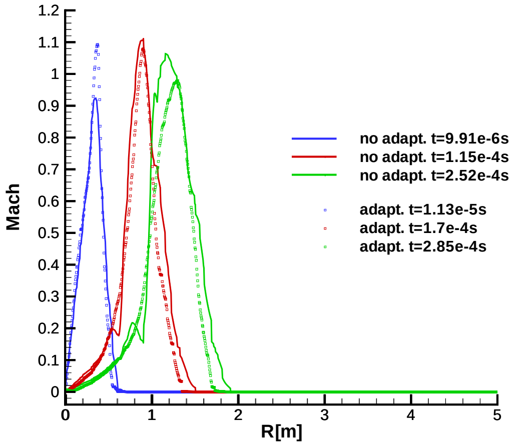


Figure 7: Time evolution of the Mach number within the blast wave.

D. Mesh Adaption Cost

To evaluate the code performance, clock time to perform the adaption of the mesh and to compute the equations solution is measured throughout the simulation. The adaption time t_{adapt} is defined as that required to realize all the necessary tasks to adapt the mesh, namely the error indication computation, the refinement of the elements, the boundary conditions

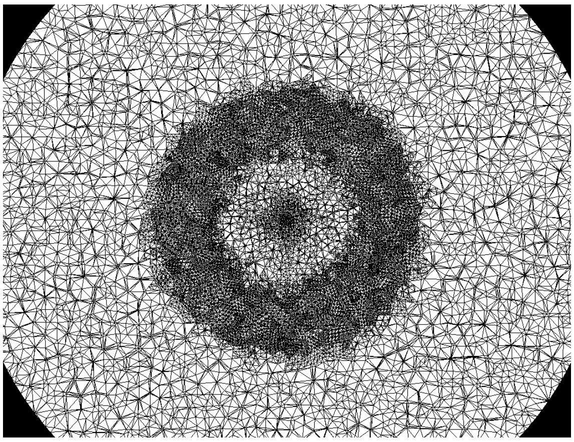


Figure 8: Adapted mesh on a plane of symmetry at time $t = 0.645$ms.

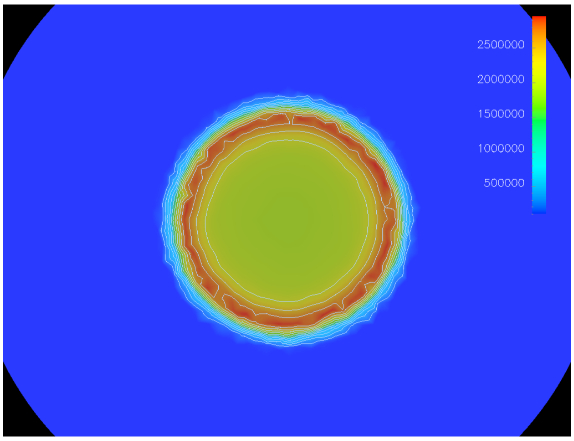


Figure 9: Pressure field on a plane of symmetry at time $t = 0.645$ms.

update, the state projection, the time step size update through the CFL condition and the writting to disk of all the files requiered by the flow solver. On the other hand, the solution time t_{sol} takes into account both the time requiered to advance the solution and the overhead incurred to restart the computation. The overall time is then defined as $t_{all} = t_{adapt} + t_{sol}$. Figure 10 shows that the ratio t_{adapt}/t_{all} keeps almost constant and equal to 0.04, which enables to state that, in this case, the adaption of the mesh takes just a small fraction of the solution time. Given that the biggest effort is involved in the solution of the flow equations, maybe a higher updating frequency for the mesh could be used.

On the other hand the refinement algorithm scalability is depicted in Fig.11, wherein the measured clock time taken by the recursive refinement algorithm is shown in the ordinates and the number of elements to be refined is shown in the abscissa. It is seen that an almost linear relationship is attained, at least for the range of refined elements $50.000 < Nele_{ref} < 240.000$. A linear fit is superimposed in the same figure for reference purposes.

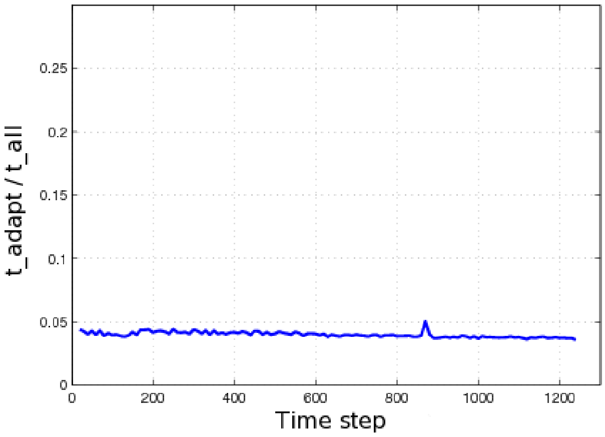


Figure 10: Relative cost for the mesh adaption.

V. CONCLUSIONS

The mesh adaption strategy is used to solve the spherical blast wave problem, improving the sharpness of the shock front and removing the spurious oscillations in the expansion that are present in the non-adapted mesh solution. The behaviour of the flow variables within the blast wave agrees rather well with the theoretical results from the Taylor-Sedov self-similar solution. It is noticed that although the shock front position is not accurately predicted, this cannot be ascribed to the adaption of the mesh since the non-adapted solution also shows a similar lack of precision.

The overhead introduced by the adaption of the mesh is just a small percentage of the time required to compute the flow, thus allowing to greatly reduce

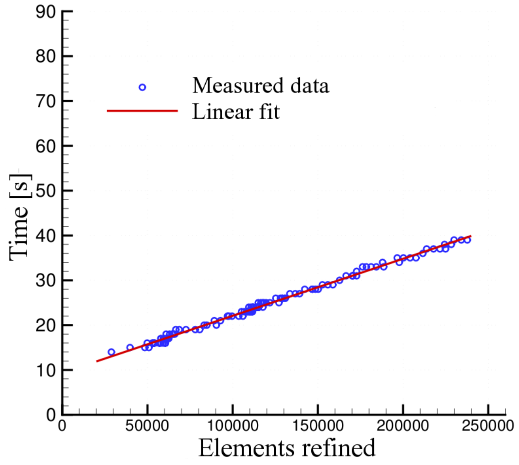


Figure 11: Refinement algorithm scalability.

the computational effort. If we were to solve the problem with a fixed mesh with a similar resolution (i.e. if the base mesh used for the simulations were homogeneously refined twice following the 1:8 pattern used by the adaption procedure), a fixed mesh made up of 26.9 million would be requiered. So it is concluded that true benefits are achieved because of adapting the mesh, namely an accuracy improving and a reduction in the computational effort.

Acknowledgment

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (`http://www.conicet.gov.ar`) (CONICET, Argentina, PIP 5271/05), Universidad Nacional del Litoral (`http://www.unl.edu.ar`) (UNL, Argentina, grants CAI+D-UNL 65/331,333,334) and Agencia Nacional de Promoción Científica y Tecnológica (`http://www.agencia.gov.ar`) (ANPCyT, Argentina, grants PICT-1506/2006, PICT-1141/2007). We made extensive use of *Free Software* (`http://www.gnu.org`) as GNU/Linux OS, MPI, PETSc, GCC/G++ compilers, Octave, Open-DX among many others. In addition, many ideas from these packages have been inspiring to us.

References

S. Arya and D.M. Mount. *ANN: Approximate Nearest Neighbors Library*. Department of Computer Science, University of Maryland, 2006.

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

A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov Galerkin methods for advection dominated flows. In *Third Internat. Conf. of Finite Element Methods in Fluid Flow*, Banff, Canada, 1980.

A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov Galerkin formulations for convective dominated flows with particular emphasis on the imcompressible Navier-Stokes equations. *Comput. Mehods Appl. Mech. Engrg.*, 32:199–259, 1982.

C. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.

A. Liu and B. Joe. On the shape of tetrahedra from bisection. *Mathematics of Computation*, 63(207):141–154, July 1994.

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

G.A. Ríos Rodriguez, M. Storti, and N.M. Nigro. An h-adaptive unstructured mesh refinement strategy for unsteady problems. *Latin American Applied Research*, 39:137–143, 2009.

J. R. Schewchuck. What is a good linear finite element? interpolation, conditioning, anisotropy and quality measures. URL http://www.cs.berkeley.edu/ jrs/papers/elemj.ps, 2002.

L.I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, New York, 1959.

Standard Template Library. Silicon Graphics, Inc., http://www.sgi.com/tech/stl, 1993-2006.

M. Storti, N. Nigro, R. Paz, L. Dalcín, L. Battaglia, E. López, and G.A. Ríos Rodriguez. *PETSc-FEM, A General Purpose, Parallel, Multi-Physics FEM Program*. CIMEC-CONICET-UNL, 1999-2009.

G.I. Taylor. The air wave surrounding an expanding sphere. In *Mathematical and Physical Sciences*, volume 186 of *Series A*, pages 273–292. Royal Society of London, Royal Society, September 1946.

G.I. Taylor. The Formation of a Blast Wave by a Very Intense Explosion. I. Theoretical Discussion. In *Mathematical and Physical Sciences*, volume 201 of *Series A*, pages 159–174. Royal Society of London, Royal Society, March 1950a.

G.I. Taylor. The Formation of a Blast Wave by a Very Intense Explosion. II. The Atomic Explosion of 1945. In *Mathematical and Physical Sciences*, volume 201 of *Series A*, pages 175–186. Royal Society of London, Royal Society, March 1950b.