NATIONAL TECHNICAL UNIVERSITY OF ATHENS
DEPARTMENT OF MECHANICAL ENGINEERING
SECTION OF FLUID MECHANICS

Development of mesh refinement methods at CFD codes for computational fluid mechanics problems

DIPLOMA THESIS

KONSTANTINOS LYRAS

SUPERVISOR:
SPYRIDON VOUTSINAS
ASSOCIATE PROFESSOR

Athens, February 2013

People to thank for:
I would like to thank my professor and supervisor of this thesis Mr. Spyros Voutsinas for giving me the opportunity to study this topic and for all the help he offered me all this time. I would also like to thank the PhD candidate Mr. George Papadakis for his help and interest. Without his assistance most of this project would be poor.
I would like to thank my parents who supported me all this time during my studies and my whole life. Anna Maria and Panagioths my sister and brother that I love them and my friends Sotiris and Makis.

Contents

1. Introduction

1.1 CFD in general.

Computational fluid dynamics known as CFD (Computational Fluid Dynamics) has target flow simulation with using computers. CFD has to do with solving the equations that describe the phenomenon that we want to solve numerical. The equations of interest are partial differential equations in time and space and a discretization scheme is needed, in order to solve them. This discretization is a proper division of space in smaller parts (computational cells and volumes). The set of the points of these parts are the computational grid.

Area of study of Computational Fluid Dynamics [1]

- CFD has to do with numerical simulation of fluids motion and the results can be used after the initial study of one specific problem, for further observations and conclusions. More time and capabilities comparing to experimental procedure.
- There is no human error affecting the results. The user can study one problem without affecting the problem analysis and flow parameters.
- The user has the ability to observe and analyze all the areas of the computational domain, having knowledge of the important quantities everywhere, something that in the experiment is not always either possible or easy.
- Ability to find the best solution. Using CFD we can work different case studies and decide which the best model is that will be tested at the experimental equipment. We can conduct only the tests needed at the wind tunnel using a specific and promising geometry (an airfoil for example).

CFD uses methods for solving non-linear partial differential equations of fluids, known as των Navier-Stokes equations, with respect to the geometry and boundary conditions. The result of the visualization of the flow is the universal knowledge of quantities of interest like pressure, temperature, speed, density, Mach number etc.

The solutions of Navier-Stokes equations although they are approximate solutions, they satisfy the conditions of the problem in a high level of accuracy. This level of accuracy has to do with the computational capabilities which have huge progress the last decades, and with choices, like the method of discretization, the method for solving the matrix systems of NxN equations. These are choices that can reduce the time of calculations and affect the accuracy and convergence. One of these decisions of the user must make in order to obtain a better solution of the flow equations is the generation of the computational grid, which is the main purpose of this thesis.

In order to represent what mesh changes we studied we will examine all the parts that a CFD problem is composed of and some of the basics of CFD analysis.

STEPS OF SOLVING CFD PROBLEM

The analysis of a CFD problem in aeronautics has to do with [2]:

– Topology
– Geometry
– Surfaces, internal and external (if they exist)
– Matching of topology with surfaces
– Grid generation.

Grid generation can be an iterative procedure. The grid has nodes and a certain way that they are connected. The nodes are organized in blocks. Both nodes and surface boundaries (if there exist more than one block) are allowed to move in the surfaces that are corresponding. In the unstructured grid, different methods of mesh generation are developed that will be represented below.

Grid Generation with source code

The user gives the coordinates at the boundaries of the geometry which must contain at least one internal boundary of any shape (the body) such as an airfoil, and one external one (the farfield). Using this first set of points, and defining some parameters for coercer of finer set of internal points (internal nodes) the code generates the grid. The node placing is iterative and computes the circumcircles for every pair of points in order to connect them. The final grid that is generated for the case of 2D unstructured mesh is a particular triangulation – Delaunay triangulation, which has some characteristics such as:

– There exist no points in the interior of the circle that fix 3 random points at the final grid.
– The angles of the triangles (cells) approximate the ones of the equilateral (maximization of the minimal angle of all the triangle cells). In that way thin computational cells that can cause problems while solving Euler equations are avoided (see ch.3, h-refinement). An example of this triangulation is given below (fig. 1):

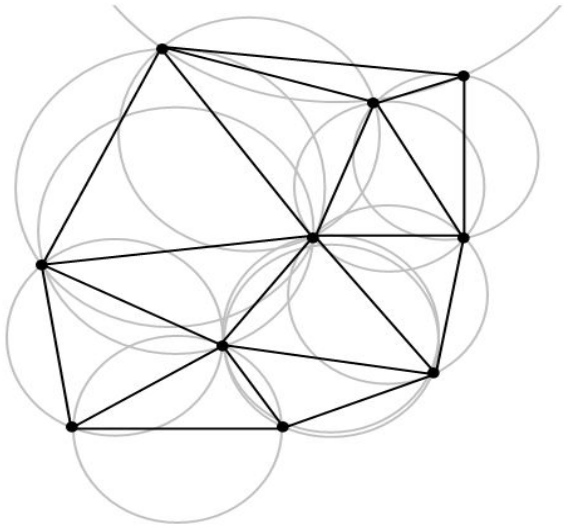


Fig.1: Delaunay triangulation of a set of points.

In case of an airfoil NACA0012, the domain between the internal and external boundary will be filled with a unique for those points Delaunay triangulation (interior and exterior nodes). The result can be like below (fig.2):

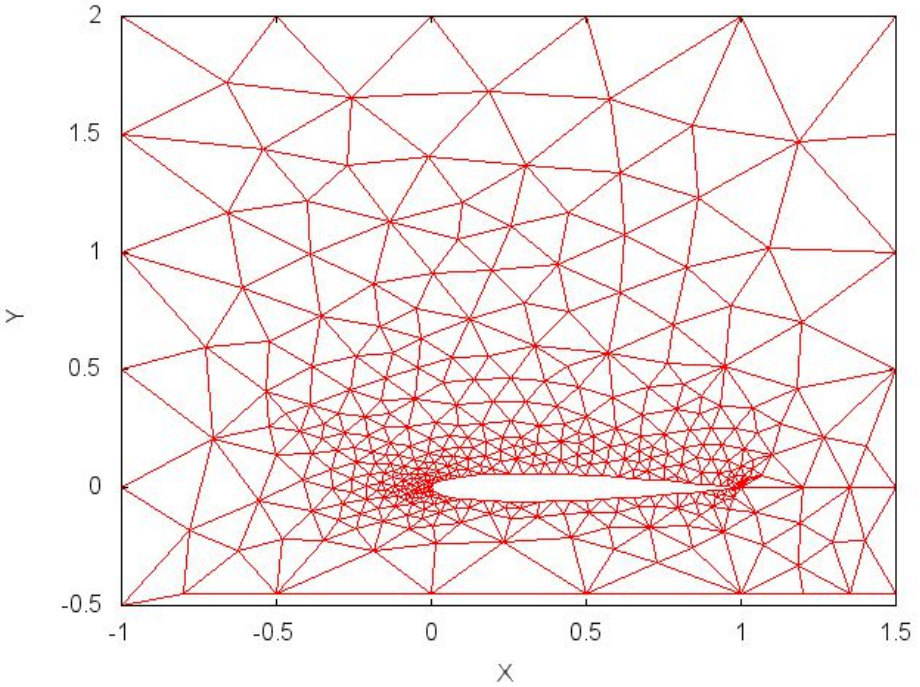


Fig.2: Unstructured grid generation from source code over Naca0012.

Grid Generation with commercial programs

The domain is divided into parts that can be filled separately. There is a clear differentiation of Topology-Grid and the options for setting up all the parameters that are equivalent with the grid generation, give the user the ability to construct the best possible grid for a given problem, choosing in between different models of geometry (3 geometry models: H,O,C-grid).

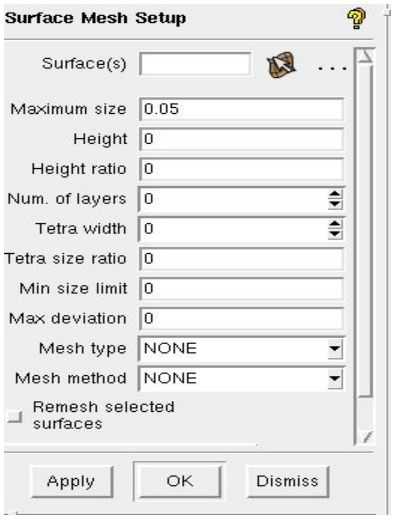


Fig.3: Ansys-Icem task bar defining geometry of grid.

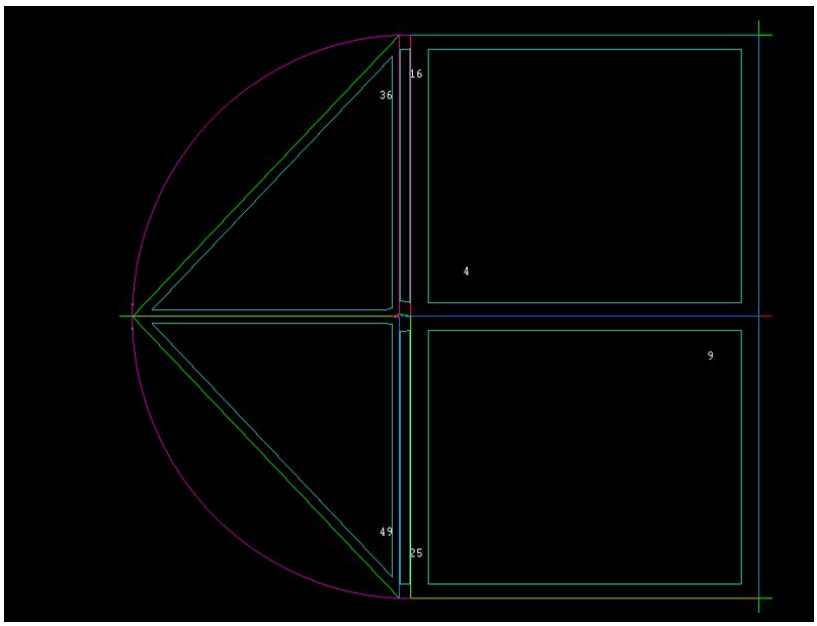


Fig.4: Defining topology over Naca0015

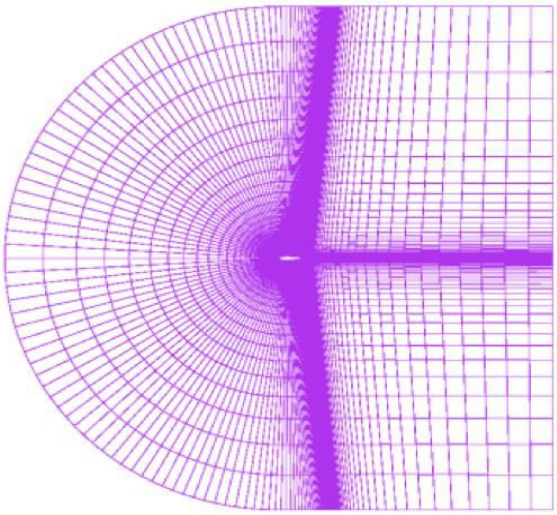


Fig.5: Grid over Naca0015 ,from Ansys-Icem.

Methodology for grid generation with many blocks (Multiblock)

Although it is a simple idea, grid generation in multiblocks can be a tough task. This has to do with stuff like numerical integration in areas of interface of the neighboring blocks, where different geometries exist, or stuff that have to do with defining which the neighboring blocks are. The key for the aerodynamic analysis and modeling is the geometry independent grid description.

- Defining the topology of grid structure of a multiblock, with the best possible structure, for the flow modeling and for all the component geometries.
- Defining nodal density of grid and grid point clustering για τις formed surfaces.
- Grid generation at every geometry.

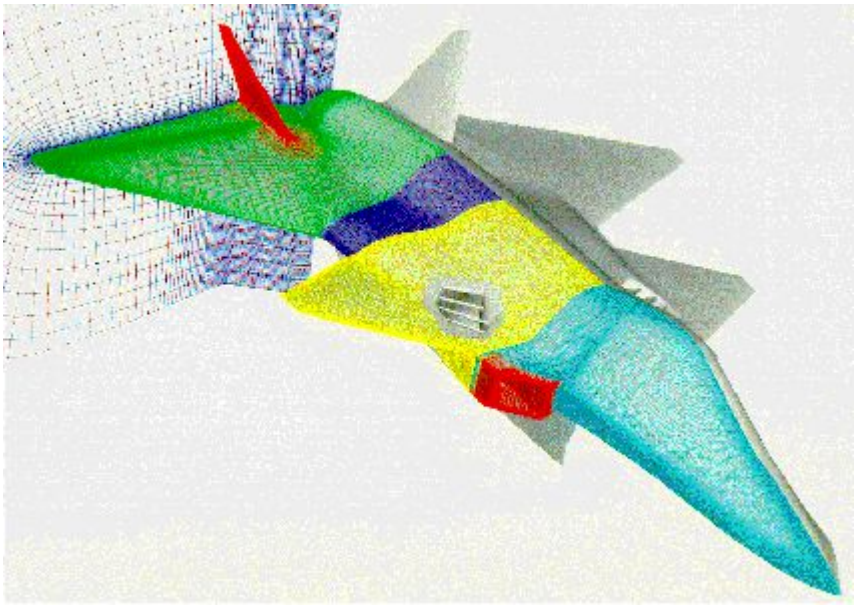


Fig.6: Block-structure jet

In conclusion, the steps for every Multiblock method are (fig.7):

Fig.7: General structure for solving a CFD problem.

1.2 Equations of flow motion

1.2.1 Navier-Stokes equations

After the grid construction and defining the boundaries (interior-exterior) of the domain, we are done with the mapping from physical to the computational domain. The next step that follows is to solve the equations that describe our problem. Navier-Stokes equations and the method of discretization used from our CFD solver are the following: integration in volume Ω with boundary $\partial\Omega$:

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{U}d\Omega + \oint_{\partial\Omega}(\overrightarrow{Fc}dS - \overrightarrow{Fu})ds = \int_{\Omega}\vec{Q}d\Omega \qquad (1)$$

where:

- $\vec{U}$, is the vector of the conservative variables :

$$\vec{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{pmatrix}$$

- $\vec{F_c}$, the vector of the convective fluxes:

$$\vec{F_c}=\begin{pmatrix}\rho V\\ \rho uV+n_x p\\ \rho vV+n_y p\\ \rho wV+n_z p\\ \rho(E+\frac{p}{\rho})V\end{pmatrix}$$

, and $V=\vec{u}\cdot\vec{n}$,

- $\vec{F_v}$, the vector of Viscous Fluxes:

$$\vec{F_v}=\begin{pmatrix}0\\ n_x\tau_x x+n_y\tau_x y+n_z\tau_x z\\ n_x\tau_y x+n_y\tau_y y+n_z\tau_y z\\ n_x\tau_z x+n_y\tau_z y+n_z\tau_z z\\ n_x\Theta_x+n_y\Theta_y+n_z\Theta_z\end{pmatrix}$$

Where:

$$\Theta_x=u\tau_{xx}+v\tau_{xy}+w\tau_{xz}+k\frac{\partial T}{\partial x}$$
$$\Theta_y=u\tau_{yx}+v\tau_{yy}+w\tau_{yz}+k\frac{\partial T}{\partial y}$$
$$\Theta_z=u\tau_{zx}+v\tau_{zy}+w\tau_{zz}+k\frac{\partial T}{\partial z}$$

The system of equations is completed with the perfect gas equation of state:

$$p=(\gamma-1)\rho[E-\frac{u^2+v^2+w^2}{2}]$$

1.2.2 Spatial discretization

Spatial discretization is about numerical approximation of the conservative and viscous fluxes and the integrals that occur. The methods of discretization are in general:

- Finite Differences methods
- Finite Volume methods
- Finite Element methods

Moreover, these methods can be categorized with respect to the type of grid which is used, in:

- Structured Grid methods

◆ Unstructured Grid methods

For the solution of the above equations, it has been chosen the method of finite volumes because of the following benefits:

◆ Spatial discretization is developed in the physical computational domain
◆ Finite Volumes methods can be applied in both structured and unstructured meshes

Finally, Finite Volumes methods can be:

◆ Cell-centered, where the variables of the equations are calculated at the center of the computational cells. In this case computational cells coincide with the cells of the grid.
◆ Cell-vertex, where the variables of the equations are calculated at the nodes of the grid. In this case computational cells are defined from some volume around the nodal points of the grid.

The cells are defined at the nodes of the grid and the flow variables are calculated at the center of the computational cells **(Finite Volume - Cell Centered Scheme**).
Assuming that the cell volume is time independent we get:

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{U}d\Omega = \Omega\frac{\partial\vec{U}}{\partial t}$$

Where:

$$\vec{U} = \frac{1}{\Omega}\int_{\Omega}(\vec{U}_{exact})\,d\Omega$$

The initial equation (1) becomes :

$$\frac{\partial\vec{U}}{\partial t} = -\frac{1}{\Omega}\left[\oint_{\partial\Omega}(\vec{F_c}-\vec{F_v})dS - \int_{\Omega}\vec{Q}\,d\Omega\right] \quad (2)$$

The surface integral at the above equation can be approximated with the sum of the fluxes at the surfaces (*faces*) of each cell.
Most of the times we assume that the flux in every surface remains constant and is calculated at its center.

Equation (2) for each cell **I**, is written:

$$\frac{d\vec{U_I}}{dt} = -\frac{1}{\Omega_I}\left[\sum_{m=1}^{N_f}\left(\left((\vec{F_c}-\vec{F_v})_m\Delta S_m\right)-(\vec{Q}\Omega)_I\right)\right] \quad (3)$$

Where N_f is the number of surfaces that each cell consists of and Δs_m is the surface of the edge "m". The quantity $R_I = \left[\sum_{m=1}^{N_f}\left(\left((\vec{F_c}-\vec{F_v})_m\Delta S_m\right)-(\vec{Q}\Omega)_I\right)\right.$ is called residual. The final expression of the discretized equation is written:

$$\frac{d\vec{U_I}}{dt} = -\frac{1}{\Omega_I}\vec{R_I}$$

1.2.3 Approximation of the variables at the boundaries of each cell

The discretization of the convective fluxes is about finding the fluxes at the boundaries of each cell. As mentioned previously, in *Cell-centered* schemes the values of conservative variables ($\rho, \rho\vec{U}, \rho E$) are considered known in the cell centers. In order to calculate the fluxes at the cell boundaries we need to approximate either the values of the conservative variables there, or directly the values of the fluxes.

The method that was developed for the solver of our laboratory, calculates the values of the primitive variables at the boundaries and afterwards the fluxes. This procedure is called reconstruction of the variables. In order to compute the values at the cell surfaces, we use left and right states. The interpolation of the variables on a certain surface of a cell is computed twice: One on the left and one on the right for a given surface and afterwards we calculate the flux through this surface. At this method we may make the assumption that the solution is Piecewise linearly distributed in the finite volume. The left and right states are computed with the following formulas:

$$\overrightarrow{W_L} = \overrightarrow{W_I} + \Psi_I\left(\nabla\overrightarrow{W_I} \cdot \overrightarrow{r_L}\right) \qquad (4)$$

$$\overrightarrow{W_R} = \overrightarrow{W_J} + \Psi_J(\nabla\overrightarrow{W_J} \cdot \overrightarrow{r_R}) \qquad (5)$$

Where $\vec{W}$ is the vector of primitive variables:

$$\vec{W} = \begin{pmatrix} \rho \\ u \\ v \\ w \\ E \end{pmatrix}$$

And $\overrightarrow{r_L}, \overrightarrow{r_R}$ are the distances of the cell centers from the centre of the common surface.

An important factor in the above formulas is $\nabla\overrightarrow{W_I}$ equations (4), (5). The calculation of the derivative is done with Green-Gauss approximation. The derivative is approximated by a surface integral:

$$\nabla\vec{W} \approx \frac{1}{\Omega}\int_{\partial\Omega} \vec{W}\vec{n}dS$$

In Cell-centred schemes, the above equation is discretized like this:

$$\nabla\overrightarrow{W_I} \approx \frac{1}{\Omega}\sum_{J=1}^{N_f}\left(\frac{1}{2}(\overrightarrow{W_I} + \overrightarrow{W_J})\overrightarrow{n_{IJ}}\Delta S_{IJ}\right)$$

The function Ψ is a function that allows the variables to take extremely large values, where discontinuities exist. These functions are called limiters. In our case we used the limiter of Venkatakrishnan.

1.2.4 Calculation of conservative fluxes

The calculation of conservative fluxes is accomplished with the Roe scheme.

At first, we compute the Roe variables:

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}$$
$$\tilde{u} = \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$
$$\tilde{v} = \frac{v_L\sqrt{\rho_L} + v_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$
$$\tilde{w} = \frac{w_L\sqrt{\rho_L} + w_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$
$$\tilde{H} = \frac{H_L\sqrt{\rho_L} + H_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}$$
$$\tilde{c} = \sqrt{(\gamma - 1)(\tilde{H} - \widetilde{q^2}/2)}$$
$$\tilde{q} = \widetilde{u^2} + \widetilde{v^2} + \widetilde{w^2}$$

The flux at the surface $I + \frac{1}{2}$ is defined as:

$$(\overrightarrow{F_c})_{I+\frac{1}{2}} = \frac{1}{2}\overrightarrow{F_c}(\overrightarrow{U_R}) + \overrightarrow{F_c}(\overrightarrow{U_L}) - |\, A_{Roe}\,|_{I+1/2}\, \overrightarrow{U_R} - \overrightarrow{U_L}$$

$$|\, A_{Roe}\,|_{I+1/2}\, (\overrightarrow{U_L} - \overrightarrow{U_R}) = |\, \Delta\overrightarrow{F_1}\,| + |\, \Delta\overrightarrow{F_{2,3,4}}\,| + |\, \Delta\overrightarrow{F_5}\,|$$

$$|\, \overrightarrow{\Delta F_1}\,| = |\, \tilde{V} - \tilde{c}\,|\, \left(\frac{\Delta p - \tilde{\rho}\tilde{c}\Delta v}{2\widetilde{c^2}}\right) \begin{pmatrix} 1 \\ \tilde{u} - \tilde{c}n_x \\ \tilde{v} - \tilde{c}n_y \\ \tilde{w} - \tilde{c}n_z \\ \tilde{H} - \tilde{c}\tilde{V} \end{pmatrix}$$

$$|\, \overrightarrow{\Delta F_{2,3,4}}\,| = |\, \tilde{V}\,| \left\{ \left(\Delta\rho - \frac{\Delta p}{\widetilde{c^2}}\right) \begin{pmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \widetilde{q_2}/2 \end{pmatrix} + \tilde{\rho} \begin{pmatrix} 0 \\ \Delta u - \Delta V n_x \\ \Delta v - \Delta V n_y \\ \Delta w - \Delta V n_z \\ \tilde{u}\Delta u - \tilde{v}\Delta v + \tilde{w}\Delta w - \tilde{V}\Delta V \end{pmatrix} \right\}$$

$$|\, \widetilde{F_5}\,| = |\, \tilde{V} + \tilde{c}\,|\, \left(\frac{\Delta p + \tilde{\rho}\tilde{c}\Delta V}{2\widetilde{c^2}}\right) \begin{pmatrix} 1 \\ \tilde{u} + \tilde{c}n_x \\ \tilde{v} + \tilde{c}n_y \\ \tilde{w} + \tilde{c}n_z \\ \tilde{H} + \tilde{c}\tilde{V} \end{pmatrix}$$

$$\Delta(*) = (*)_R - (*)_L$$

Variables at the left and right state are calculated with Piecewise Linear scheme. The above formulation may give solutions that are not correct from physical aspect. In order to fix this, the entropy correction of Harten is introduced ([4],[5]) for the eigenvalues of the system:

$$|\Lambda_c| = |\Lambda_c| \text{ if } |\Lambda_c| > \delta.$$
$$|\Lambda_c| = \frac{\Lambda_c^2 + \delta_2}{2\delta} \text{ if } |\Lambda_c| \leq \delta.$$

Where $\delta = \frac{1}{10} c$.

1.2.5 Boundary conditions

Boundary conditions are divided in the following categories:

- Solid Wall boundary conditions
- Farfield Inflow/Outflow boundary conditions
- Multiblock boundary conditions

For the formation of the boundary conditions we may use the so called dummy cells which increase the computational domain, in order to make it easier to compute all the variables at the boundaries.

Solid Wall boundary conditions

Solid Wall boundary conditions formulation differs in the case of viscous and inviscid flow. At the inviscid case, solid wall boundary conditions are described from the no-slip condition:

$V = \vec{u} \cdot \vec{n} = 0$ at the solid wall.
Consequently, the vector of the conservative fluxes at the solid wall (inviscid component) becomes:

$$(\vec{F_c})_w = \begin{pmatrix} 0 \\ n_x p_w \\ n_y p_w \\ n_z p_w \\ 0 \end{pmatrix}$$

where p_w is the pressure the solid wall.

For viscous flows, the velocity at the wall is zero (non slide boundary condition). Consequently, solid wall boundary condition is:

$$u = v = w = 0.$$

Farfield Inflow/Outflow boundary conditions

The physical information is transferred with respect with the sign of the eigenvalues of the conservative variables from or at the outside of the computational domain at the characteristics. The number of the boundary conditions should be equal with the number of the characteristics that come through the computational domain. The rest conditions are computed from the existing solution at the domain. The flow can be incoming or outcoming at the domain, so there are four types of

boundary conditions, with respect to the local Mach number:

- Supersonic Inflow
- Supersonic Outflow
- Subsonic Inflow
- Subsonic Outflow

Supersonic Inflow: In this case all the eigenvalues are positive and are coming inside the domain. The conservative variables at the boundary are computed only from the incoming flow.

$$\vec{U}_{boundary} = \vec{U}_{freestream}$$

Supersonic Outflow: In this case all the eigenvalues have the same sign and exit the domain. The conservative variables at the boundary are computed projecting the existing solution at the domain.

$$\vec{U}_{\mathrm{boundary}} = \vec{U}_{computational}$$

Subsonic Inflow: In this case, four characteristics are coming in the domain and one comes out of it. Only one characteristic variable is computed at the boundary from the interior of the domain.

$$p_b = \frac{1}{2}[p_a + p_d - \rho_0 c_0(n_x(u_a - u_d) + n_y(u_a - u_d) + n_z(u_a - u_d))]$$
$$\rho_b = \rho_a + (p_b - pa)/c_0^2$$
$$v_b = v_a - n_y(p_a - p_b)/(\rho_0 c_0)$$
$$w_b = w_a - n_z(p_a - p_b)/(\rho_0 c_0)$$

Where ρ_0, c_0 are computed at the interior of the domain.

Subsonic Outflow: In this case, four characteristics are coming out the domain and one comes in it. Four characteristic variables are computed at the boundary from the interior of the domain and one is formed at the exterior (in most of the times the pressure):

$$p_b = p_a$$
$$\rho_b = \rho_d + (p_b - p_d)/c_0^2$$
$$u_b = u_d + n_x(p_d - p_b)/(\rho_0 c_0)$$
$$v_b = v_d + n_y(p_d - p_b)/(\rho_0 c_0)$$
$$w_b = w_d + n_z(p_d - p_b)/(\rho_0 c_0)$$

At the above boundary conditions we may make the assumption of zero circulation, which fails for the case of a lifting body. Because of this, the farfield should be far away from the body. Farfield can be reduced significantly, if the outcoming flow be rotated according to the circulation. This correction is called vortex correction.

1.2.6 Time discretization

For the time discretization, the method of lines was applied. This means that time and space discretization is separated and gives a system of interlaced differential equations in time for each cell:

$$\frac{d(\Omega \bar{M} \vec{U})_i}{dt} = -\vec{R_i}$$

The above system of differential equations should be integrated in time either to give us the solution for steady problems or to recover the flow history for unsteady problems.

For non-moving grids the above equations is written as:

$$\frac{(\Omega M)_I}{\Delta t_I} = \frac{\beta}{1+\omega}\vec{R_I^{n+1}} - \frac{1-\beta}{1+\omega}\vec{R_I^n} + \frac{\omega}{1+\omega}\frac{(\Omega M)_i}{\Delta t_i}\Delta\vec{U_i^{n-1}} \qquad (6)$$

Where:

$$\Delta\vec{U_I^n} = \vec{U_I^{n+1}} - \vec{U_I^n}$$

Is the correction of the solution.

The matrix $\bar{M}$ is the mass matrix, which is considered unique for the problems that are studied. The time step can be a crucial factor for the stability of the problem. The maximum time step for each cell is computed with the following formula:

$$\Delta t_I = \sigma \frac{\Omega}{(\hat{\Lambda} + 4 \cdot \hat{\Lambda_v})_I}$$

Where $\hat{\Lambda_c}, \hat{\Lambda_v}$ are sums of conservative and viscous eigenvalues of the surfaces of each cell. They are given from the next formulas:

$$(\hat{\Lambda_c})_I = \sum_{J=1}^{N_f} (|\vec{u_{IJ}} \cdot \vec{n_{IJ}}| + c_{IJ})\Delta S_{IJ} \quad (2.30)$$

$$(\hat{\Lambda_v})_I = \frac{1}{\Omega_I}\sum_{J=1}^{N_f}[max(\frac{3}{3\rho_{ij}}, \frac{\gamma_{IJ}}{\rho_{IJ}})(\frac{\mu_L}{Pr_L} + \frac{\mu_T}{Pr_T})_{IJ}(\Delta S_{IJ})^2]$$

Implicit Time-Stepping

There are many schemes for time integration. An implicit scheme is for $\Omega = 0$ at equation (6) and for $\beta = 1$ is:

$$\frac{(\Omega M)_I}{(\Delta t)_I}\Delta\vec{U_I^n} = -\vec{R_I^{n+1}} - \vec{R_I^n} \qquad (7)$$

From equation (7), we need to know the values at time step $(n+1)$. In order to calculate the residual at this time step, it is linearized through the current time step:

$$\vec{R_I^{n+1}} \approx \vec{R}I_n + (\frac{\partial \vec{R}}{\partial \vec{U}})_I \Delta\vec{U^n}$$

Where the component $(\frac{\partial \vec{R}}{\partial \vec{U}})_I$ is the Jacobean of the fluxes.

If we apply the Jacobean of the fluxes in the equation (7) and put $\bar{M} = 1$ we get:

$$[\frac{(\Omega)_I}{\Delta t_I} + (\frac{\partial \vec{R}}{\partial \vec{U}})_I]\Delta\overrightarrow{U^n} = -\overrightarrow{R_I^n}$$

where the component $[\frac{(\Omega)_I}{\Delta t_I} + (\frac{\partial \vec{R}}{\partial \vec{U}})_I]$ is the implicit operator.

Implicit Operator

The Jacobean of the fluxes is:

$$\frac{\partial \vec{R}}{\partial \vec{U}} = \sum_{m=1}^{N_f} \frac{\partial(\vec{F_c})_m}{\partial \vec{U}} \Delta S_m - \sum_{m=1}^{N_f} \frac{\partial(\vec{F_v})_m}{\partial \vec{U}} \Delta S_m - \frac{\partial(\Omega \vec{Q})}{\partial \vec{U}}$$

The fluxes in the above equation are necessarily the same with the ones that we used at the space discretization because the Jacobean effects only the correction of the solution.

Using the Roe fluxes and considering the Jacobean of the viscous fluxes equal to zero, we get:

$$\frac{\partial \vec{R}}{\partial \vec{U}}\Delta\overrightarrow{U^n} = \sum_{m=1}^{N_f} \Delta S_m(\bar{A_c})_{L,m}\Delta\overrightarrow{W_{L,m}^n} + (\bar{A_c})_{R,m}\Delta\overrightarrow{W_{R,m}^n}$$
$$-(\frac{\partial}{\partial \overrightarrow{W_{L,m}}})[|\widetilde{A_{Roe}}|_m(\overrightarrow{W_{R,m}^n} - \overrightarrow{W_{L,m}^n})]\Delta\overrightarrow{W_{L,m}^n}$$
$$-(\frac{\partial}{\partial \overrightarrow{W_{R,m}}})[|\widetilde{A_{Roe}}|_m(\overrightarrow{W_{R,m}^n} - \overrightarrow{W_{L,m}^n})]\Delta\overrightarrow{W_{R,m}^n}$$

Assuming that locally Roe Jacobean are constant, we have:

$$\frac{\partial \vec{R}}{\partial \vec{U}}\Delta\overrightarrow{U^n} \approx \sum_{m=1}^{N_f} \Delta S_m[(\bar{A_c})_{L,m}\Delta\overrightarrow{W_{L,m}^n} + (\bar{A_c})_{R,m}\Delta\overrightarrow{W_{R,m}^n} - |\widetilde{A_{Roe}}|_m(\Delta\overrightarrow{W_{R,m}^n} - \Delta\overrightarrow{W_{L,m}^n})].$$

Unsteady Flows

For the calculation of unsteady flows we may apply the method of dual-time stepping. For β=1 and ω= ½ equation (6) becomes:

$$\frac{3\Omega_I^{n+1}\overrightarrow{U_I^{n+1}} - 4\Omega_I^n\overrightarrow{U_I^n} + \Omega_I^{n-1}\overrightarrow{U_I^{n-1}}}{2\Delta t} = -\overrightarrow{R_I^{n+1}}$$

where Δt is the physical time step. In order to solve the above system, the concept of Pseudo-Time. In this case, the unsteady problem in real time becomes a steady state problem in Pseudo-Time.

Assuming that:

$$\frac{\partial}{\partial t^i}(\Omega_I^{n+1}\overrightarrow{U^i}) = -\overrightarrow{R_I^i}(\overrightarrow{U^i}) \quad (2.39)$$

where i is the approximation of $\overrightarrow{U^{(n+1)}}$ and t^i is the variable of Pseudo-Time.
The unsteady residual is formulated as:

$$\overrightarrow{R_I^i}(\overrightarrow{U^i}) = \overrightarrow{R_I^i}(\overrightarrow{U^i}) + \frac{3}{2\Delta t}\Omega_I^{n+1}\overrightarrow{W_I^i} - \overrightarrow{Q_I^i}$$

All the components that remain constant during the Pseudo-Time steps are imported at the following component (source of component):

$$\overrightarrow{Q_I^i} = \frac{2}{\Delta t}\Omega_I^n\vec{U}I^n - \frac{1}{2\Delta t}\Omega_I^{n-1}\overrightarrow{U_I^{n-1}} \quad (2.41)$$

If the solution in Pseudo-Time converges, the residual $\overrightarrow{R_I^i}$ will be equal to zero and $\overrightarrow{U_I^i} = \overrightarrow{U_I^{n+1}}$. Therefore, the initial equation will be satisfied.
More specifically, in case of indirect methods of integrations in time, the above method is applied in the following formulation:

$$\frac{\partial}{\partial t^i}(\Omega_I^{n+1}\overrightarrow{U^i}) = -(\overrightarrow{R_I^i})^{l+1}$$

where (l+1) is the new Pseudo-Time step.
The next step is the linearization of the unsteady residual in Pseudo-Time:

$$(\overrightarrow{R^i})^{l+1} \approx (\overrightarrow{R^i})^l + \frac{\partial\overrightarrow{R^i}}{\partial\overrightarrow{U^i}}\Delta\overrightarrow{U^i}$$

with $\Delta\overrightarrow{U^i} = (\overrightarrow{U^i})^{l+1} - (\overrightarrow{U_i})^l$ where the Jacobean of the fluxes defined as:

$$\frac{\partial\overrightarrow{R^i}}{\partial\overrightarrow{U^i}} = \frac{\partial\vec{R}}{\partial\vec{U}} + \frac{3}{2\Delta t}\Omega^{n+1}$$

With this linearization at the initial equation, we get the indirect numerical scheme:

$$[(\frac{1}{\Delta t_I^i} + \frac{3}{2\Delta t})\Omega_I^{n+1} + \frac{\partial\vec{R}}{\partial\vec{U}}]\Delta\overrightarrow{U^i} = -(R_I^i)^l$$

where Δt^i is referred in Pseudo-Time (which differs for each cell) and Δt is referred in real time stepping.

2. Adaptive mesh refinement.

2.1 Types of refinement.

Mesh Adaptation during the solution iterations

Main purpose of the adaptation:
The main purpose of the mesh adaptation is finding a solution of Navier Stokes equations, in satisfying levels of accuracy, starting with one initial c*oarse mesh* and ending up with a final *fine mesh* that is refined at areas which is necessary to have more elements, with respect to fluidic phenomena. The initial mesh of the problem affects the solution that we will get after a number of iterations. This influence is mostly about the convergence and less about the accuracy.

Benefits of the mesh adaptation [3]
The new mesh, has an appropriate geometry, adapted to the problems flow conditions, which is more suitable than the initial's one.
Refining the mesh (final – fine mesh) will cause increment of the computational grid that will be used for the solution, but only at areas that this is demanded. The purpose of the mesh adaptation is the selective local enrichment of the mesh. It was observed that (chapter 4, Results) that applying this local refinement:

a) The convergence stays at the same or in acceptable levels.
b) Increasing the number of nodal points (increasing the number of cells) leads to *better flow simulation*, at any given geometry.
c) Simplicity of the refinement at unstructured grids.

Types of Mesh Adaption

1) Total reconstruction
2) Local enrichment (refinement-derefinement or h-refinement)
3) Moving nodes
4) Projection techniques

We will study type no2 at the above list, and apply an h-refinement with some flow criteria.
There are 3 types of refinement at CFD problems:

- *r – Refinement*: Nodes, elements remain the same. We relocate the cells at the area that we need to increase the resolution. We are not interested in this type of refinement.
- *h – Refinement:* The parent cell is divided, giving new children-cells and new nodes. This is the type of refinement that we will develop.
- *p– Refinement:* We take under consideration, the physical information at the parent cell, for example for a shock wave: Increasing p or ρ for boundary layers: $curl$ etc and the refinement is applied by changing the degree of the basis polynomial of the grid (at finite element methods).

The researcher must decide, according to the problem, the type of the refinement. There are no specific criteria that make a certain method more suitable than the others, for a certain problem. In this project we applied methods of ***h-Refinement*** using the appropriate fluidic criteria. Before we proceed to a detailed description of these methods, we will represent briefly the other two methods of

refinement:

► **r –Refinement**

As mentioned before, the number of nodes and cells remain the same, but they change order and connectivity, giving a different mesh, locally refined. This method can be applied for structured and unstructured grids, steady and unsteady, 2D or 3D fields [4].

Assumptions for r-Refinement

1) The mesh adaptation algorithm is mainly designed for the problems that concervation laws hold. This does not exclude the fact that it can be used for the numerical solution of any partial differential equation, for structured and unstructured grid.

2) The solution of the conservative variables and fluxes is fixed with respect to an inertia grid. The solution computed at time step $n+1$, (let as call it u^{n+1}) is applied for the grid at time step *n* (let us call this grid at n time, G_n) that might have the desirable resolution capable for an acceptable solution. The adaptation algorithm is called with input the solution u^{n+1} such that after the relocation, the new grid G_{n+1}, will give us better resolution. This node transportation (relocation) (the mapping$G_n \rightarrow G_{n+1}$) should not alter the solution but only affect the spatial resolution.

3) It does not change the grid connectivity.

Ways of r – Refinement (criteria)

- A function for the error (an error indicator), which is a quantity for measuring the lack of the resolution at the spatial domain.
- A method of reordering the nodes that will preserve the connectivity of the grid.
- An interpolation the solution at the new cells, after the relocation of the nodes.

At the picture below we can see a simple example of relocation at a three dimensional cell. The connectivity remains the same.

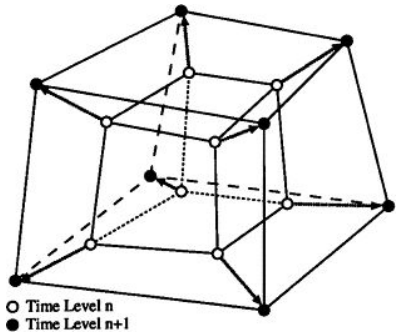


Fig.1: Relocation of grid nodes.

Relocation with weight functions

Since we want to increase the nodal density at the domain (if we define as nodal density the set of the nodes at the same space scale) we can use a weight function. At [4] for example for wave equation: $U_t + F_x = 0$, we firstly apply the mapping from $(x,t) \rightarrow (\xi,\tau)$ with the transformation $\tau = t, \xi = \xi(x,t)$.

For the 3D cell at system of coordinates (i,j,k), we may define the weight function as $\omega_{i,j,k} = \sum_\alpha \lambda_\alpha \sigma(u_\alpha)$ where λ_α is a coefficient that is used to change the influence of a given u_α at

the weight function, and σ is a mapping that has to do with the processing steps selected from those given.
When the user chooses weight function the nodal relocation can be applied considering the function $\omega_{i,j,k}$, like a mass that is related with the nodes of each cell. In that way the new nodes at (ξ,η) domain will be defined as:

$$\xi_{i,j,k}^{New} = \frac{\sum_{i,j,k=1}^{3} \xi_{i,j,k}\omega_{i,j,k}}{\sum_{i,j,k=1}^{3} \omega_{i,j,k}}$$

The exact same technique is applied for the other two coordinates. The result will be grids with higher resolution at the areas of interest. A nice example is given below:

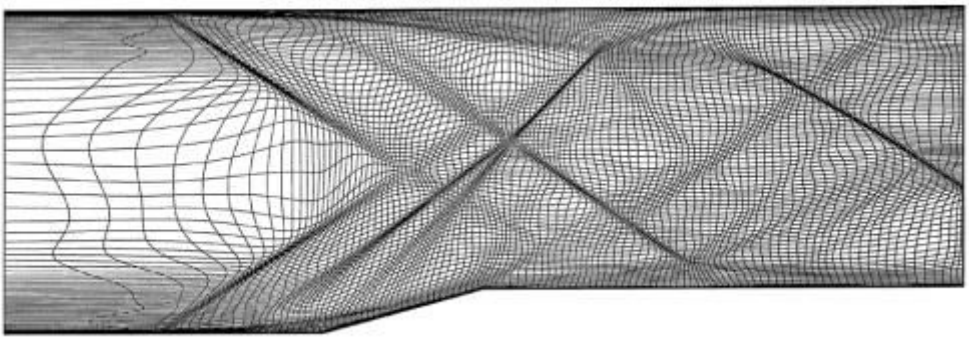


Fig.2: r-refinement increases resolution where is needed.

r-Refinement with Stretched Grid Method

Another technique with common logic, is the Stretched Grid Method (SGM), where the grid is manipulated as an elastic media where the refinements-deformations will give the desirable level of resolution.

Short description of SGM [5]

Assuming a grid with triangle cells, then for a random cell with boundary a polygonal continuing closed boundary, if we assume a successfully applied number of refinements-deformations at the nodal system, then we may define the potential energy of the nodal system, and assuming that this energy is proportional of the length of a *n*-dimensional vector with coefficient all the edges of the system, then:

Potential energy of a system with *n* edges: $E = D\sum_1^n R_i^{\,2}$, where D is a constant and R_i the length of the segment *i* of every edge. If X^s (smooth grid) is the vector of coordinates (of all the nodes) that is related with the non-deformable grid and X^d (distorted) then the following formula holds: $X^s = X^d + \Delta X$. The problem is then equivalent to find this vector. In order to solve X^s we apply the minimization of the square form of the energy. We demand that:
$\frac{\partial E}{\partial \Delta_{kl}} = 0$, (όπου k = the indicator of coordinates, l = the number of the internal nodes of the domain).

Finally we obtain the following system of equations:

$\begin{Bmatrix} [A][\Delta X_{\hat{x}}] = [B_{\hat{x}}] \\ [A][\Delta X_{\hat{y}}] = [B_{\hat{y}}] \end{Bmatrix}$ that will give the new nodes (with the condition that the boundaries sustained).

2.2 h-Refinement.

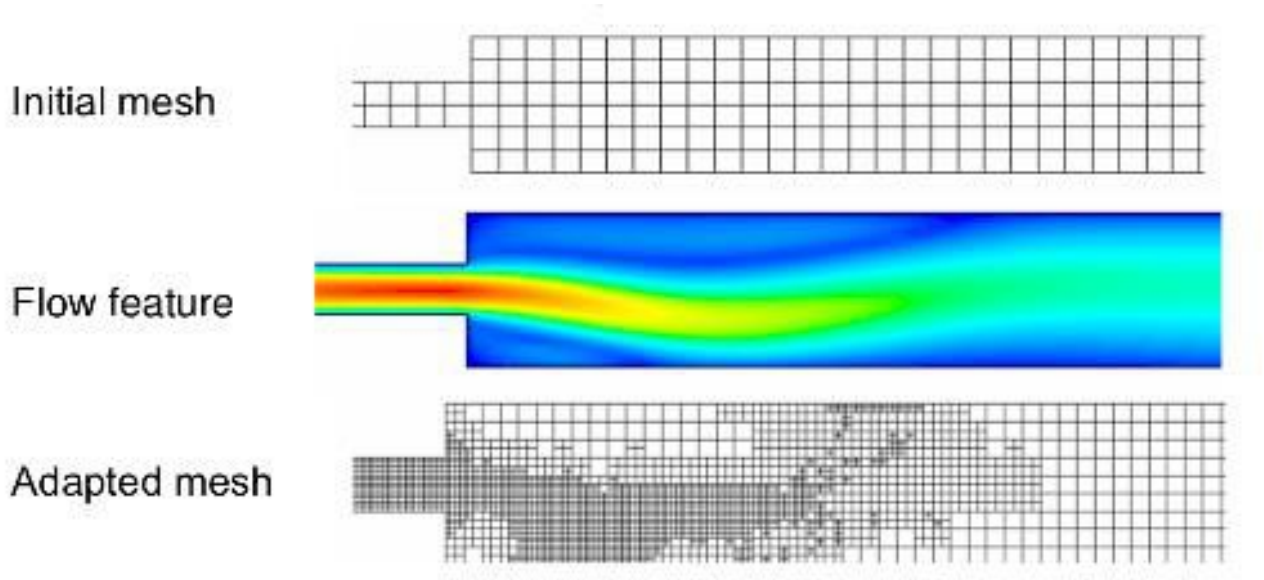


Fig.3: h-refinement (structured grid).

In this method, new nodes are constructed at the certain areas that are needed, increasing the number of the computational cells (fig 3.). This increment of cells is applied considering certain criteria that are essential for a successful mesh refinement. This method was developed at this thesis. More details about the criteria and the construction of the new grid will discussed at the next chapter. Some of the general principles are represented here.

Cell Enrichment of a mesh and the opposite (refinement/coarsening)

h-Mesh-Refinement is related with the spatial discretization of the computational domain. This means that mesh changes can be a road with two directions: increment of nodes or even vanishing them when is needed (fig.4). In these cases we get a better discretization of the domain, more suitable for our problem.

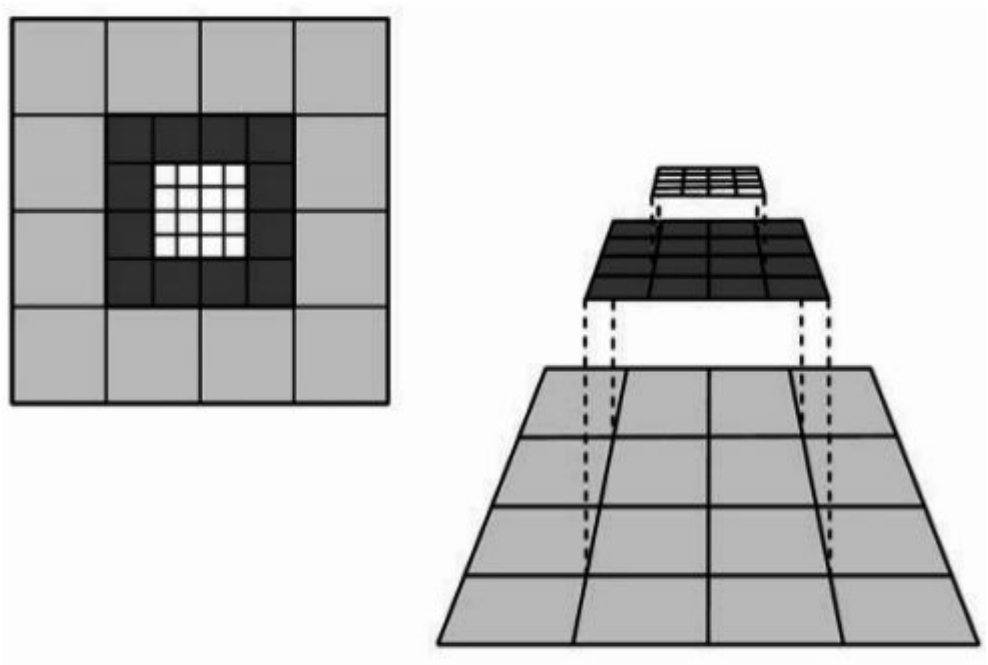


Fig.4: Refining-derefining structured grid.

Benefits of h-refinement [6],[7]

– Application to either Cartesian or unstructured grids.
– Easily applied algorithm for grid changes. Simplicity. Capability of making a recursive method.
– Especially for the unstructured grids, the subroutine that was developed, simple ideas for dividing parent cells were applied. Most of the original structure remains the same. No special treatment for the flow solver needs to be done. No special needs for the initial grid. Rules for the quality of the new grid must be implemented at the children-cells.
– Increment of the resolution, locally for a better simulation.

Aspects of h-refinement

Some issues for the mesh refinement are the following:

1) Finding the sub regions that the mesh may changes
2) Grid quality at the above areas and its effects at the solution.
3) The way that the refinement is developed.

For a successful refinement one may need:

1) One or more refinement (flow) criteria.
2) Geometric criteria for constructing the new grid.
3) An algorithm for the refinement.

For the flow criteria there is a numerous set of options with respect to the problem.

General Rules of h-refinement:

1) Flow criteria for choosing the areas of refinement. At the present thesis these were areas where shock waves occur.
2) The above criterion must be bounded, if the corresponding measurement changes while solving the governing equations.
3) It must be dimensionless.
4) The interpolation of h-refinement should be consevative, with no mass-energy or momentum.
5) Low levels numerical diffusion, because of the grid changes.
6) Fast algorithm for computer clustering.
7) Low CPU usage and memory storage.
8) Uniform grid. For instance if the initial mesh has a Delaunay triangulation, the new mesh must remain close to the Delaunay.
9) Error distribution at much more cells after every level of refinement.
10) No need for a priori knowledge of the flow.

The criteria that we applied will be discussed at the next chapter. For consistency reasons we will briefly represent the below comparisons:

h-Refinement vs Remeshing

With the word remeshing one may define the construction of a brand new grid after some time steps. This can be used for cases where higher resolution is required. The remeshing technique has the disadvantage that: At adaptive remeshing, the new mesh may not have the same nodes. That means that extra interpolations for the new nodal points will be demanded [8]. This issue shall increase the numerical diffusion. At the technique of h-refinement this is not an issue, because we apply such an interpolation only at areas where cells are divided of eliminated, reducing the computations which are demanded for the new mesh. This shows the simplicity of h-refinement methods versus the remeshing style. Moreover at h-refinement conservation laws hold by definition.

3. H-Refinement.

3.1 Over the Grid geometry.

Some basic information about the geometry of the grid will be represented here. This chapter contains some of the parameters that we should pay attention in order to obtain the best grid quality one can get for a given initial grid.

Given a set of points (2D ή 3D) even if it is a polygonal line, with internal holes or not, or even if it is multiple domain, or a curved domain, one can construct a grid that has this set of points nodal points. We may obtain structured, unstructured or hybrid grid (fig. 1). The *structured* is simply generated and the geometry info is clearly defined, whereas the *unstructured* gives the opportunity to change and refine selected sub regions of the domain, and is easily applied at complex geometries, giving a large number of cells at the solid bodies (an airfoil for instance) which means much more nodes at areas of interest which is our main target. Block-structured grid combines the benefits of the previous cases.

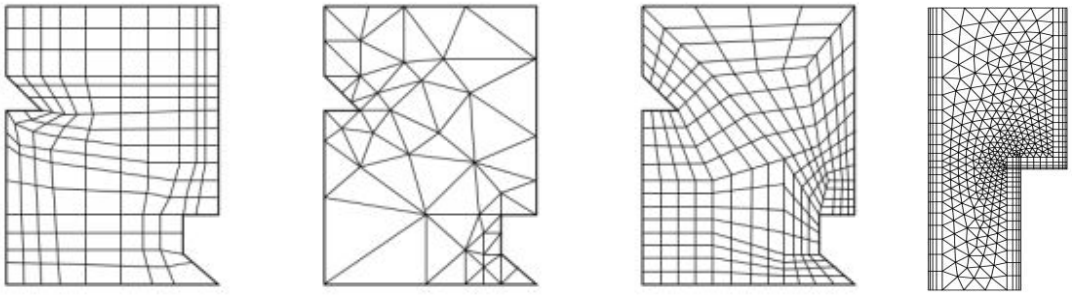


Fig.1: Grid samples a) structured, b) un-structured, c) block-structured, d) hybrid.

The effect of cell type at the numerical solution.

The shape of the cells that is used to solve the Navier-Stokes equations affects the solution obtained at every iteration. By geometry of each cell, one means edges and angles that have to do with the following:

Aspect-Ratio

We may define *aspect ratio* (AR) the fraction of the maximal and the minimum length among the cell edges (fig. 2) and *angle condition* the non-existence of angles lower of an given limit value (defined as an input).

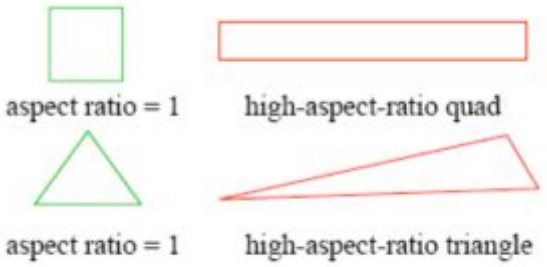


Fig 2: Aspect ratio (triangle-square cell).

In general cells with large AR do not lead at high quality grid and are not desirable. High AR means (in term of algebra) poorly conditioned matrices that may cause negative effects at the accuracy of the solution for a linear solver. Problems also may occur at the interpolation error for high AR even if we obtain acceptable solutions.

There some cases where elements with high AR may be desirable.

At [9] some of these cases are mentioned (cases like these where the solution of the partial differential equation is *anisotropic*). This means that the second order derivative differs significantly at some direction. In these cases it is possible that cells with high AR may be desirable.. In the same source the ideal AR in a grid is described as the square root of the fraction of

max and min eigenvalues of the Hessian matrix which includes the partial derivatives of the solutions.

An example of anisotropic problem is the solution of *Navier-Stokes equation* where the suitable AR can be quite large near the airfoil body. It is clear that whether we have high Re numbers or not this area can be sensitive with respect to the cell shape (see chapter 4, Results). This is obvious in the diagram of the pressure coefficient when working in unstructured meshes (triangle cells) compared to structured and hybrid cases (with square cells). As shown in the final chapter, quad elements have the advantage over triangle elements after consecutive refinements as far the error indicator in methods of finite volume. This is explained from the fact that the edges (boundary elements) in the boundary are placed parallel, fitted in a appropriate way.

– Another example of anisotropic solution is those related to shock fronts simulations in supersonic flows around airfoils. These cases sometimes can demand sub regions with large AR. So small error in the solution can be opposed with grids with triangle cells (at least in inviscid flows) affected with the level of refinements.

At [10] is developed of a method of Delaunay triangulation where locally generated cells with large AR are proven to be efficient in high Reynolds numbers in laminar flows.

Small angles

Another parameter in the element-shape category is the angles of the cell. The level of the solution error and convergence can be affected. At our methods of refinement, we applied criteria that keep the angles bounded and away from the useless cases of 0º and 180º, giving the user the ability to select the minimum and maximum of the angle as input. Typical values for our solver was 30˚ και 45˚ for cells that neighbor with others that pass the flow criterion. This is shown in chapter 4 where the refined areas have almost equal angles.

Refinement and grid structure.

Structured grid

The advantages of structured were mentioned before. The memory use is also one of them for the same number of cells, due to the simplicity of neighbor connectivity. This simplicity means that in order to find the cells close to a refined one, we can count the indexes with discrete steps saving computing time. For complex geometries can be also demanded unstructured cells because of the overwhelmed structured part after the refinement (can be affected from the flow solver). So for the test cases of viscous flows, a method of refining hybrid grids was developed. In these cases boundary layers are present and we demand to increase the local resolution.

Unstructured grid

Some of its advantages were mentioned before. Most of the refinement algorithms used unstructured meshes and has multiple applications in CFD methods. Meshes are most of the times triangles formed with *Delaunay triangulation* which for this thesis were developed in ANSYS-ICEM. ICEM like any other software package uses a code that fixes
Delaunay triangles. There many such algorithms such: Devine and Conquer, Sweep line,
incremental, gift wrapping etc. They all can be quite fast generating grids that can be the initial (coarse grid) with ($O(nlog(n))$) run time or even ($O(nlog(logn))$) for some cases, where n = the number of the points. A detailed report and comparison of the most popular of them can be find in [11].

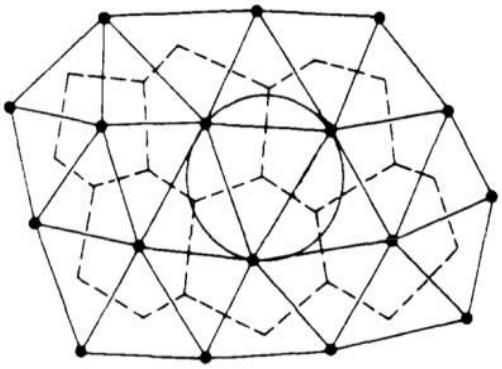


Fig. 3: Delaunay triangulation and Voronoi diagrams of a set of given points in 2D.

Adaptive Mesh Generation

Finally it is worth to mention the method called *adaptive mesh-generation* in which are used algorithms like the others above and codes starting from an initial set of points and generating a mesh applying iterative mesh changes that lead to a more suitable grid. There are direct and implicit methods for this purpose that do not apply the mesh changes while solving the governing differential equations. For example if one defines a *local stretching factor* for every single node of the initial grid (ξ, η-plane) with Delaunay triangulation, then we can have a criterion of the local nodal density. Define vector $\vec{s}$ [10] with norm: (where ξ, η are the discretization lines in the structured grid and defining the orientation with the unit vector),

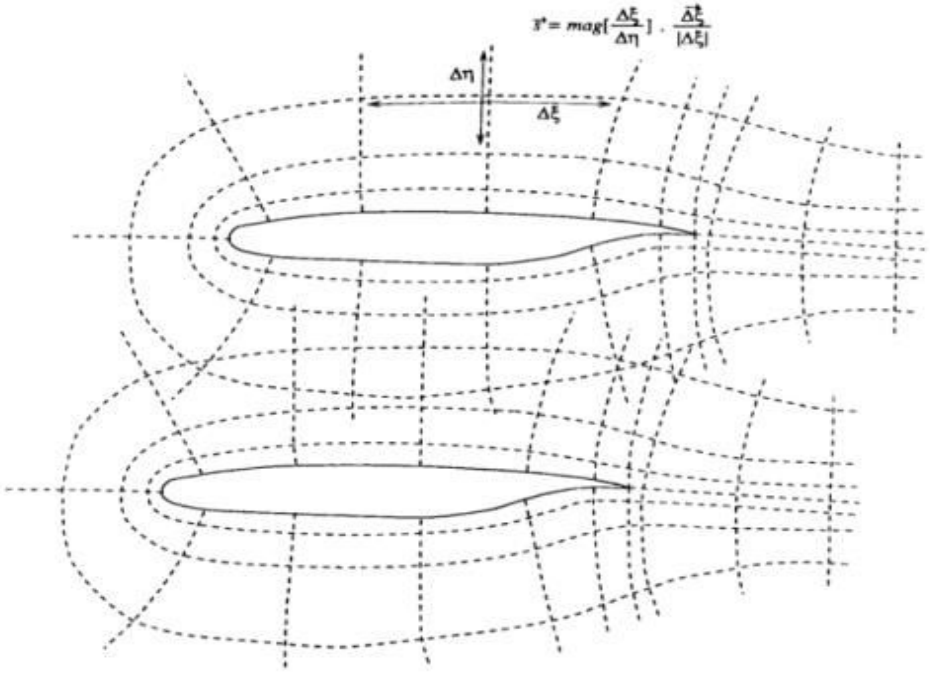


Fig. 4: Stretching factors (structured mesh 2D)

$$s = \max\left\{\frac{\Delta\xi}{\Delta\eta}, \frac{\Delta\eta}{\Delta\xi}\right\}.$$

When defining this vector/criterion for the local nodal distribution we can choose a subset of points at the structured part and using one triangulation algorithm (for instance Boyers devide-and-conquer) we can generate the unstructured part connecting the diagonals at the quad cells. This procedure may give triangle elements with large AR efficient for viscous flows (fig. 5).

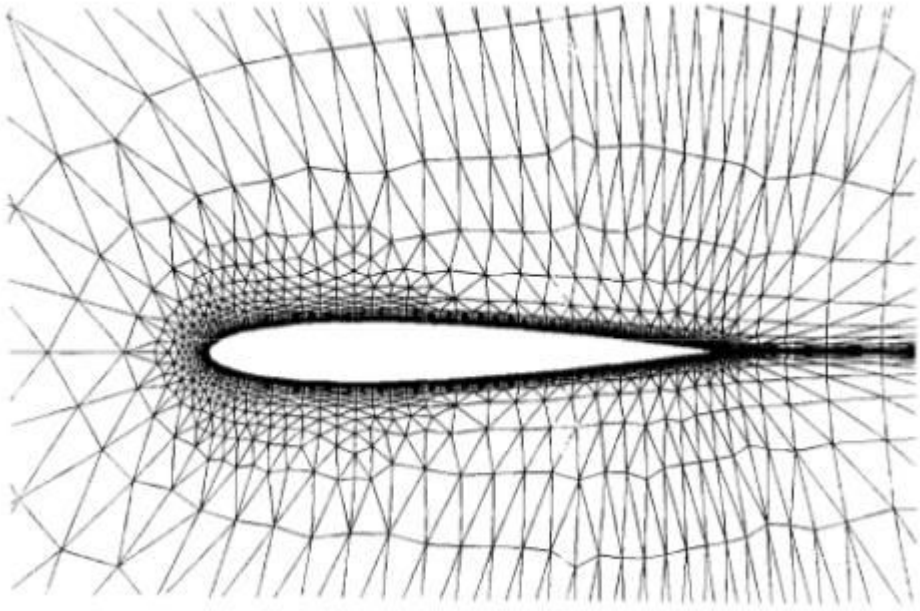


Fig.5: unstructured adaptive mesh generator NACA0012 (2D-viscous)

3.2 Steps of h-Refinement.

Mesh Refinement is a dynamic procedure, which means that the mesh that the solution is obtained changes after a number of time steps. The flow solver starts with an intial coarse grid and appling a local factor related with flow properties, the solver continues until the desirable convergence is accomplished, where the calculated mean error can be near zero, typically ~$1e-10$. The way the convergence rate behaves after every level of refinement is interesting because it is affected for the grid changes (more cells and increasing local resolution reveal the flow details for the shock wave over the airfoil, see chapter 4).

The psedocode of the computational plan of the refinement solver is:

$k = 0$
solve Navier Stokes on the initial triangulation T_0
if **mod**(Ntime,Number_of_Ntimes_to_be_refined) = 0 **then**
Estimate error for each cell
while maximum error on a cell is larger than the given input tolerance **do**

Mark the cells that have to been refined S_K
procedure refine the cells of S_K , get triangulation T_{K+1}
Interpolate the variables at T_{K+1}
Solve Navier Stokes on T_{K+1}
Estimate the error on each cell
$k = k + 1$
endwhile

The procedure that has to do with the cell division of the computational cells will be described detailed later.

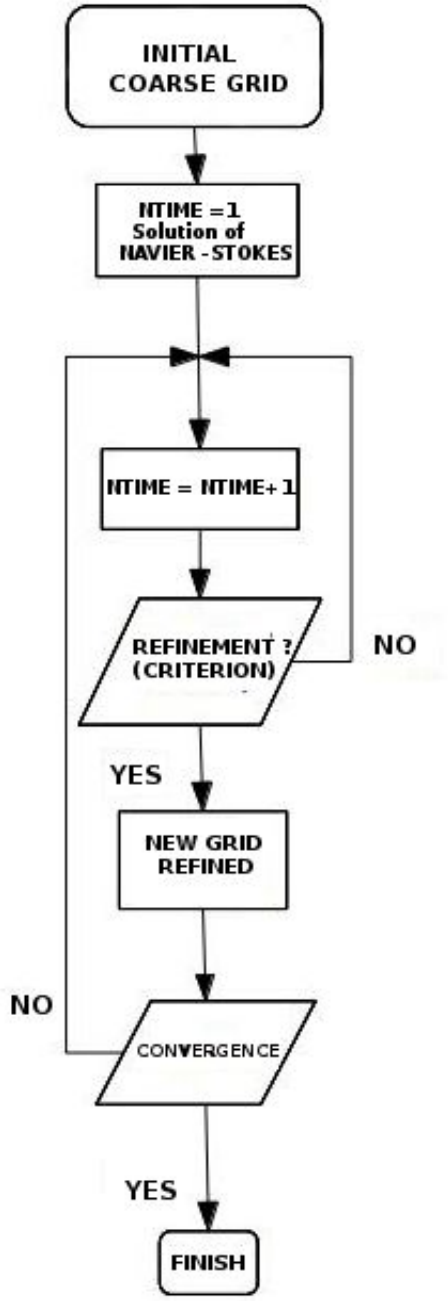


Fig. 6: Flow chart of refinement procedure.

3.3 Applying Mesh Refinement- Cell decomposition.

The discussion that follows is referred to:

- Two dimensional grids.
- Unstructured type.
- Triangle and Quad elements.
- In the flow solver of our laboratory (see references).

Triangle cells

In generall these are different ways of refinement for triangle cells e.g. where a parent cell can be divided. If we take the point where the diameters of each edge meet (barycenter) in 2D-plane we get a new node. As result 3 new cells occur with each one having as third vertex the new node. Another way of cell division is three new nodes over the edges, and consequently the 4 new triangles that occur, have at least one of their edges that belongs in the inside triangle. This case was applied in our refinement solver. If the initial triangulation is Delaunay, so it must be the new one.

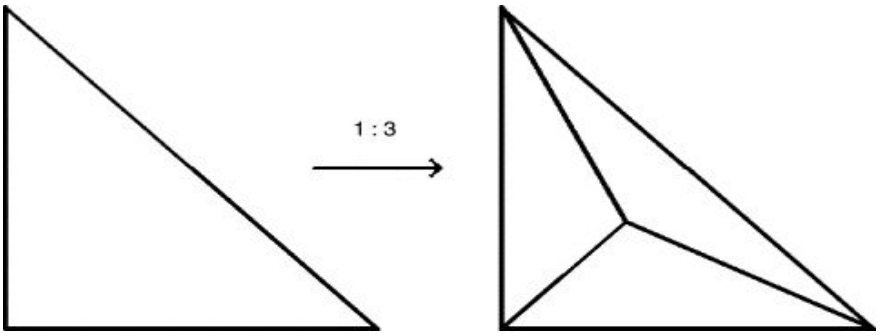


Fig.7.a) 1 new node, 3 new cells.

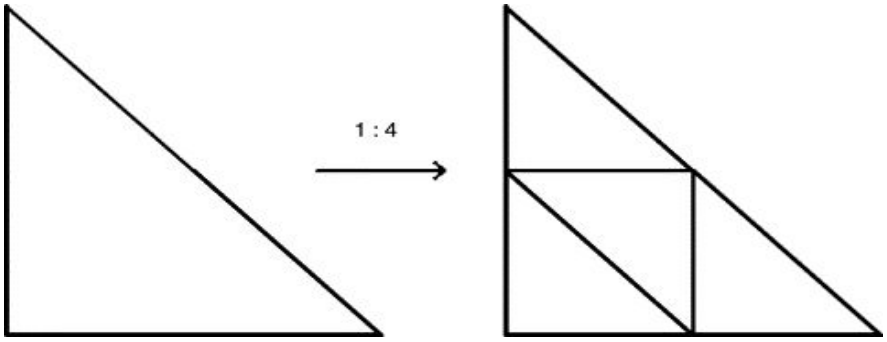


Fig.7.b) 3 new nodes, 4 new cells.

This although does not mean that the new mesh will be Delaunay after each refinement. The reason is that the new nodes that are made belong to an edge that is shared by two cells: Left-Right elements. If one of them fails the flow criterion of refinement (for example local Error estimation) it will not be divided in four new ones. But then we will have edges that one side only fluxes. These nodes, called hanging nodes and must be connected with nodes from the initial (this because the flow solver does not deal with hanging nodes).

Quad cells

In this particular type, there also different ways of division. As seen in the pictures below the cells with four edges can be:

α) Splited in two-dimensional space, to four new ones (fig.8.a) ,connecting the midpoints of the facing edges.

β) Splited at only two edges and so the parent cell is divided to two new ones (fig. 8.b).

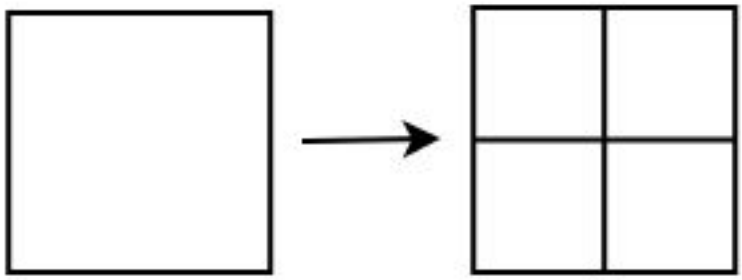


Fig. 8a): 1 parent-cell gives 4 new children-cells.

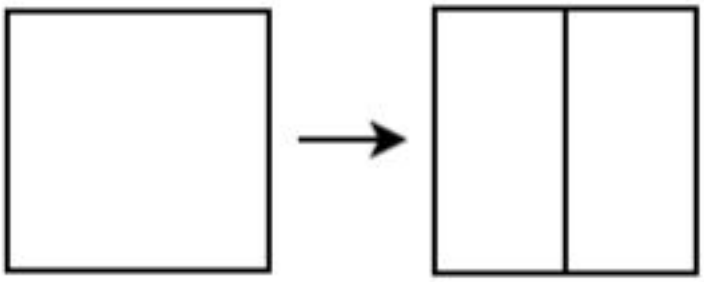


Fig. 8b): 1 parent-cell 2 new children-cells.

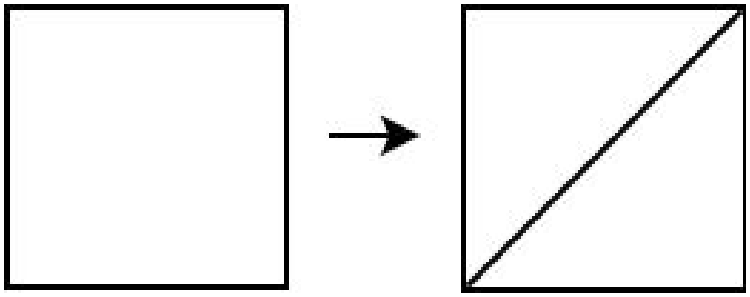


Fig.9):1 tet-parent can give 2 tri-children .This case can give another kind of refinement (is not applied at our current refinement solver)

For the purposes of this thesis, the second case was selected: One parent cell splited to two new equally sized ones. The obvious discomfort of this method is that the new cells may have large aspect ratio and also give two and not four (assuming that in this cases of four new ones, we would get more cells and higher resolution).

There is a basic disadvantage of our choice : limited Refinement, only at certain structured parts. This means that practically the refinement retains the local character as we can see at the following picture (fig.10).

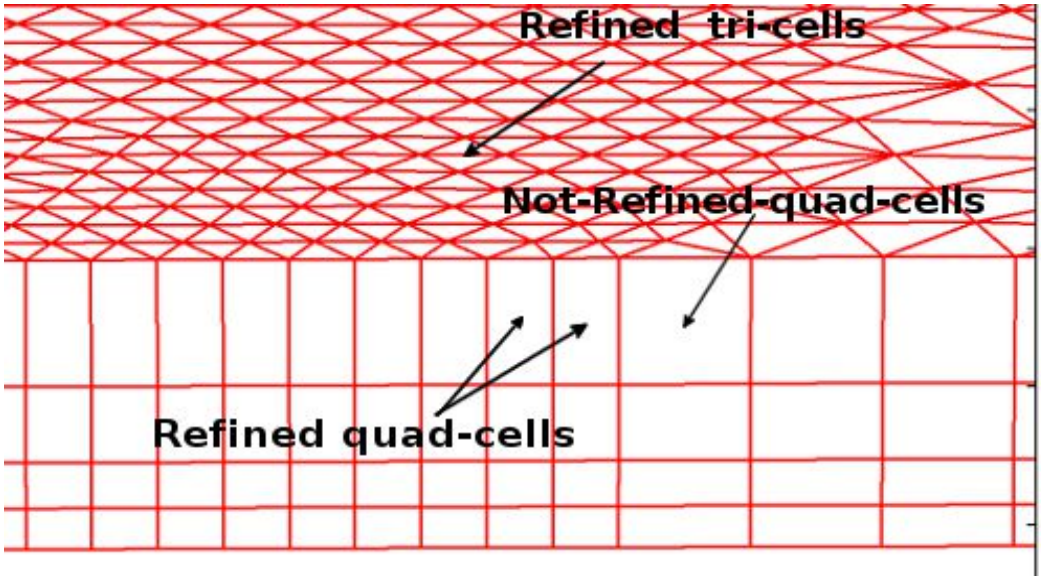


Fig. 10: local refinement at hybrid grid.

Test cases that we used and modified structured meshes, were, as mentioned previously at viscous flows with boundary layer, and due to this fact (friction around the geometry cannot be excluded) structured grids are used. Structured grids are also applied at inviscid flows as shown at chapter 4, with hybrid meshes.

HYBRID MESHES

After the subdivision of triangles and quadrilateral elements, it is worth mentioned to the way these two are combined and applied at each mesh change and every level of refinement.

Combining Triangles - Quadrilaterals

The decision to using both of them has a specific reason: Obtaining results that are close to realistic features. This practically means that possible discontinuities or distortions at some diagrams that will also occur due to other factors, are eventually avoided. A useful tool for choosing the best possible mesh, is proven to be the pressure coefficient plot.

A characteristic case is shown in the following picture 11.

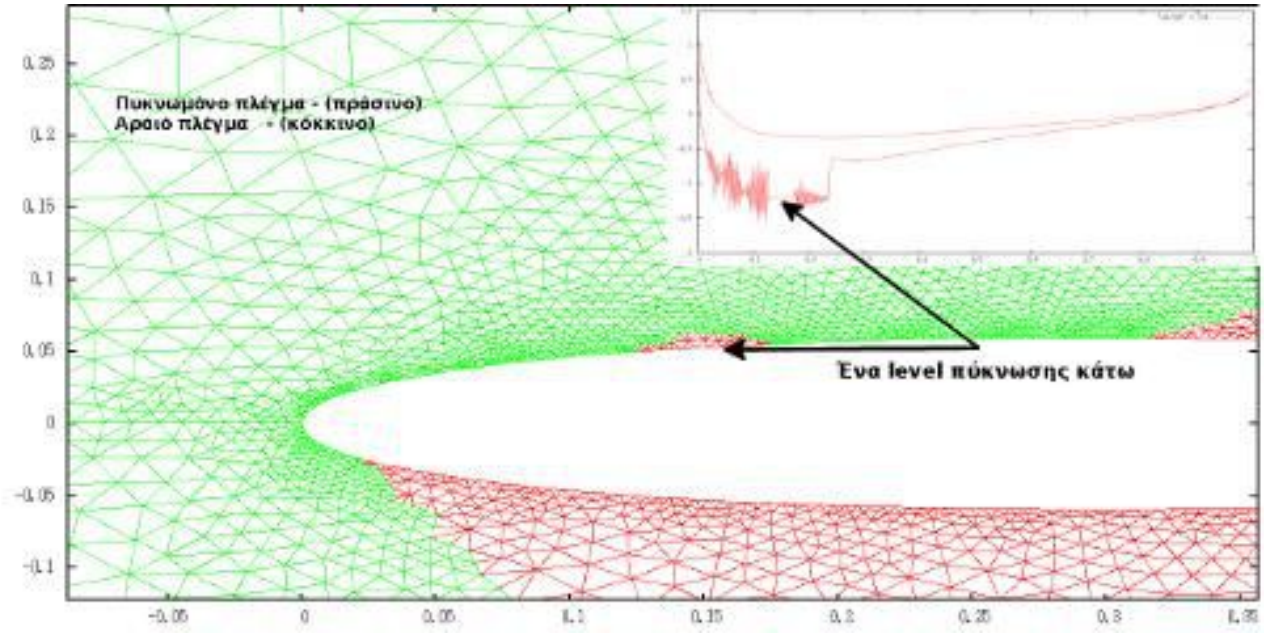


Fig. 11: Wiggles at C_p diagram for unstructured grid at $Ma_\infty = 0.7$, angle inflow = 2˚ ,NACA0012.

This case is from airfoil NACA0012, with mach number, $Ma_\infty = 0.7$, angle of inflow = 2°. The unstructured grid with triangles only has been refined (after a number of iterations so that the solution convergences). The grid with green color (refined) is o ne level of refinement up than the other with the red (coarse) .

The "wiggles" in this plot, mainly occur because of the fact that we use cell-centered finite volumes. As presented previously for the Navier-Stokes equations, at cell-centred methods of finite volumes the solution is calculated at the center of each cell and afterwards the fluxes at the edges. We calculate the the solution at a finite volume and then at the boundary the related fluxes by integrating the governing equations (Euler equations) using with this method, the neighboring cells (*left+right states*). So it is clear that the spatial discretization is related to the neighboring computational volumes and the distances from their centers. The big difference between the structured and the unstructured mesh is that these distances, from the centers to the faces are equal across the boundary (airfoil body). This is something that does not happen at the triangles where the distances can vary a lot at the boundary cells (fig. 12).

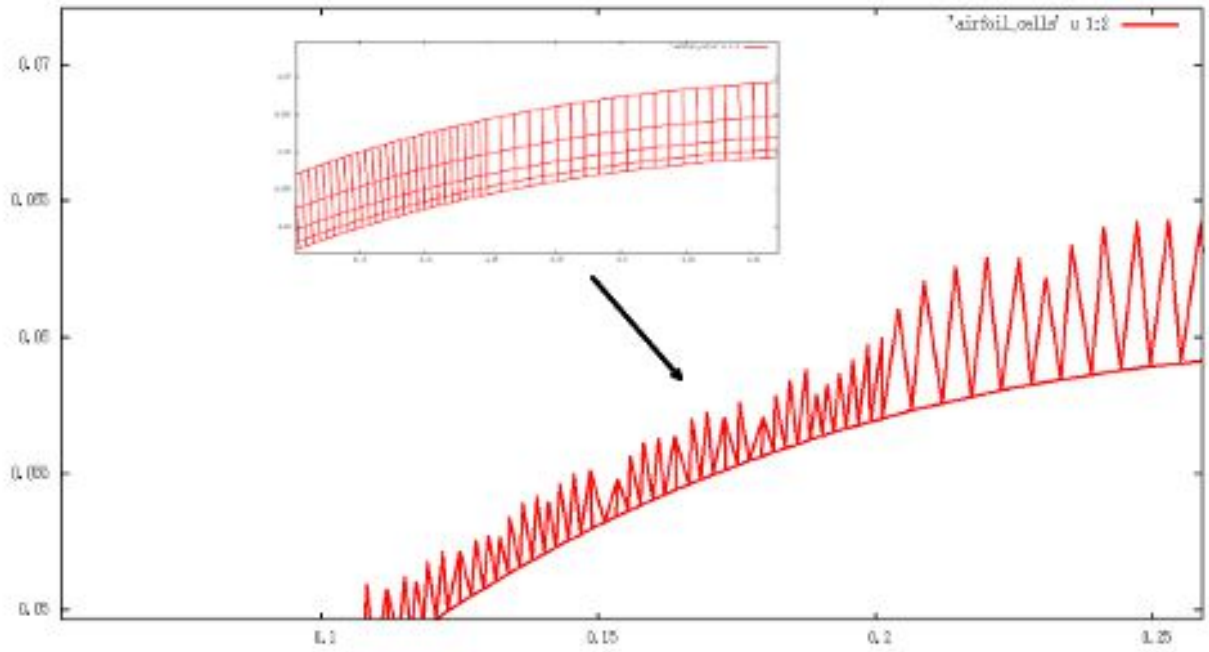


Fig.12: Boundary elements over NACA0012

Some more concepts about mesh Refinement

The solvers code, gives us the opportunity to indentify in every mesh the following among othes useful features:

– Geometric features of the nodes, like their coordinates.
– Geometric features of the cells of different shape, like volume, unit vectors, etc.
– Flow variables at the nodes.
– Neighbors of each cell and the connectivity.

This last one is important and gave us an idea for different refinement method. This method was focused to avoid the refinement of the quadrilateral cells, generating new cells only at the triagle elements. This can be easily achieved by defining an unstructured zone of at the coarse grid that is not allowed to be refined even if some elements satisfy the flow criterion (error-indicator). This area of not refinement will be triangles and quadrilaterals as shown in the picture 13.

So despite the fact that we may have cells that demand refinement and are not eventually refined, it is possible that we may also have overwhelmed nodes, meaning that many cells use one node. This unpleasant result is shown below at picture 14 and must be avoided.

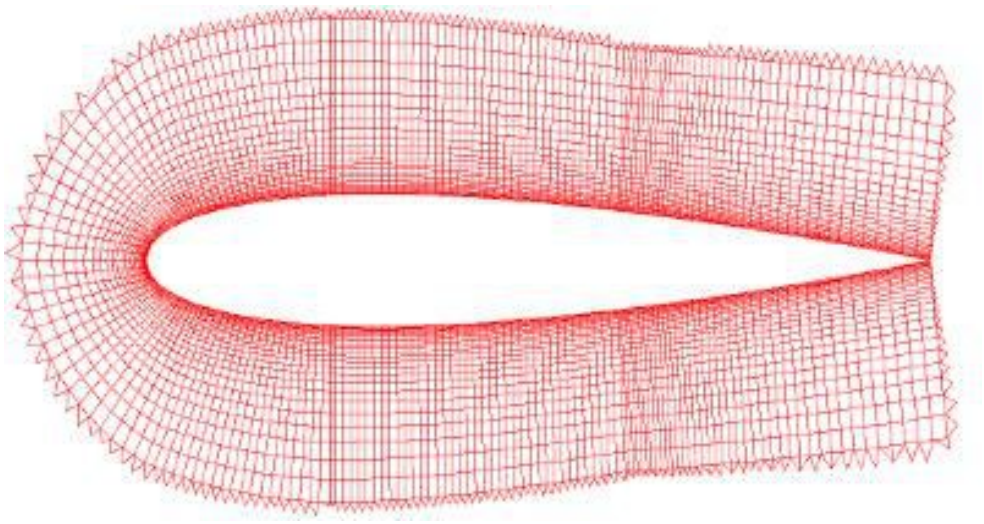


Fig. 13: Area of not-refinement -20 layers of structured and 1 layer unstructured grid.

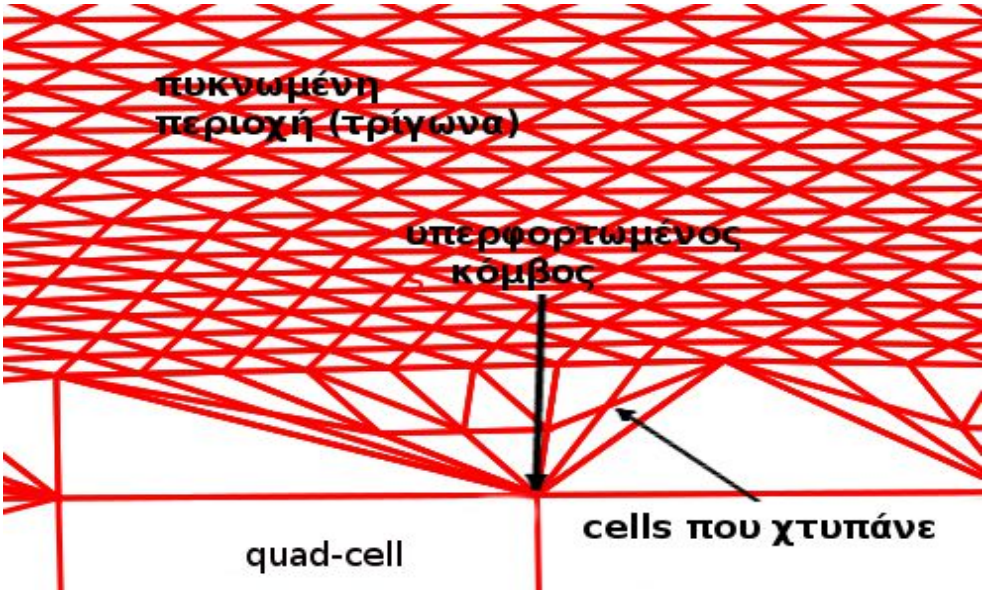


Fig. 14: Refinement after defining one layer of not-refinement can give

High-degree nodes and bad quality grids.

The above shows that the solution of no refinement case of one layer (coarse-element-layer) is not efficient. One can take the same result generating another layer of triangles where, together with the structured layer no refinement will take place (inviscid flow). This can easily be done adding another loop at the external loop of the code that checks for the first layer (procedure):

```
Initialise array layer not to be refined
index = 0
For each cell = 1 , NTE
   if Ngeom(cell).eq.2 then
      index = index +1
      procedure find neighbooring_cells
      if Ngeom(neighbooring_cell).eq.3 index = index +1
```

```
      endif
!  second layer of not-refinement ?
      If (second layer of not-refinement) then
         procedure find neighbooring_cells
         if Ngeom(neighbooring_cell).eq.3 index = index +1
      endif
! third layer of not-refinement ?
      ...
 area of refinement has been
 defined (only tri-elements)
```

Hanging nodes

The problem of these nodes has to do with edges that only one of the two elements that share this edge, is divided, as shown in picture 15, for the cases that occurred in this thesis and is about triangles and quadrilateral cells and is also related to the way the flow solver deals with these nodes.

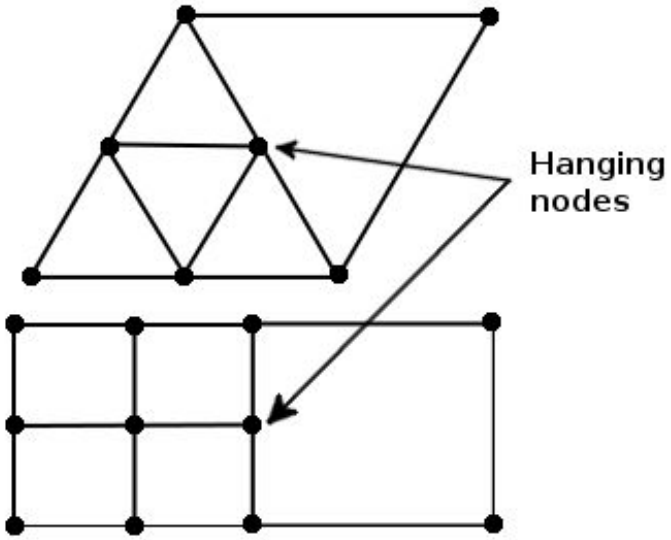


Fig.15) : Hanging nodes cases .The refinement solver makes Sure for the appropriate connections.

The solution to this problem is the connection of these nodes with the vertex that faces the divided edge. It is more clear in the picture 16. In that way we shall have two new children cells from the parent cell.

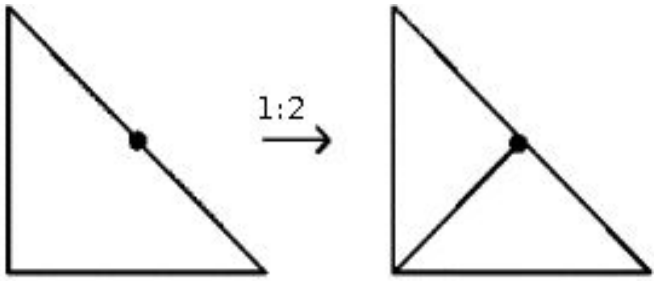


Fig. 16: Second kind of refining tri-elements (2D)

Finally it is worth mentioned that hanging nodes is not necessary to be connected in the above way and they can remain unchanged if the flow solver has this capability. A necessary condition is that the neighboring cells that hanging nodes exist, should not be refined more than one time so the variations of the volume are not very high. That will make sure that that the positions of the parent cells and the children cells are such that their centers are almost the same. There are avoided in that way [12]:
α) Effects at the calculation of the error of the fluxes.
β) The occurring Wiggles at the pressure coefficient.

STEPS OF THE REFINEMENT (A DJUSTED TO THE FLOW SOLVER)

The psedocode of the computational program that builds the new grid is the following:

```
 Program refine_msh.f90
!
! define area of refinement
!
 do nel = 1 , NTE
    read coase grid (tri+quad elements)
    if error_cell (nel)  >  error_min (eg 10d-6) then
       i = i + 1
       if (Ngeom(nel).eq.3) N_tri(nel) = i
       if (Ngeom(nel).eq.4) N_quad(nel)= i
    endif
 enddo
!
!  let the dimension of N_tri(i),N_quad(i)
!  be integers  m_1 , m_2 .
!
   call  compute_4_new_tri_cells (N_tri(nel))
   call  compute_4_new_tri_cells (N_quad(nel))
!
! fix  incompatible elements
!
  call contour (N_tri(nel),hanging nodes)
!
! the refinement area has been defined
! fix the rest grid ( new numbering applied )
!
  call  rest_grid (coarse grid \cap(N_tri(i)+N_quad(i) )
!
! interpolation
```

```
!
! at coarse mesh
  copy fluxes
!
! for new elements
!
  set fluxes
  deallocate array fluxes
!
! at fine mesh
!
  allocate new array fluxes
  interpolate fluxes
  deallocate  array fluxes
!
 END of program refine_msh.f90
```

Thoughts and aspects of the geometry

In order to achieve a suitable refinement we must, regardless of the decisions that were mentioned before and have to do with the mesh quality (Aspect ratio, minimum and maximum angles, volume ratio of the finite volumes that neighbor) take care the following parameters:

• *Tolerance* of the initial grid while designing. This tolerance has to do with the line that the curve of the airfoil will be approximated. This approximation is accomplished with a method of numerical interpolation of the initial boundary points that the user gives. If this initial grid is constructed by a software package like ANSIS-ICEM we can define it and keep it small.

• *New boundary nodes*. These nodes are constructed at the surface of the airfoil and their coordinates are the half of the sum of the corresponding vertices that belong to that divided edge. As a result the new node and every new boundary point that is constructed at each level of refinements, at edges that belong to the airfoil boundary, follows the initial discretization which clearly differs from the airfoil (fig.17).

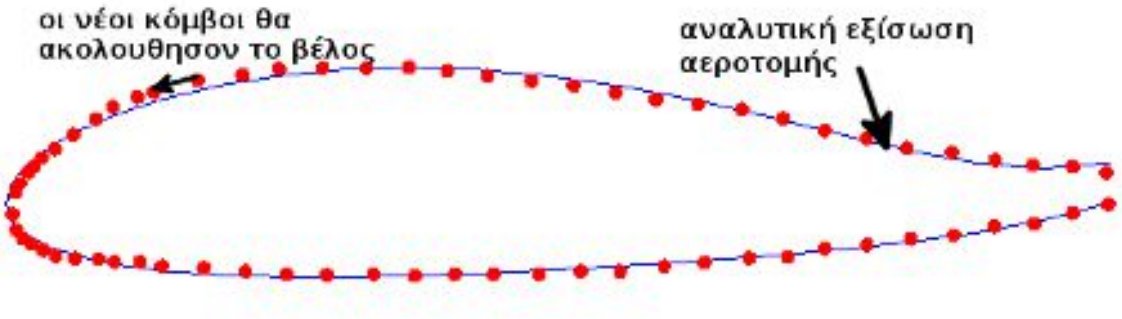


Fig.17: Initial discretization of a random airfoil and the analytical equation if known. The equation which passes through this points has a tolerance (input).

The problem that occurs is more clear at areas with much more levels of refinements. The problem of the new boundary nodes can be handled with the following ways (both tested at this thesis):

α) Approximation with interpolation at the subroutine which computes the new nodal points. The interpolation with B-Splines between two points will pass a curve that is affected by the gradient at these two points, giving a point that is well fitted at the airfoil Naca0012 or any other airfoil (*Constrained fitting for airfoil curvature smoothing* [13]).
The interpolation will be applied every time that the subroutine of the refinement is applied for the new nodes.
This method was successfully developed approximating the boundary faces with second degree splines (fig.18).

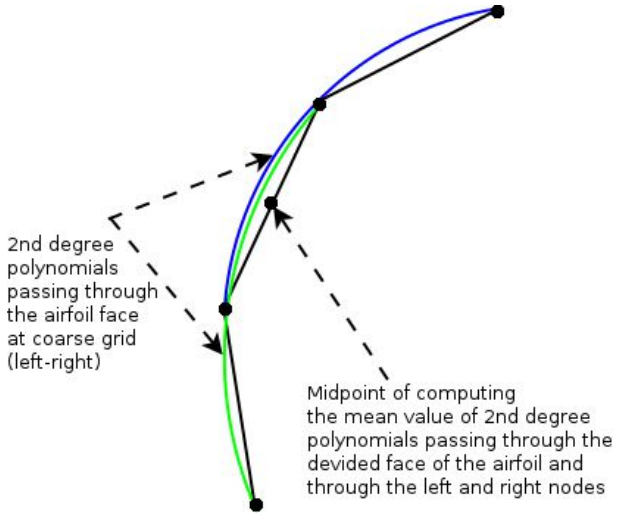


Fig.18) : Method of computing the new nodes at airfoil boundary with splines. The 1st derivative of each polynomial is continuous.

β) Airfoils equation (**no interpolation**).

We try the analytic equation of the airfoil for fiing the discretization at the boundary nodes and their midpoints. In every space $[x_i, x_{i+1}]$ corresponds to two nodes of the cell that is divided to new ones giving a new node at the midpoints.
During the iterations for the solution of Euler equations the refinement subroutine is called and the new nodes on the airfoil contour can be computed from the analytic expression of the equation. In that way points correspond to the real shape of the boundary. This method can be applied only when the analytic equation is known.

The equation of NACA0012 :

$$f(x) = \pm 0.594689181 \cdot (0.298222773x^{0.5} - 0.127125232x - 0.357907906x^2 + 0.291984971x^3 - 0.105174696x^4)$$

The method of analytic equation was applied in this thesis successfully increasing the smoothness of the boundary surface, as, especially in cases of supersonic flows as proven (chapter 4) this can be an important factor for the wiggles that occur at the pressure coefficient graph (figure 19).

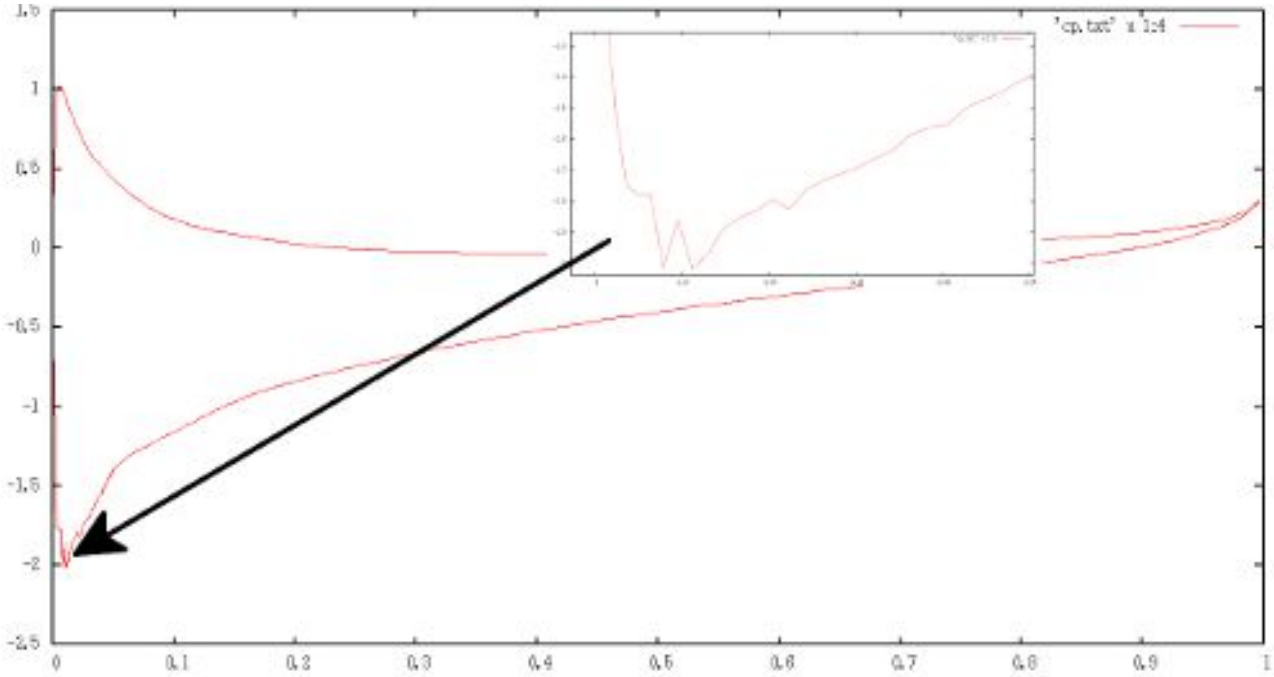


Fig. 19: Cp –x/chord diagram over Naca0012 , Ma∞ = 0.1 , angle =5˚ ,Re =10e-6.

Functions for refinement solvers at finite volume methods

Let **F** be the quatity/criterion, such as pressure, density, and others, then the grid adaptation can be applied in areas related to this quantity, meaning that a criterion will be a function of **F**. Some of these functions can be:

– Absolute difference (difference of the considering quantity) $f = |F_i - F_j|$ between two nodes i, j .

– Relative difference (divided with the local mesh size).

Let's call ds the local length, between i, j, then we may use: $f = \frac{|F_i - F_j|}{ds}$, or some other versions like the following:

$$f = 1 + \left|\frac{dF}{ds}\right|,\ f = \sqrt{1 + (\frac{dF}{ds})^2},\ f = \left|\frac{dF}{ds}\right| + B\left|\frac{d^2F}{ds^2}\right|, B = constant$$

The functions above are expressed dimensionless, using the mean value of f at the face: $f = \frac{|F_i - F_j|}{|F_i + F_j|}$ and if we want to amplify or discourage the rate of big or small vertices division, we may use weight functions increasing of decreasing the value of f ($f \rightarrow f \cdot w(l), w(l) = weight\ function$).

Adaptation and criterion functions behavior

In case of *linear distribution* of the quantity, F the division of the face (ij) will give two new children cell edges that across them, due to the linear distribution, will have variations that will be equal ($F = Cx + d \rightarrow dF = Cdx, C = const.$). It seems more possible to achieve a new level of refinement at the children - faces (*h-refinement*). This means that the adaptation can be bounded at certain areas (bounding edges Sierpinski's triangle) something that can be pleasant or not.

Refinement functions at inviscid flows [14]

Cell centered scheme

As mentioned in a previous chapter, the flow solver of Navier-Stokes equations computes the variables at the centers of each finite volume. This means that the following computations are applied at the center of each cell (fig. 20) and let us call two random cell centers i, j.

The expressions that can be used for criteria for strong shock waves and are mainly:

1) The square of the norm of velocity change:
$f(\vec{V}) = (\Delta u_{ij})^2 + (\Delta v_{ij})^2$,$\Delta u_{ij} = u_i - u_j$,$\Delta v_{ij} = v_i - v_j$.

2) The absolute difference of Mach number:
$f(M) = |M_i - M_j|$,ενός

3) The absolute difference of the (static) density:
$f(\rho) = |\rho_i - \rho_j|$.

4) The dimensionless difference of (static) density:
$f(\rho) = \frac{|\rho_i - \rho_j|}{\rho_i + \rho_j}$.

5) The absolute difference of the pressure:
$f(p) = |p_i - p_j|$.

6) The absolute difference of the static pressure:
$f(p) = \frac{|p_i - p_j|}{p_i + p_j}$.

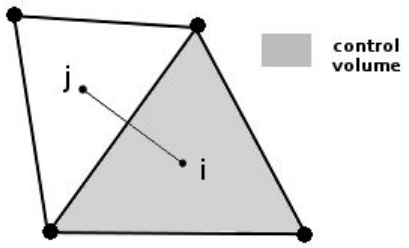


Fig.20): For every cell-center ,the neighbors are used for some criteria.

Special cases:

– For weak discontinuities of the flow we can use as refinement functions, expressions of pressure, density and velocity which can be quite efficient.
– Stagnation points are revealed only with refinement functions related to the velocity
– *Viscous layers* are captured with the vector of the rotation which is:

$$f(\overrightarrow{V}) = |rot\vec{V}| = \left|\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right|.$$

– *Circulation areas*, combination of criteria of rotation and Mach number.

The refinement criteria that used mach number and density appeared to be almost equally efficient and were selected for transonic flows with shock waves. Pressure can also be successfully used for finding the sub regions that mesh changes must be applied. The same holds for the variation of velocity. All these quantities are inserted as input at the refinement solver where the user defines the criterion/a of his choice.

The reverse procedure - Mesh coarsening

(We will briefly define the following notations for the coarsening**)**

Let us call: $G(l,t)$, a grid that the time t has a level of refinement l
(with $l,t \geq 1$)

For cases of $2D$ problems and triangle cells, there are three possibilities:

– The four new cells of $G(i,t-1)$ are replaced at $G(i-1,t)$ by the parents they come from at $G(i-1,t-2)$.
– The two new cells in the second layer of refinement, are replaced by the parent cell they were constructed.
– The four new cells are replaced by two new cells.
All these cases are shown below:

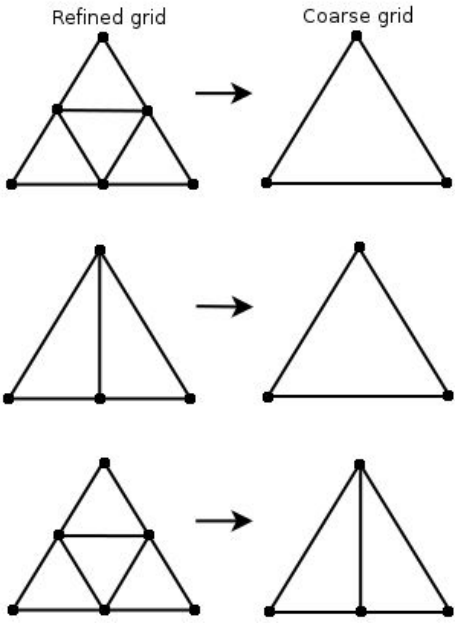


Fig.21: Derefining unstructured mesh, with triangle elements.

3.4 Interpolation of the variables

After constructing the new mesh we must transfer the fluid information at the new cells. For this purpose we may use arrays that this information (the flow variables) is temporarily is stored. This memory allocation is temporary and done only for the current time step, before the flow solver continues the solution of the problem.

```
Initialise temp_arrays
for each flux variable allocate temp_arrays  /duplicate
     temp_arrays = flux_arrays      /flux_arrays      coarse grid
     deallocate flux_arrays
 assign fluxes at refined grid
deallocate temp_arrays
```

Cells that do not change.

The interpolation at the cells that do not change and keep their geometry is simple: all the flow propertiesat the previous time step at the coarse grid are transferred to the refined grid. So the concervative fluxes remain the same at the next time step.

Cells that are decomposed.

The control volume Ω is divided to either 2 or 4 new ones Ω_i and will obviously be: $\Omega = \sum_{i=1}^{2ή4} \Omega_i$ and $(\rho\Omega)_{coarse} = \sum_{i=1}^{2ή4} \rho_i \Omega_i$, for the new volume and mass. The new cells will get the physical information of the parent cell they became from (fig.22).

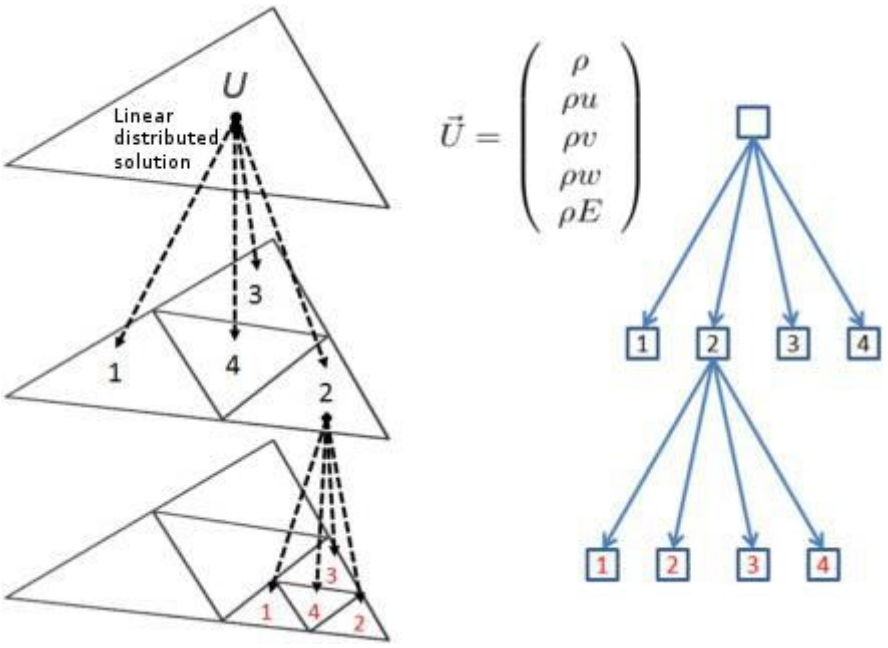


Fig.22 :Interpolation of variables for unstructured mesh.

4. Test cases

4.1 Inviscid flow, $Ma_\infty = 0.7$, angle of inflow = 2°.

In this particular case we have a shock wave at only one side of the airfoil (suction side). We want to construct a mesh suitable for this phenomenon. This is accomplished with the right criterion and the geometric limitations used by the solver. After trying multiple refinement functions, mach difference appeared to be quite efficient. The solution is at first approximated using a coarse unstructured grid with 3658 cells. The refinements that were applied to the area of interest (the shock wave at fig.1) are shown at the following pictures:

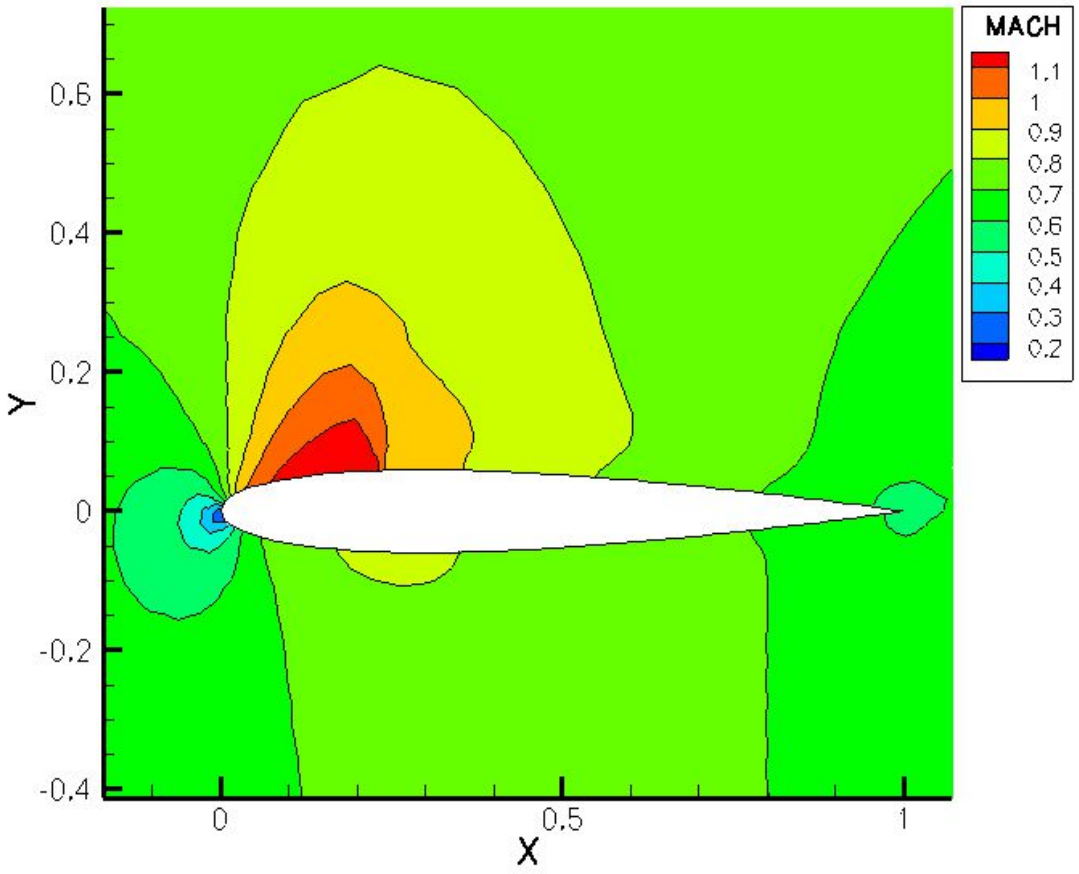


Fig. 1: Area of interest (Mach =0.7, inflow angle = 2˚).

Refinement criterion was the absolute difference of mach number (0.12), and mean error of energy at each cell (10d-03).

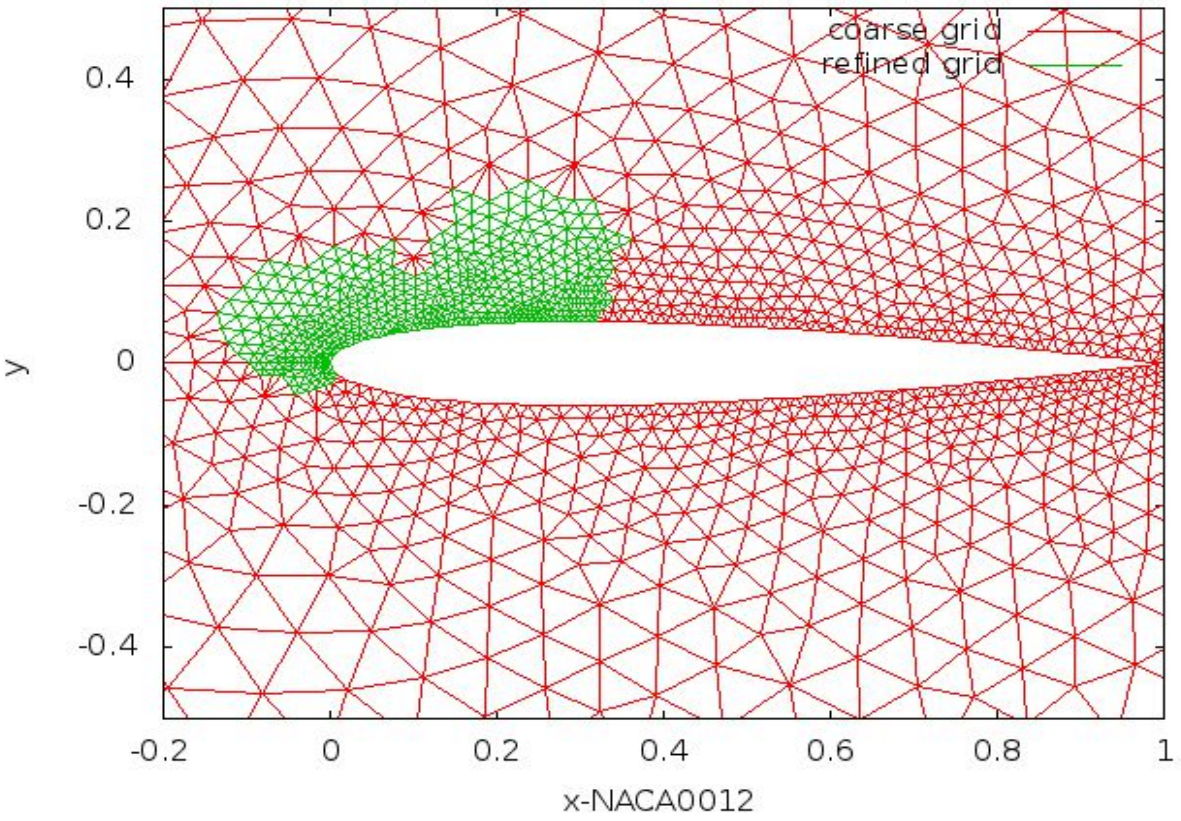


Fig.2: 1st refinement, 4323 triangle elements, Ma_∞=0.7, angle =2˚, Naca0012.

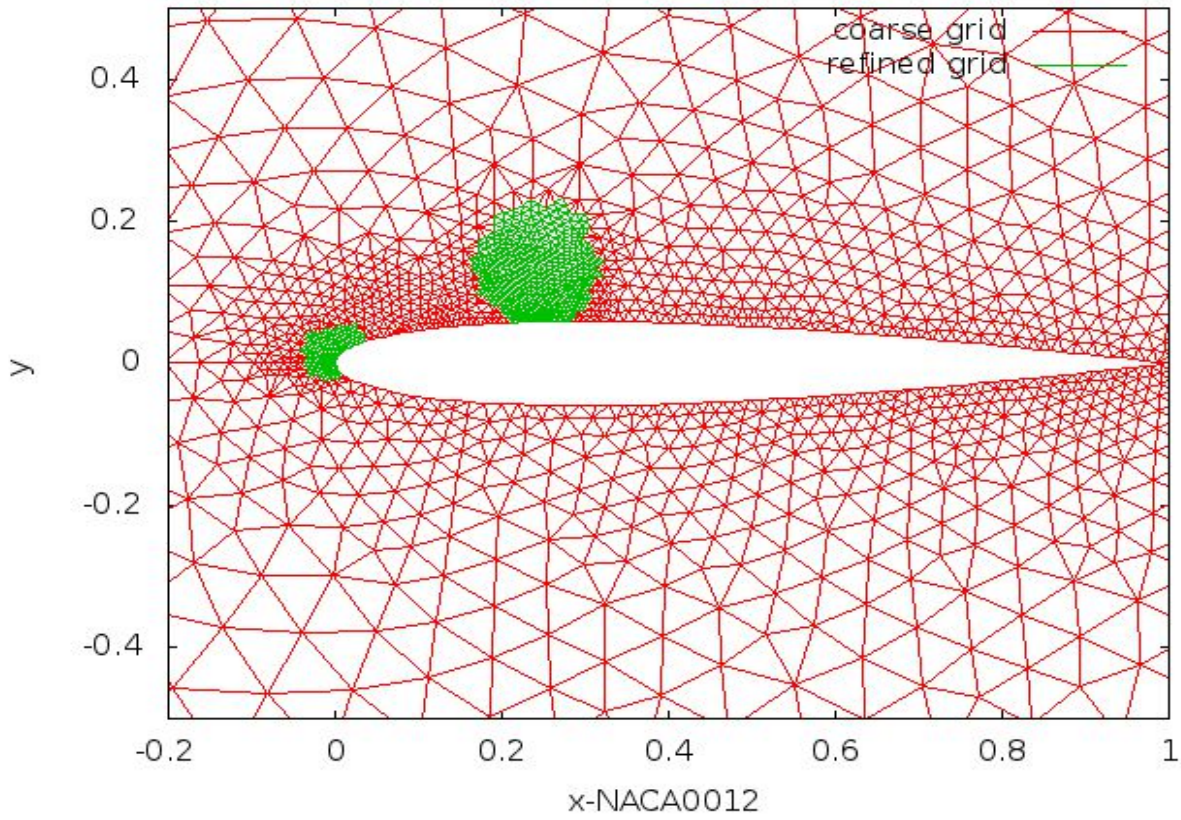


Fig.3: 2nd refinement, 5383 triangle elements, Ma_∞=0.7, angle =2˚, Naca0012.

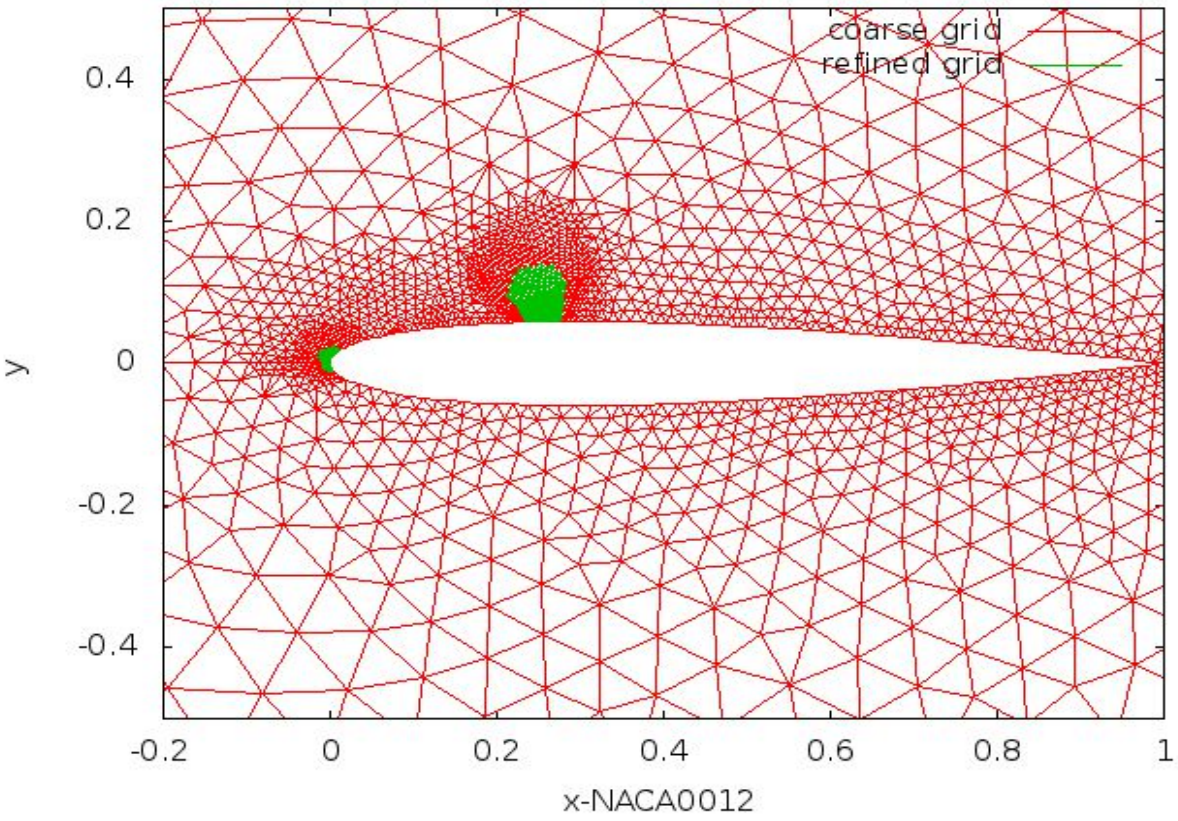


Fig. 4: 3rd refinement, 6671 triangle elements, Ma∞=0.7, angle =2˚, Naca0012.

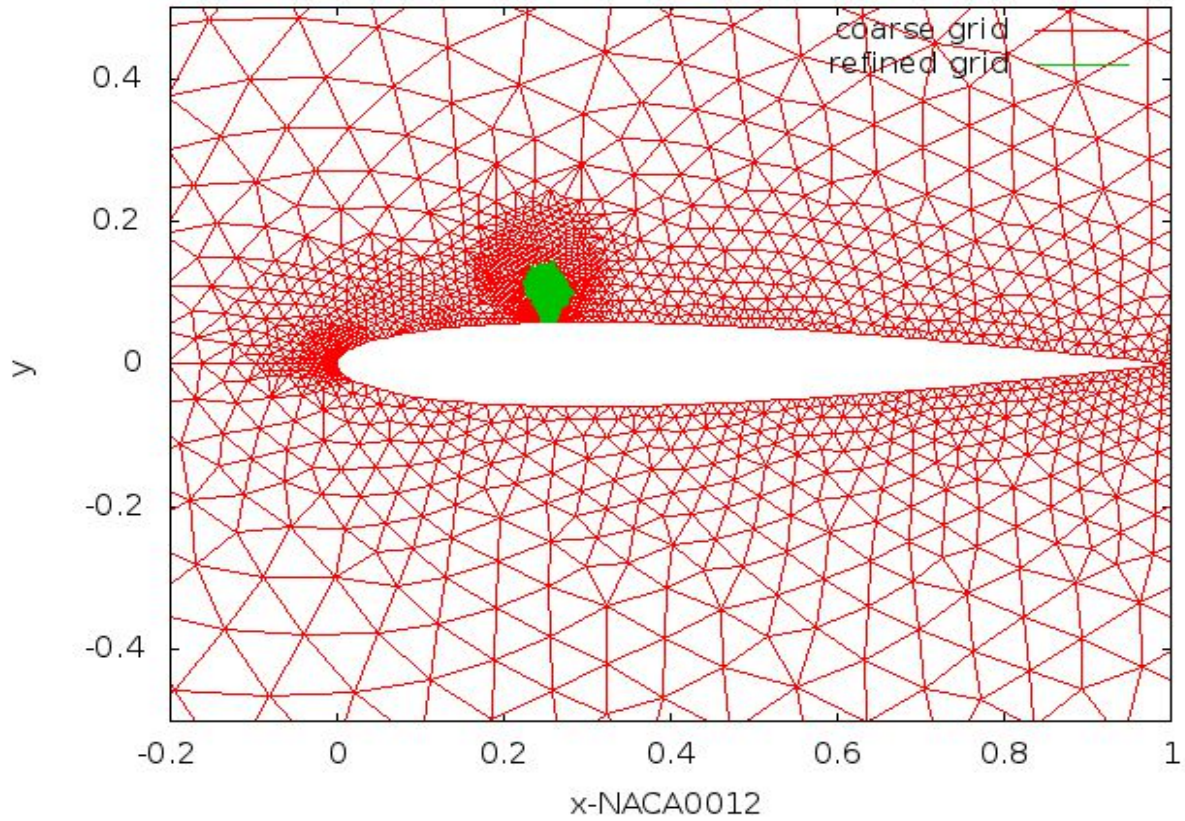


Fig. 5: 4th refinement, 9353 triangle elements, Ma∞=0.7, angle =2˚, Naca0012.

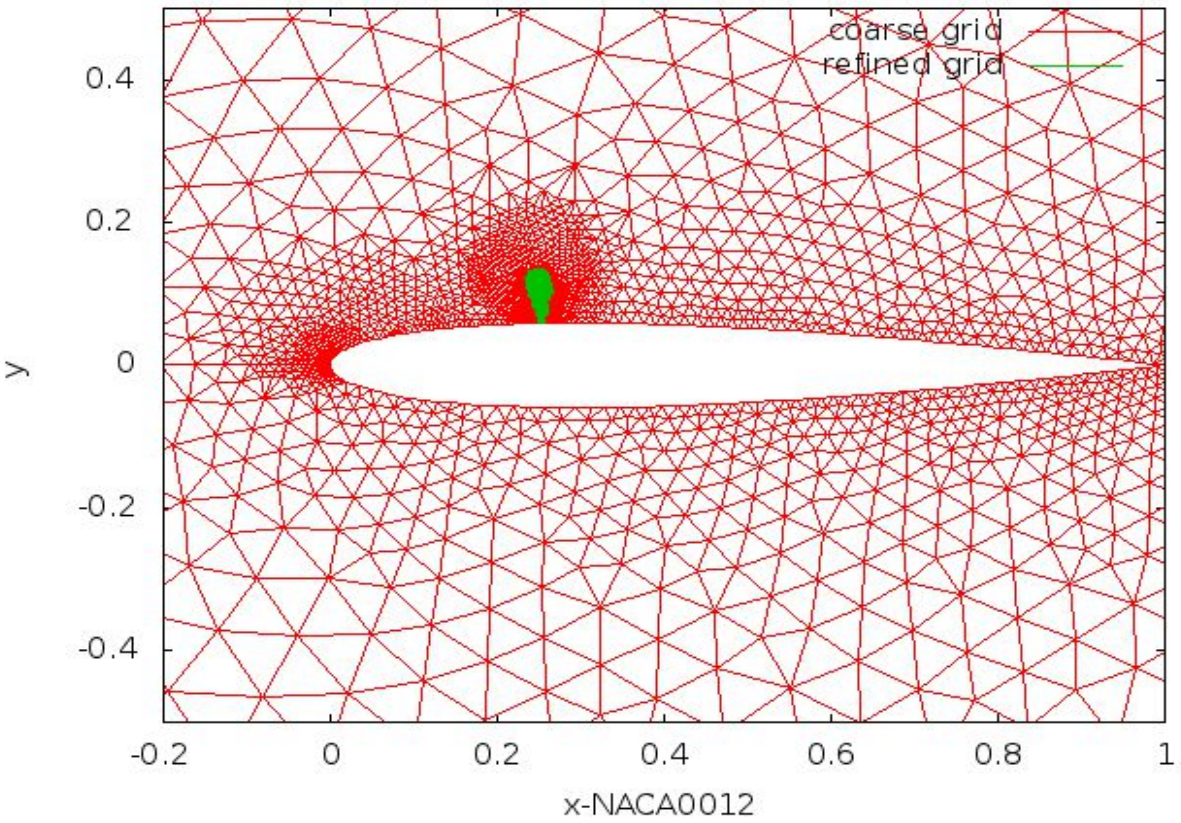


Fig. 6: 5th refinement, 14696 triangle elements, Ma_∞=0.7, angle =2˚, Naca0012.

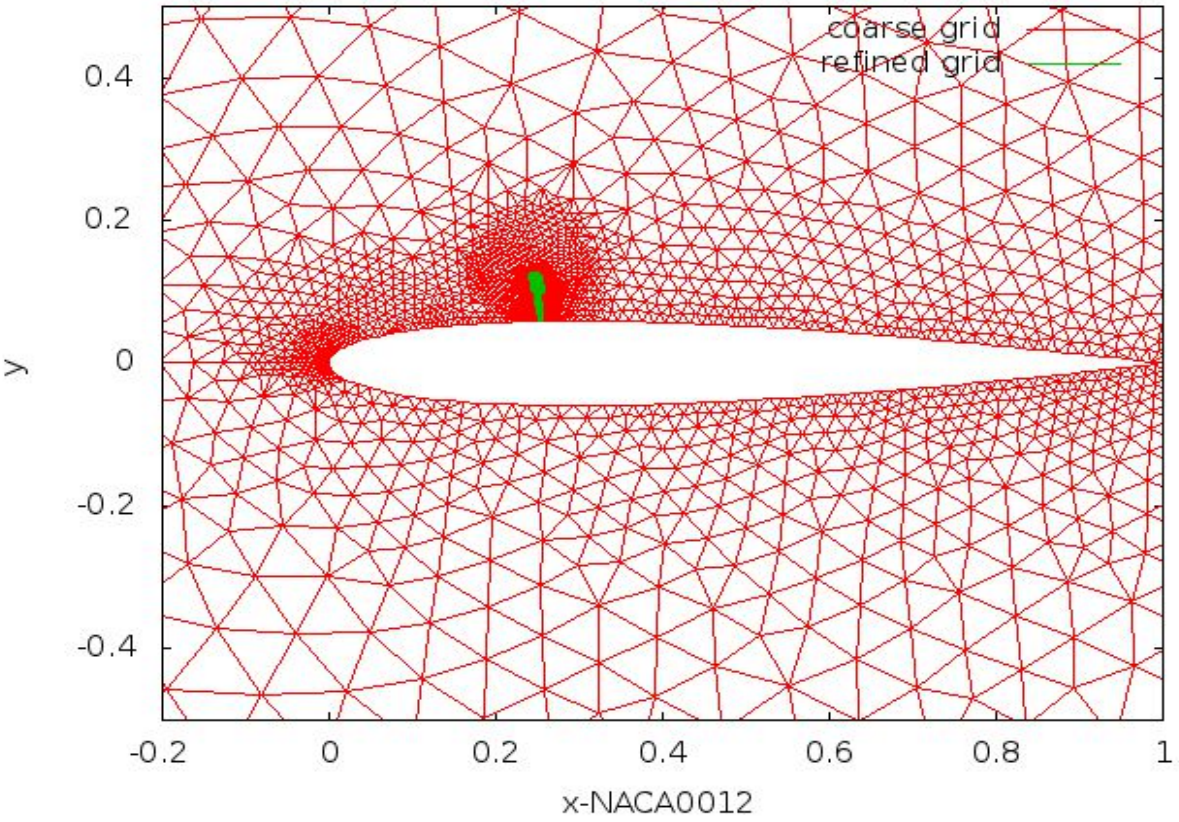


Fig.7a: 6th refinement, 24442 triangle elements, Ma_∞=0.7, angle =2˚, Naca0012.

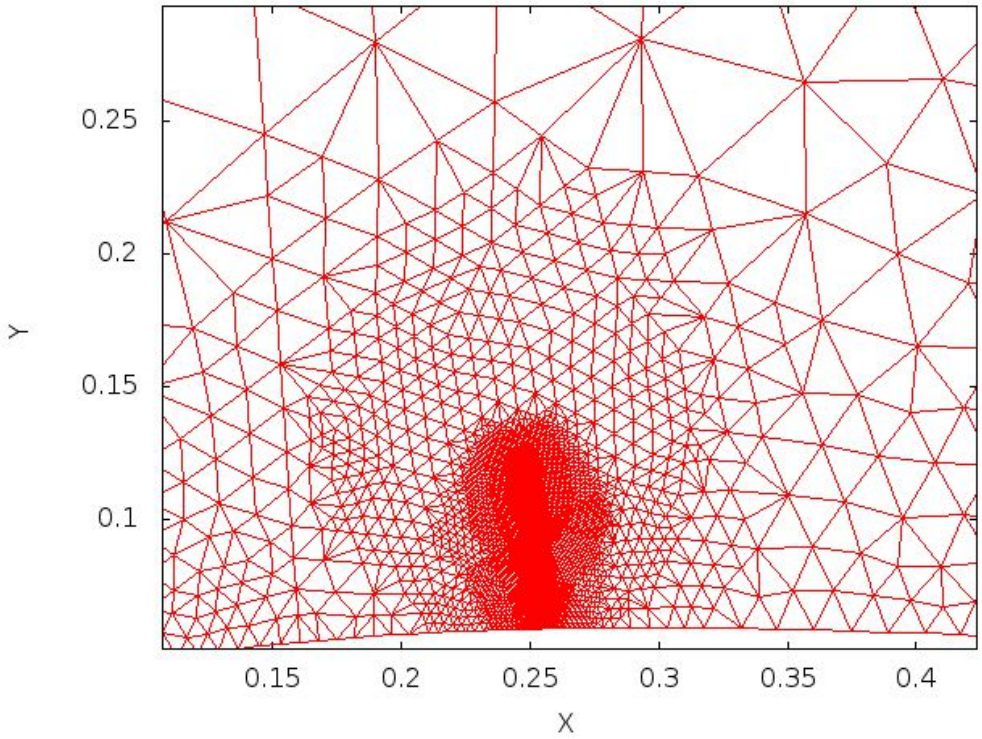


Fig.7b: Refined mesh at shock wave region, Ma∞=0.7, angle =2˚, Naca0012.

The pressure and lift coefficient graphs follow next at figures 8 and 9. Both cases of refined and non refined grids are plotted for comparison. These two as we will see the convergence of the mean error of density differ long before the solver is called. The reason is that different CFL and number of internal iterations the flow solver uses at each case. As a result final CFL equal to 20 for refined and 5 for non refined grid the iterations to reach the CFL were 100 and 1 respectively.
CFL number can be an important factor that can affect the convergence. Large values the rate of congruence increases and very high CFL may lead the solution to divergence.
In general there were used small values of final CFL such as 20 or less. So in general it can affect the levels of convergence mean error.

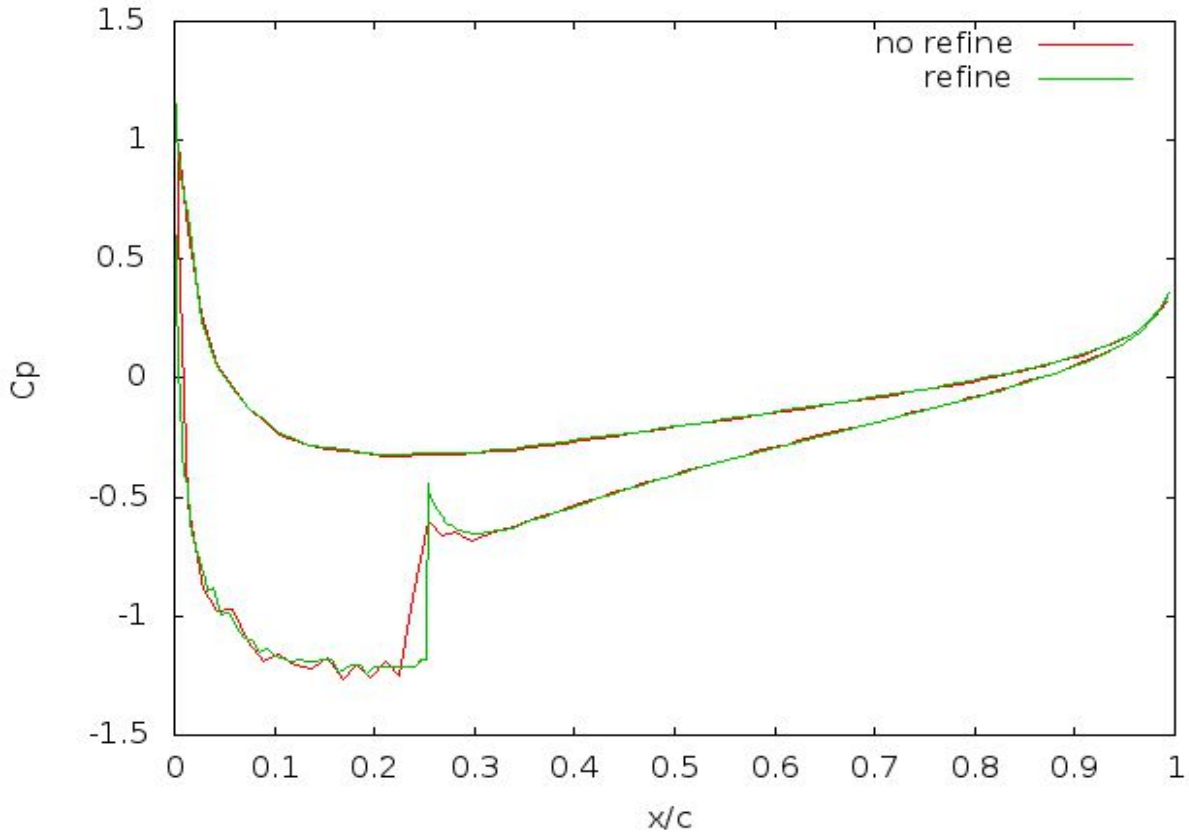


Fig.8: Pressure coefficient, for Ma_∞ =0.7, angle=2.0˚, Naca0012

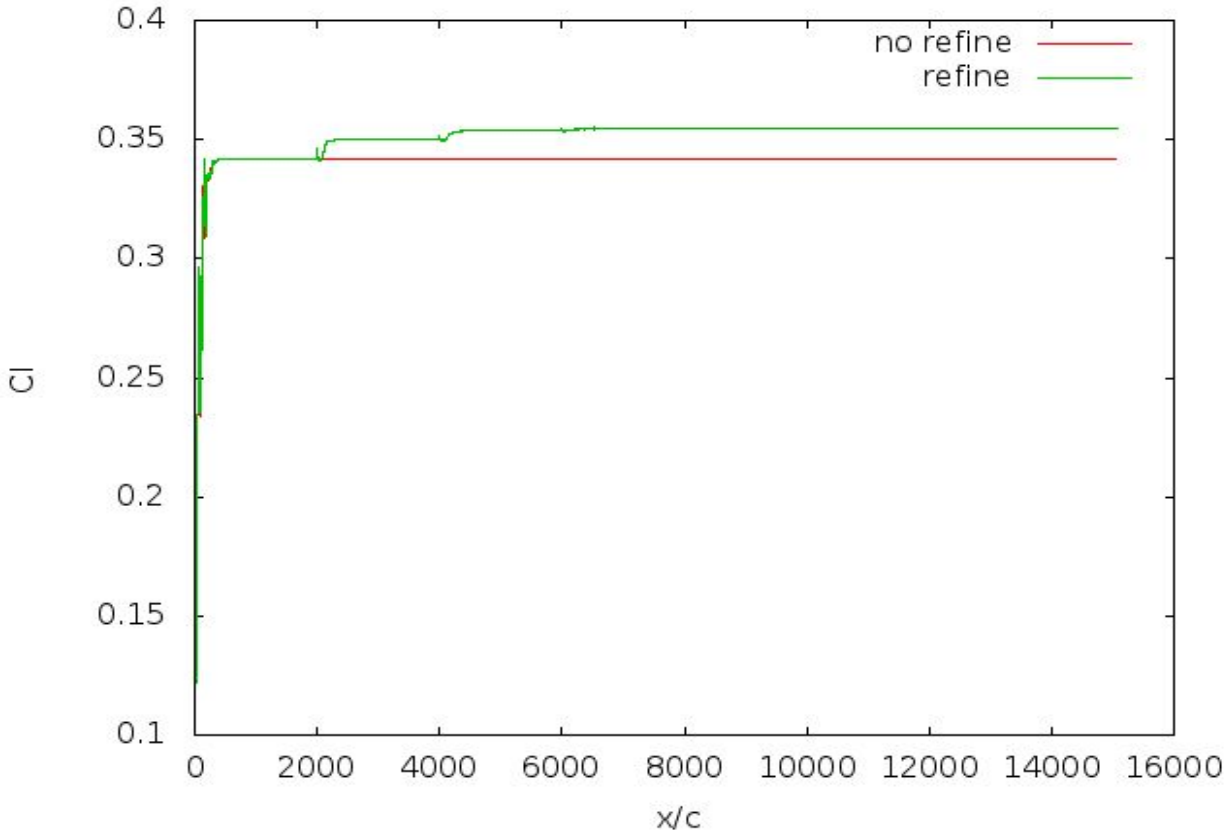


Fig. 9: Lift coefficient, for Ma_∞=0.7 angle=2.0˚ Naca0012.

The graph for the mean error of density with respect to time steps is the following (fig. 10):

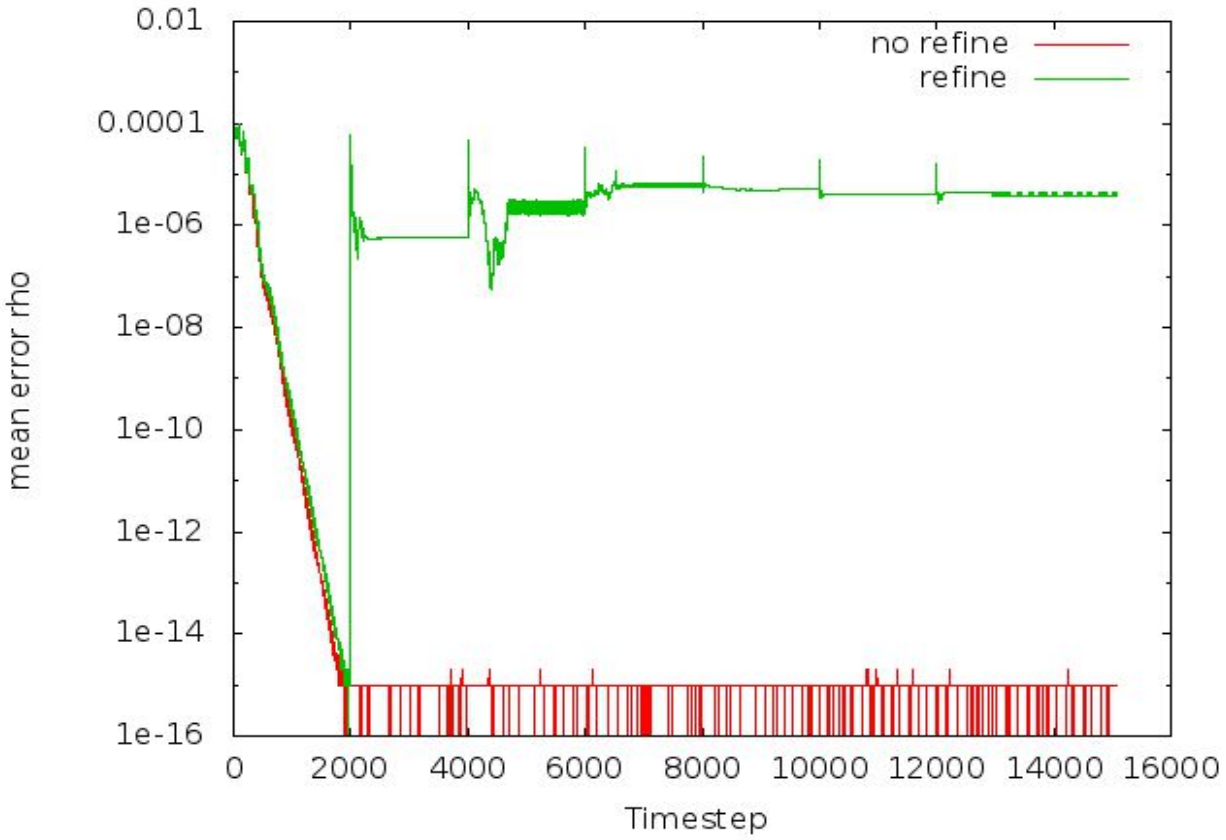


Fig.10: Mean density error through convergence history. Every peak assigns to a refinement. Refined case CFL final 20, Non-refined case CFL final 5. 100 iterations to reach cfl final for Refined case, 1 for non-refined case.

Results:

– The figure of pressure has distortions at the suction side at almost the 25% of the chord. These wiggles also occur after applying the refinements, but they are smoother. This is caused by the triangle shape of the cells across the contour of NACA0012.
– Lift coefficient changes and convergences at a higher value with the new grid.
– Mean error of density jumps from 10e-16 at 5.72e-07 reaching a mean value around 3.88e-06. In order to explain this big difference, let us have a closer look the area of interest after the 6th refinement (final grid):

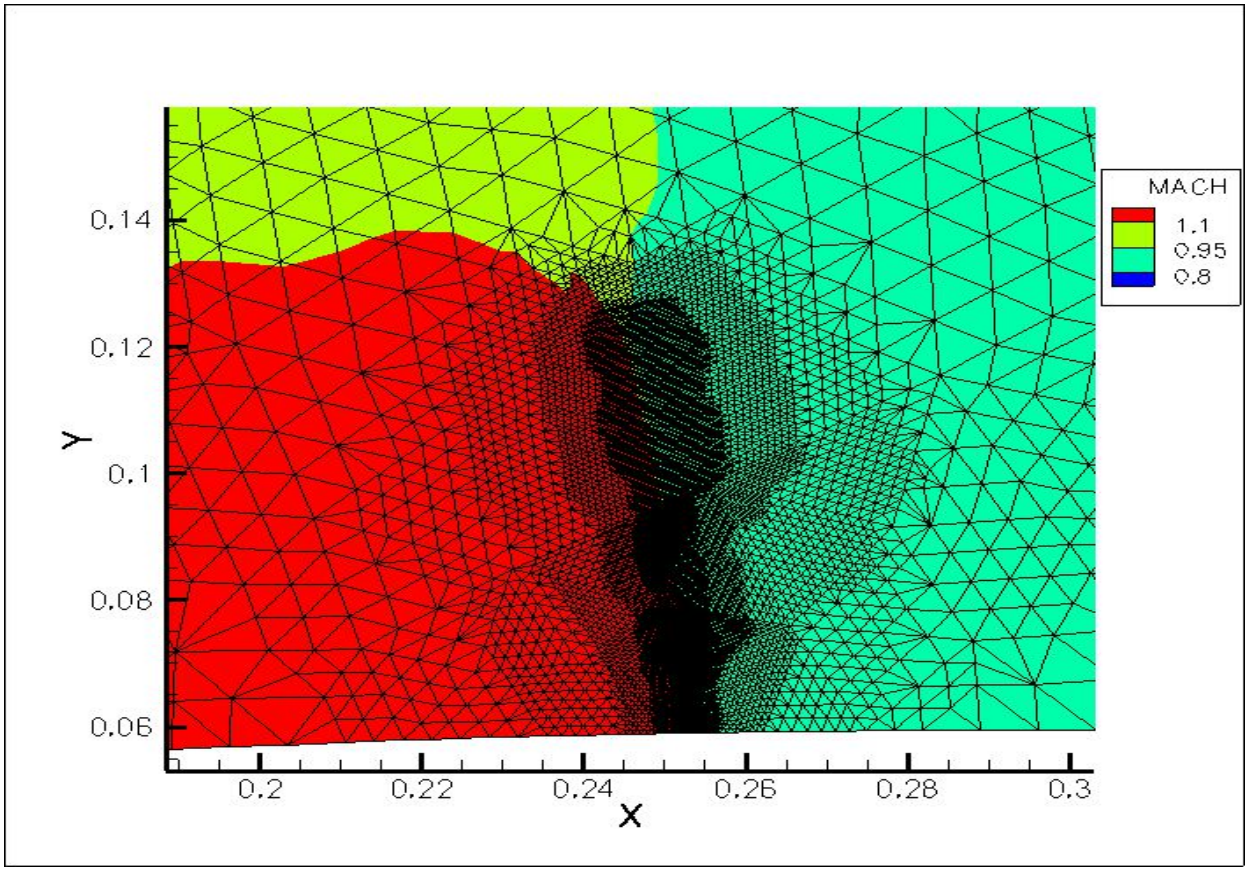


Fig.11: Shock wave and grid placing it after the final refinement.

The area of the refinement at a closer look. The shock wave is in the middle of the new grid region (fig. 11-12).

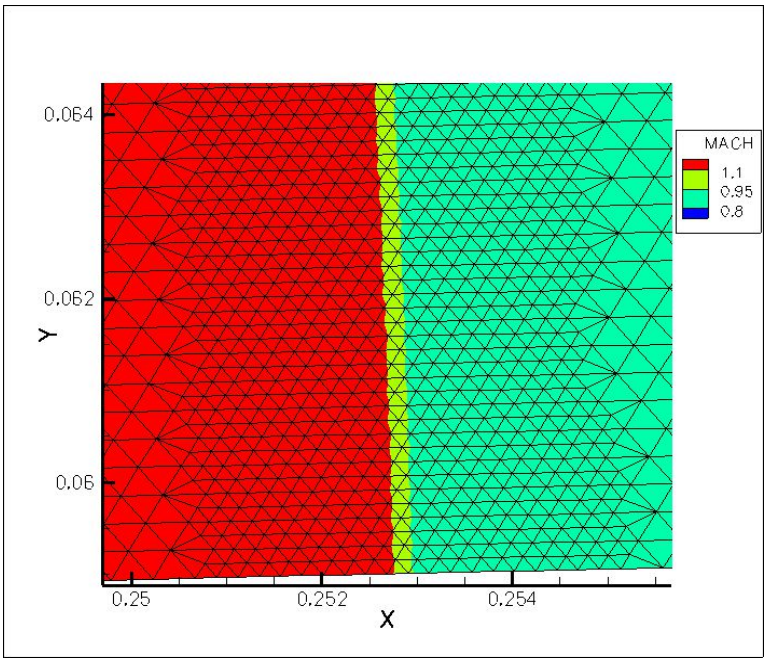


Fig.12: Shock wave grid at closer look.

Taking as criterion the absolute difference of mach number between two neighboring cells, and taking as a second criterion of refinement the mean error of density then for Mach difference = 0.10 and limit for the error at each cell = 10d-03, the shock wave is located and the solver keeps refining the mesh until a limit to the number of cells (which in out our case was chosen to be 20.000 cells). We observe that there exist inside the refined areas, sub regions where the mean energy error is larger than the others (fig. 13).

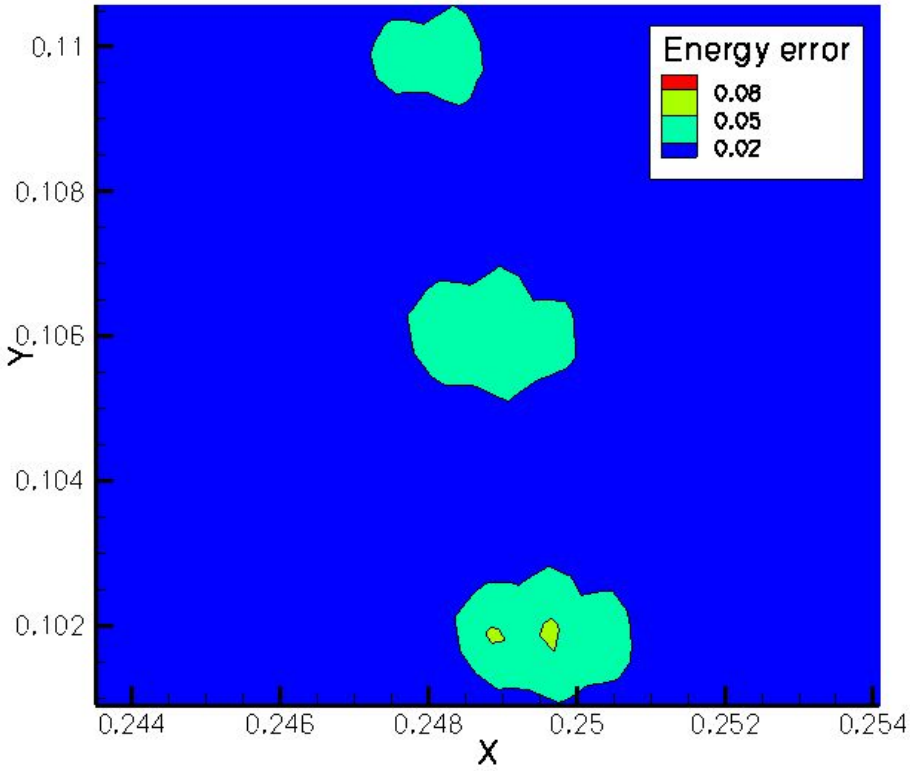


Fig. 13: Sub-regions with large error of energy .Inviscid flow, Ma_∞=0.7, angle =2.0˚.

These sub regions shown above with light green color may have a mean value that is 3 or more times larger than the rest domain and cause the increment of the mean error in high levels close to 10e-06.

– **Absolute Mach Difference = 0.12, Mean Error = 10d-03 (stricter criterion)**

In this case the refinement solver will fix the mesh only the are of the shock wave and not in the stagnation point as before. The criterion is more demanding and the refinement is even more targeted. Moreover we only get one level of refinement (fig. 14). The new grid will have only 3999 cells. This means that it will refine ~ 9% of the initial mesh (3658 cells). The mean error of density falls at the levels of the previous case (fig. 15).

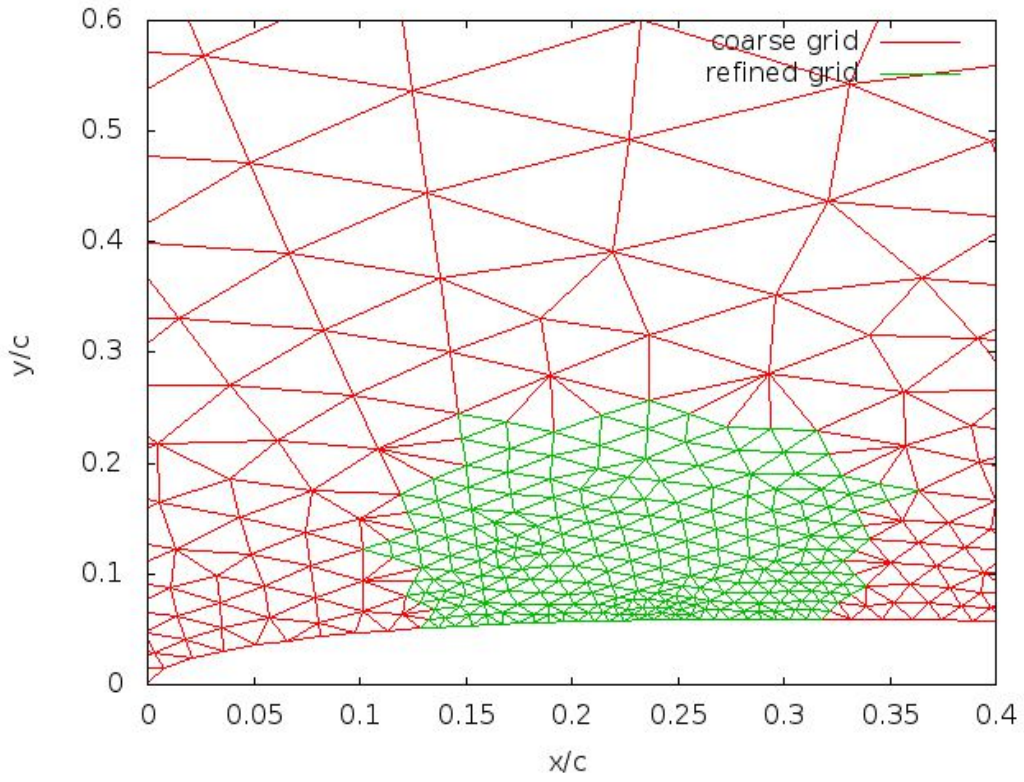


Fig.14: 1st refinement Ma_∞ =0.7 angle =2.0˚, Mach- difference with neighboring elements =0.12.

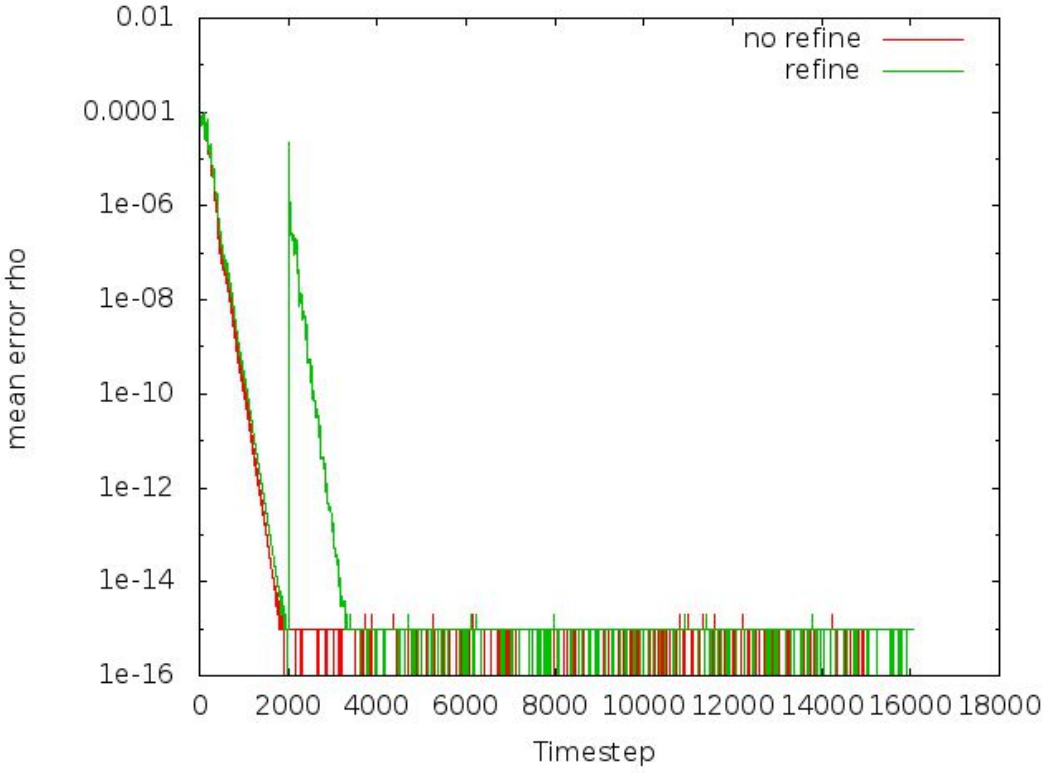


Fig.15: Mean error of density after 1st refinement with stricter criterion. Inviscid flow, Ma_∞=0.7, angle =2.0˚.

– Using triangle and quad cells.

The same criterion absolute Mach difference and mean error per cell can be successfully be applied in a coarse grid with triangles and quadrilateral cells at an initial mesh with more cells (initial number = 10413 cells):

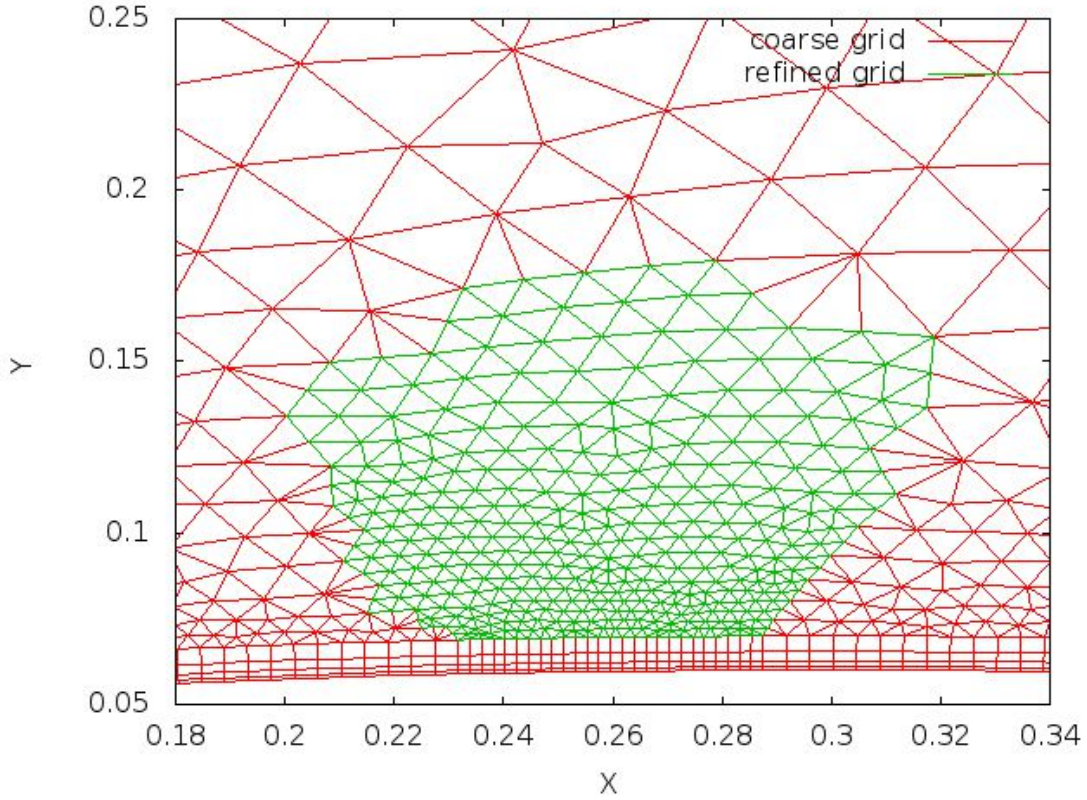


Fig.16 : Grid after 1st refinement ,10980 elements, Ma∞=0.7 ,angle =2˚ ,Naca0012.

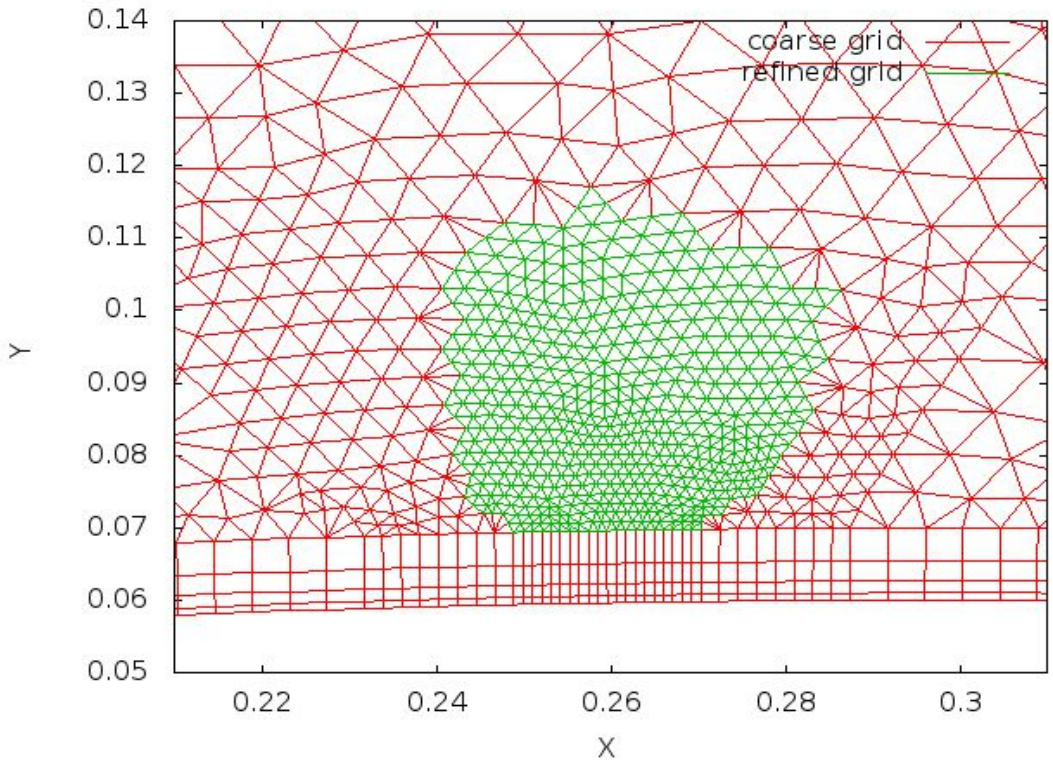


Fig.17:Grid after 2nd refinement , 11728 elements, Ma∞=0.7 ,angle =2˚ ,Naca0012.

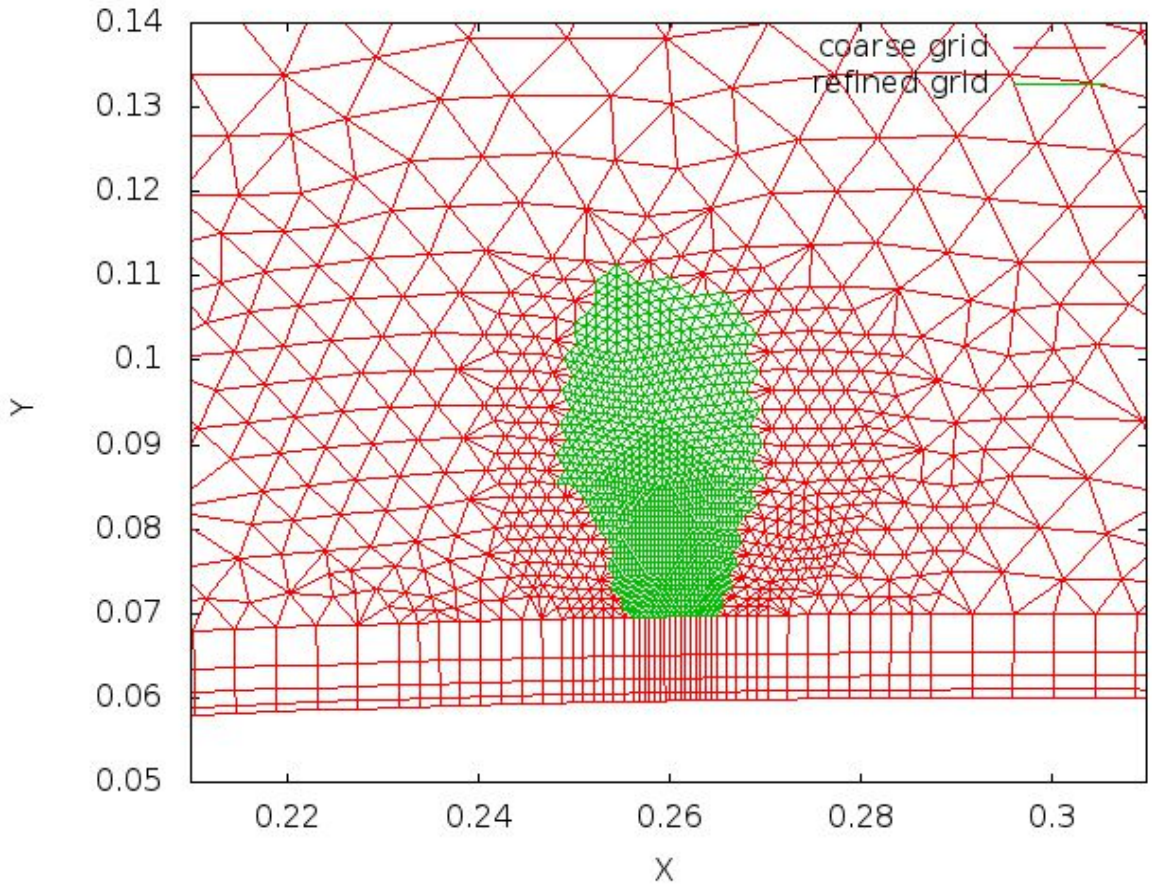


Fig.18: Grid after 3rd refinement, 13020 elements, ,Ma_∞=0.7 ,angle =2˚,Naca0012.

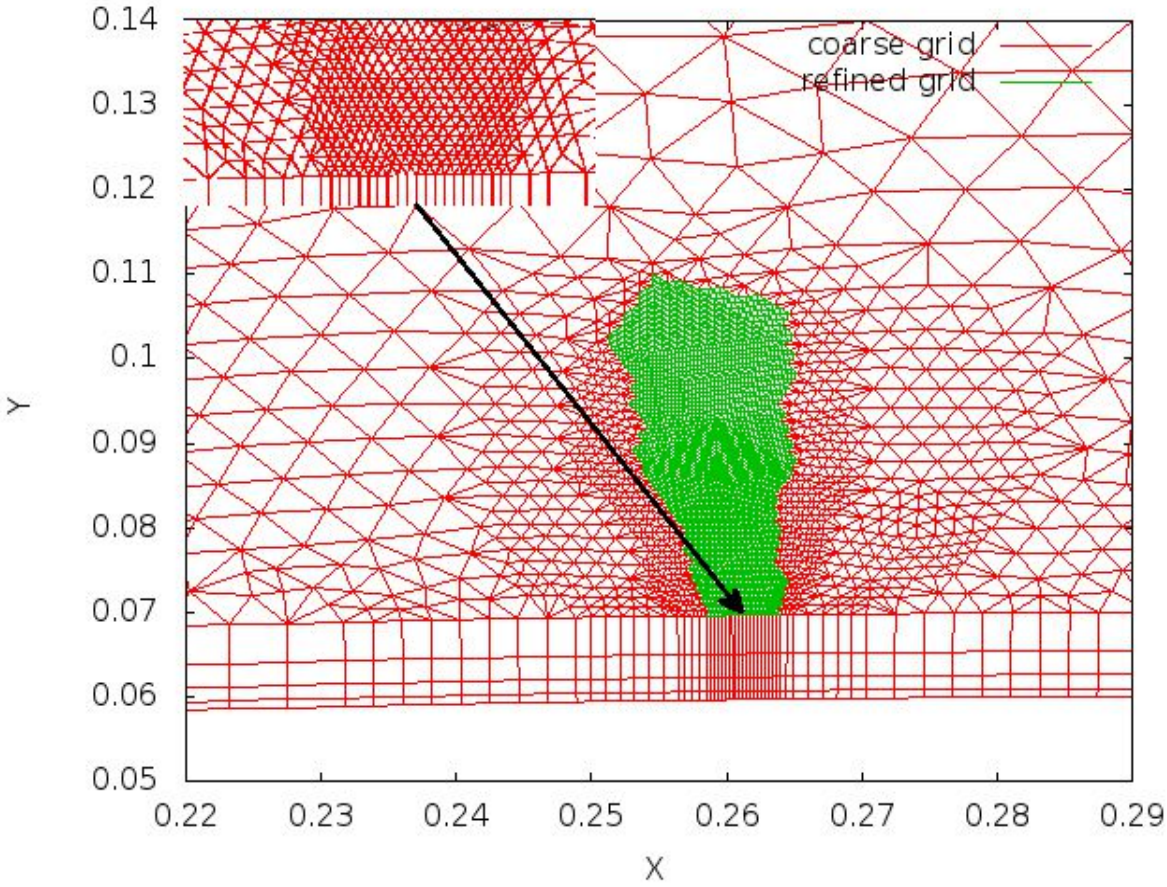


Fig. 19: Grid after 4th refinement, 15920 elements, Ma_∞=0.7, angle =2˚, Naca0012.

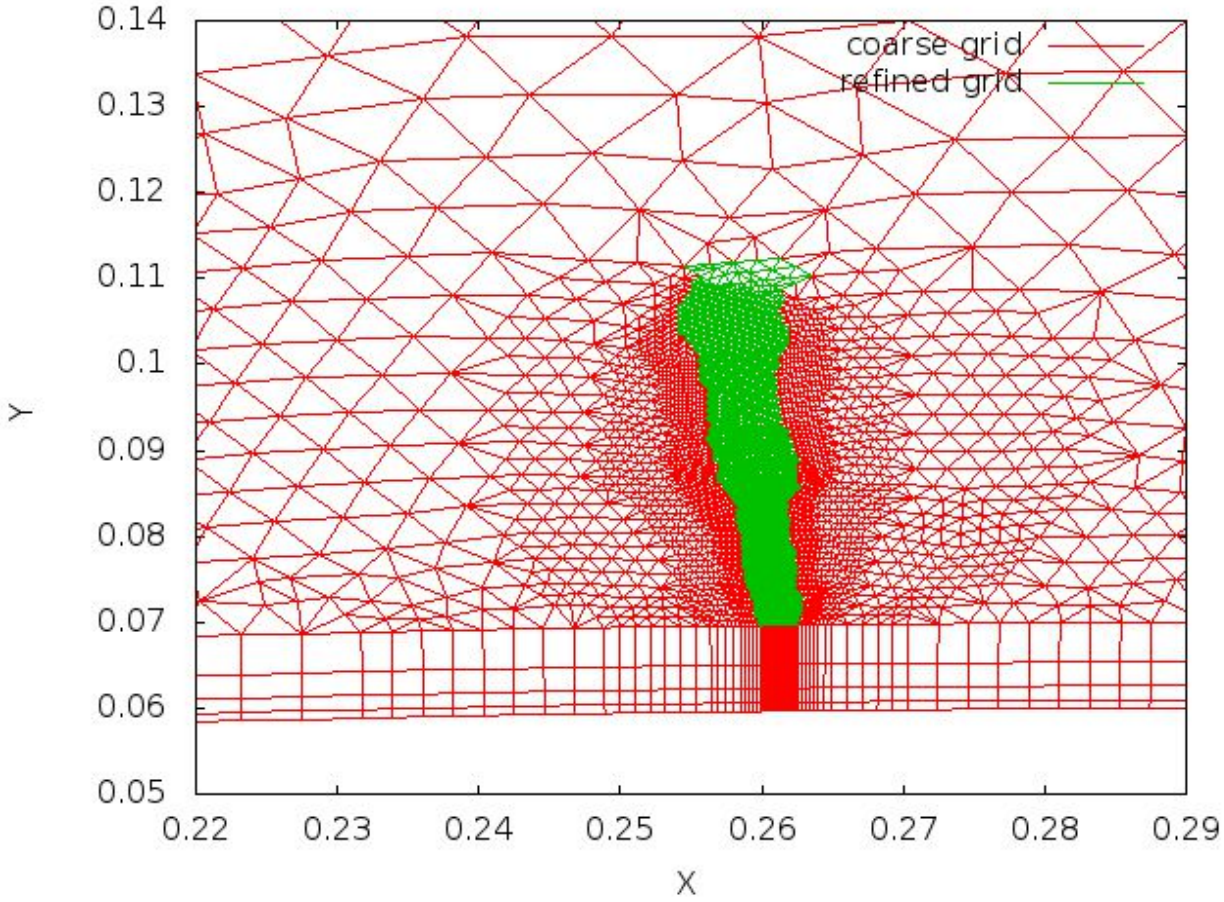


Fig.20: Grid after 5th refinements (21976 elements), Ma_∞=0.7, angle =2˚, Naca0012.

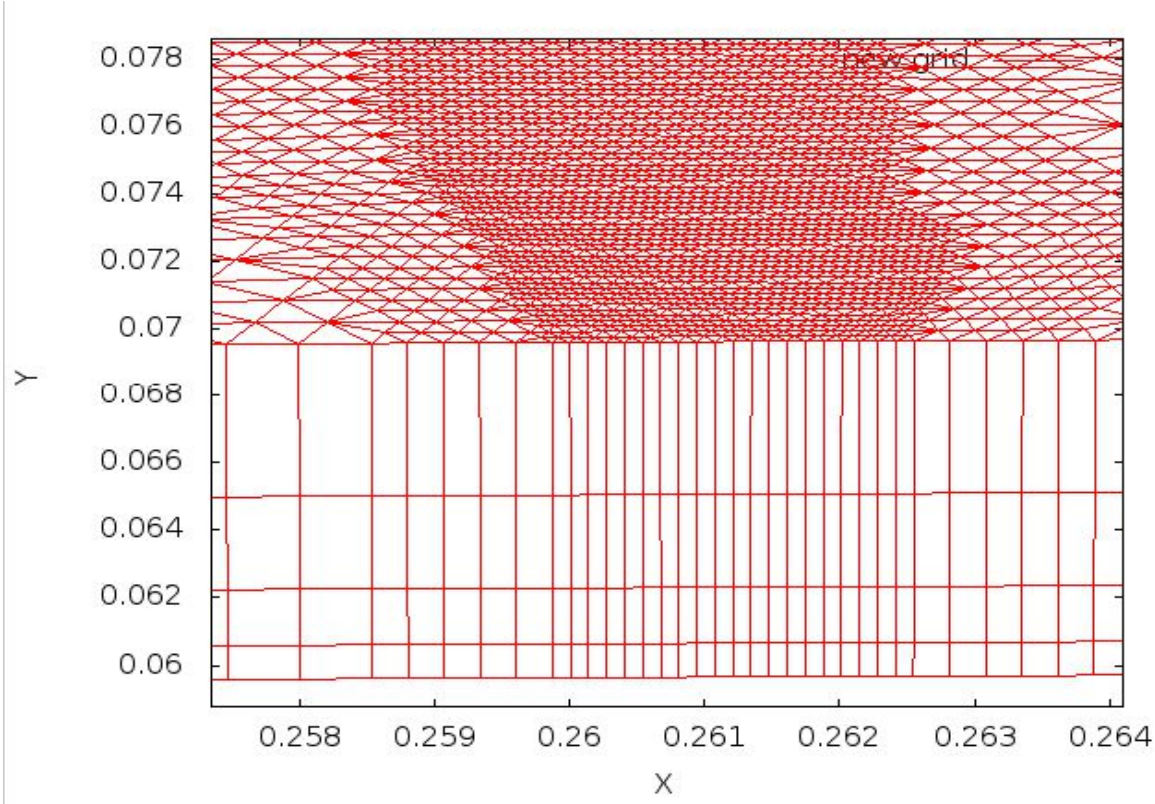


Fig. 21: Closer look at refined grid after the 5th refinement, Ma_∞ =0.7, angle=2.0˚.

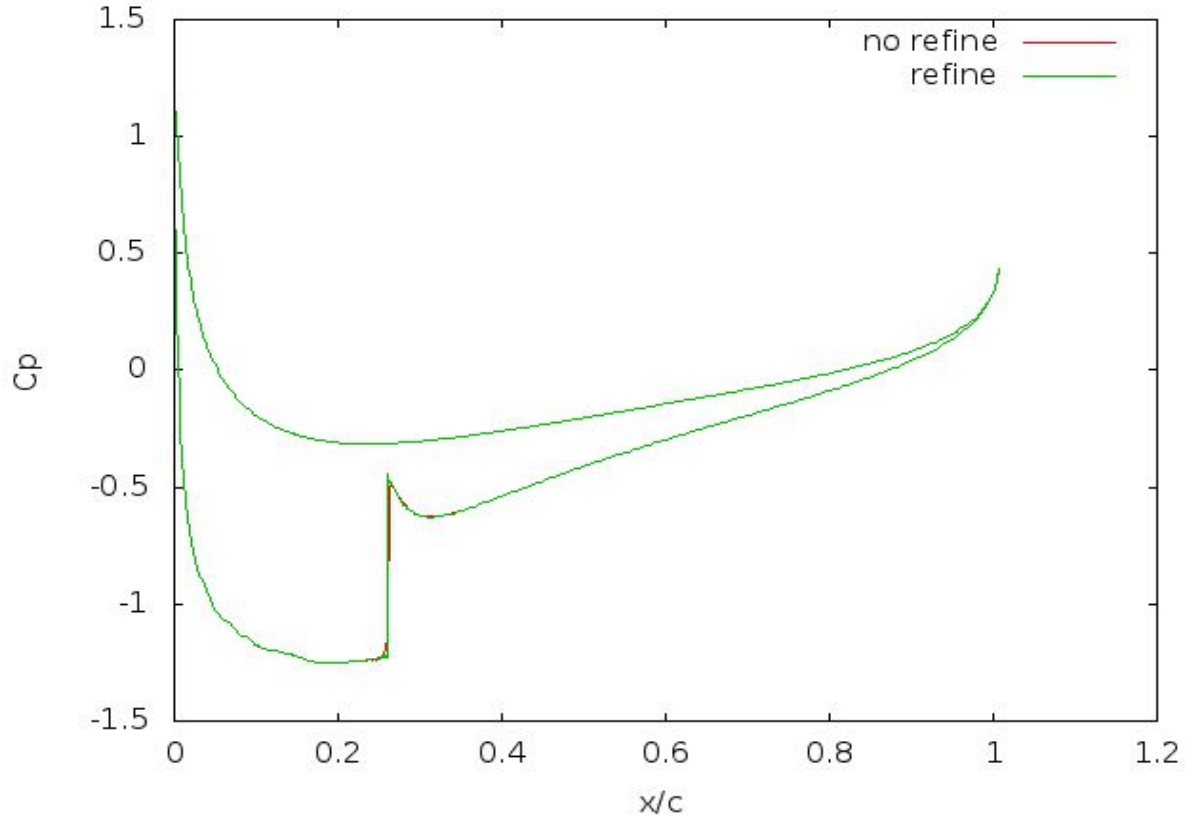


Fig.22: Comparison of C_p coefficient for coarse and refined grid for NACA0012-Mach 0.7 angle =2.0˚.

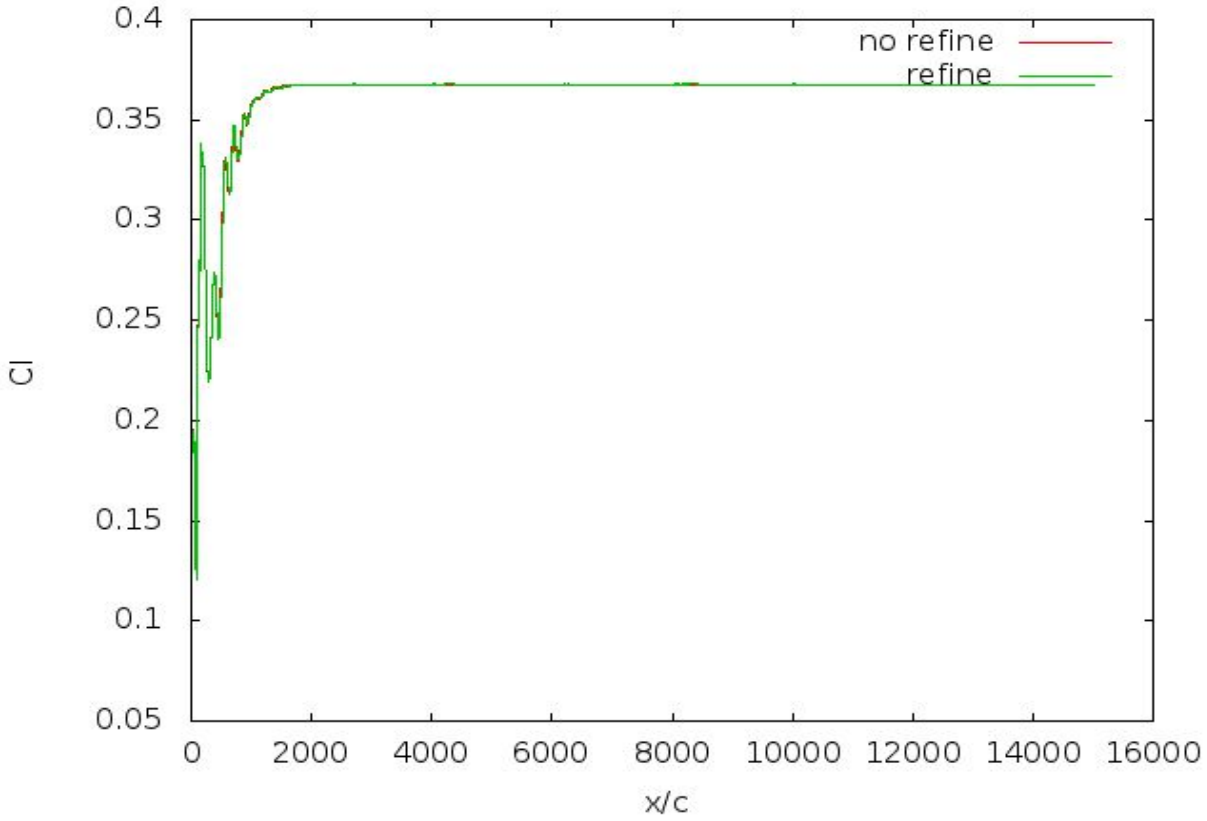


Fig.23: Comparison of C_L for the coarse and the refined grid with respect to convergence history -Ma =0.7 angle =2.0˚, Naca0012.

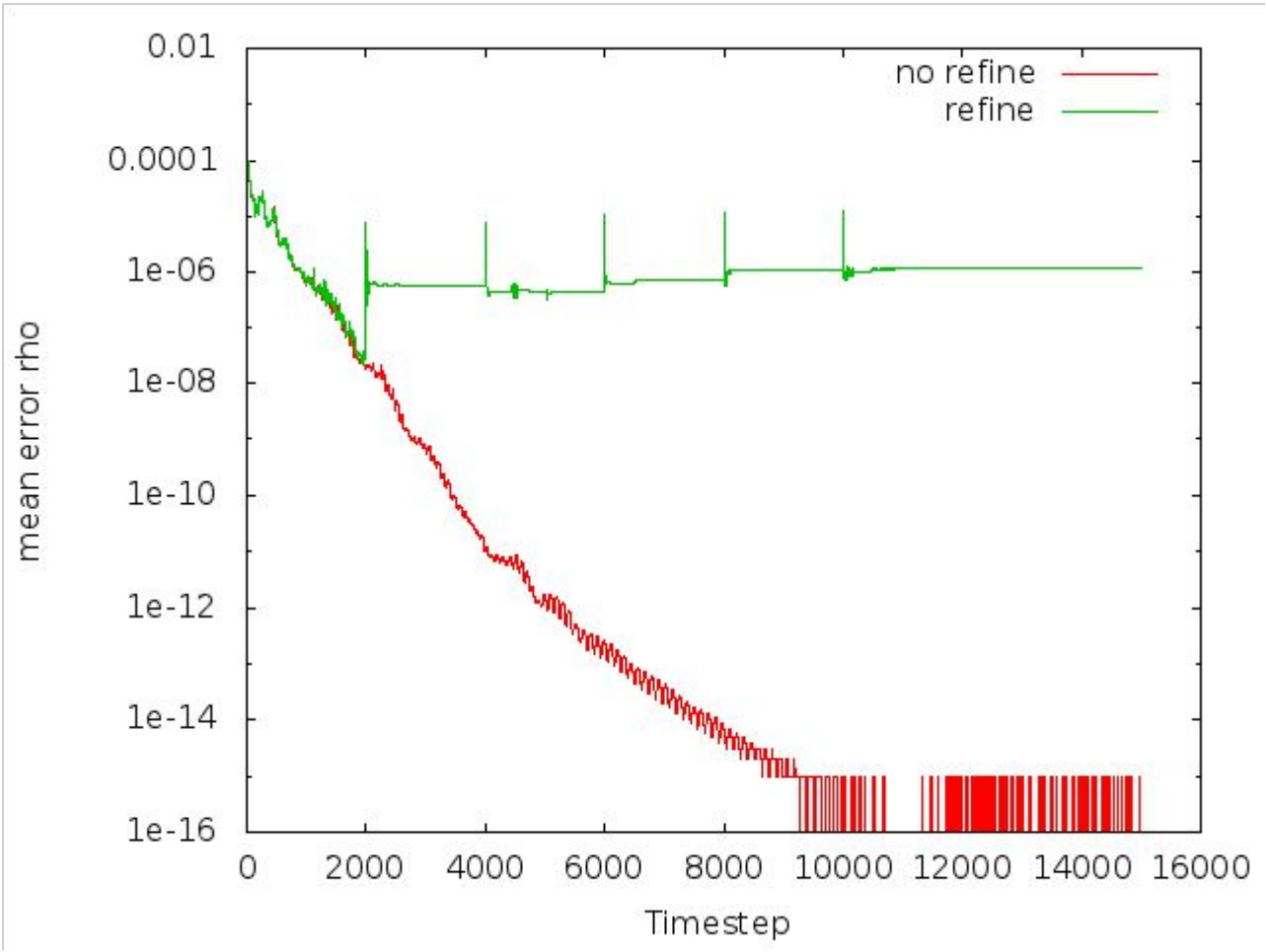


Fig.24 : Mean error of density for both Coarse-Refined grid ,Ma_∞=0.7 ,angle=2.0˚,Naca0012.

In the case of hybrid mesh the criterion of absolute Mach difference remains the same, at 0.12 and the limit for the criterion of mean error at each cell becomes 10d-02. The refinement area is figured in figures 25-26. We may observe that:

– Pressure and lift coefficient figures (fig. 22-23), remain the almost the same (refined and non refine cases). There are no wiggles at the pressure because of the structured layer. Moreover the mean error of density (fig. 24) increases to the levels of 10e-06 reaching a value close to 1.21033e-06 after the last refinement.

– The area of the shock wave is shown in figures 27a)-b). Again there exist sub regions with high values of energy error (fig. 28a). The maximum error is close to 0.10.

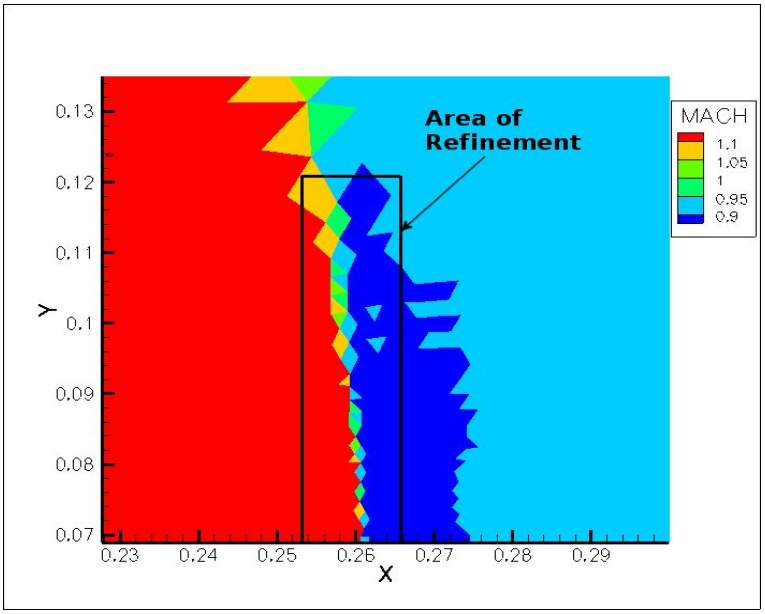


Fig.25:Area of refinement –Mach =0.7 ,angle =2.0˚.

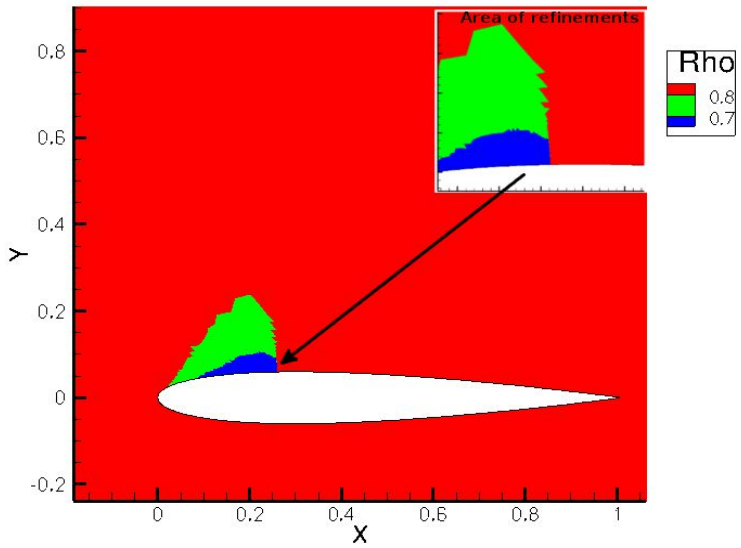


Fig.26: Shock Wave over Naca0012, Ma∞=0.7, angle=2.0˚

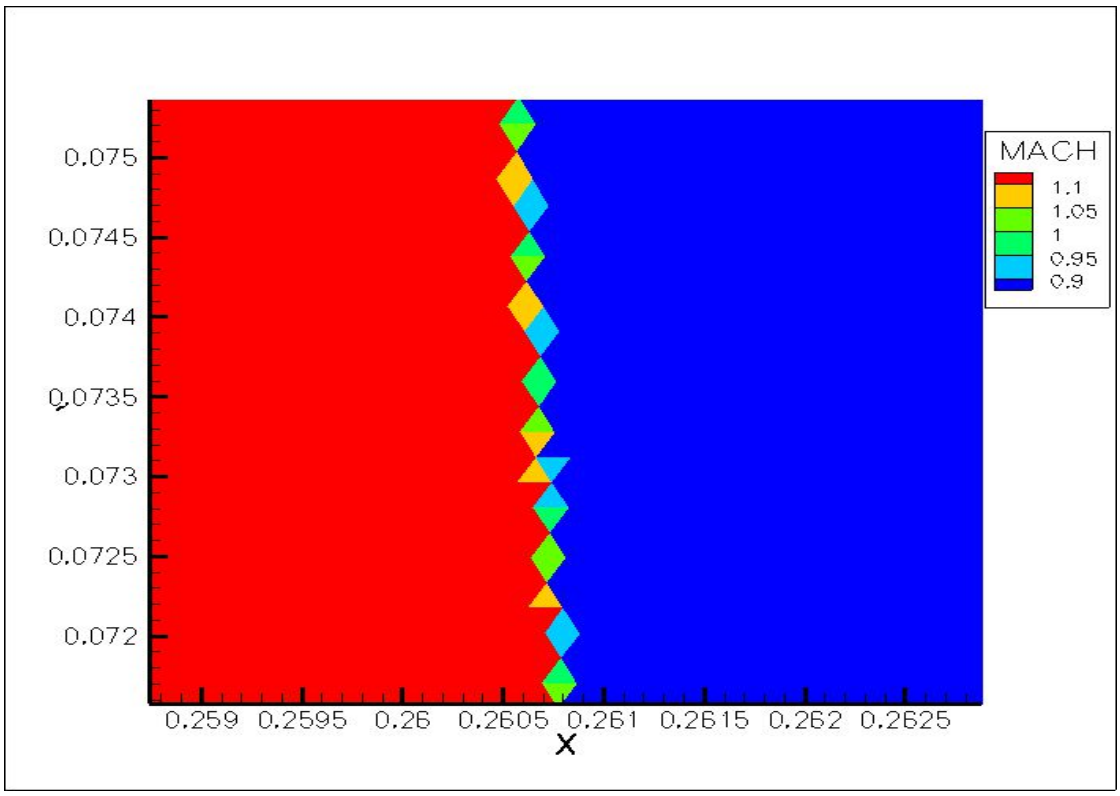


Fig.27a: Shock wave after the refinement. The red area (supersonic) is even closer with the blue one (subsonic).

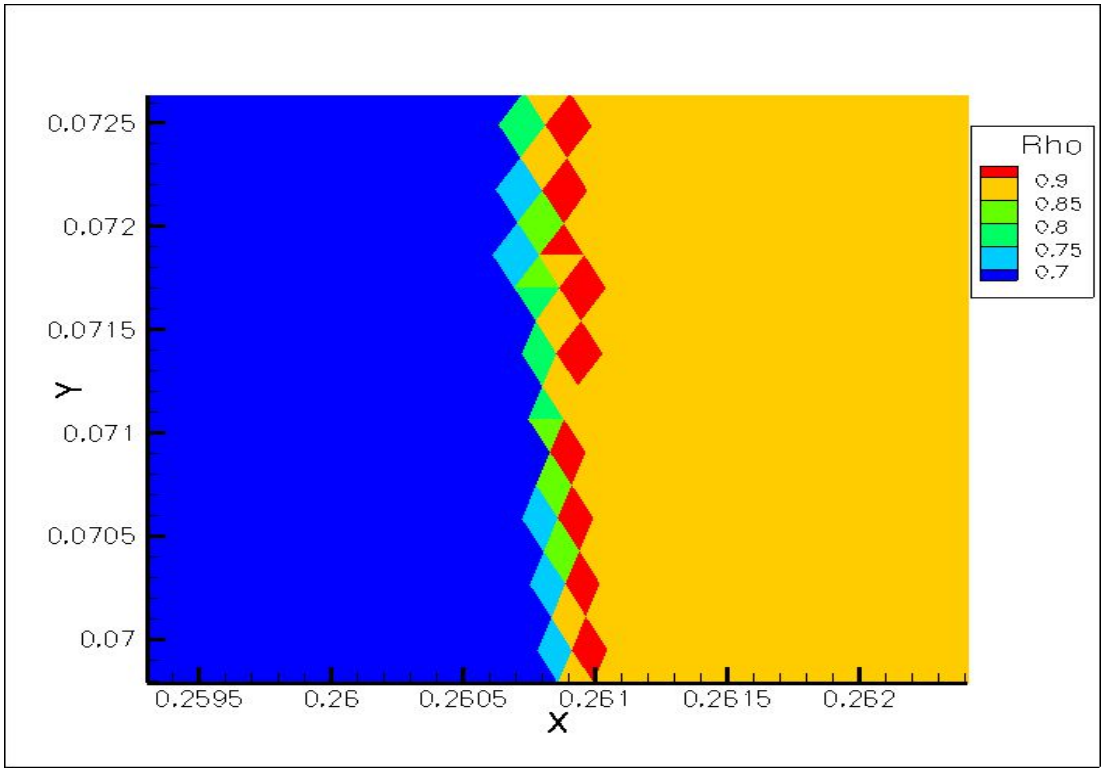


Fig 27b: Density at an area of big mean error of energy, Ma_∞=0.7, angle=2.0˚.

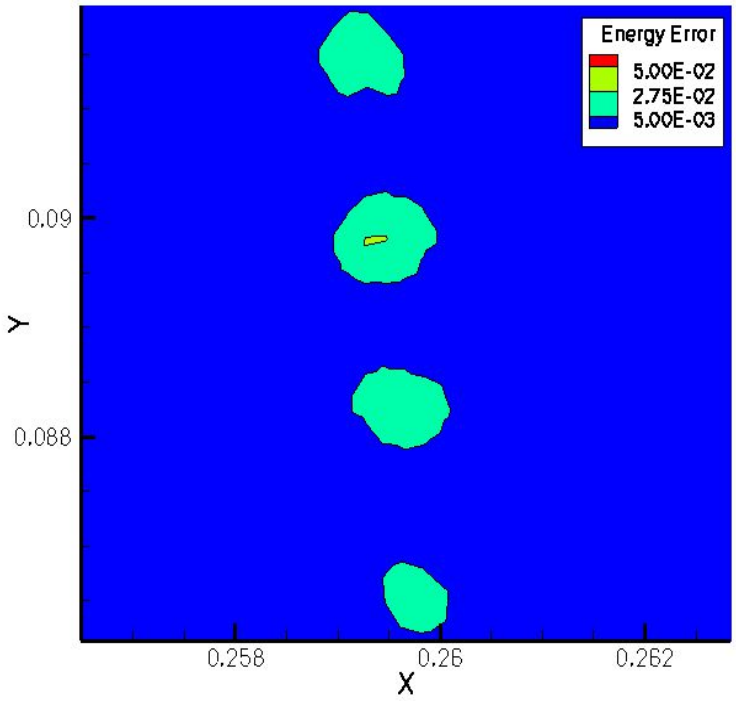


Fig.28a: Typical example of an area with large mean error of energy. These sub-regions meet across the shock wave increasing mean error of density.

If the refinement solver is called at a time step, where the convergence is even more greater (with even lower mean error) the result will be the same: 5 levels of refinement with the figures of C_p ,C_L the same as the above case. The mean error of density in this case is shown below (fig. 28b).

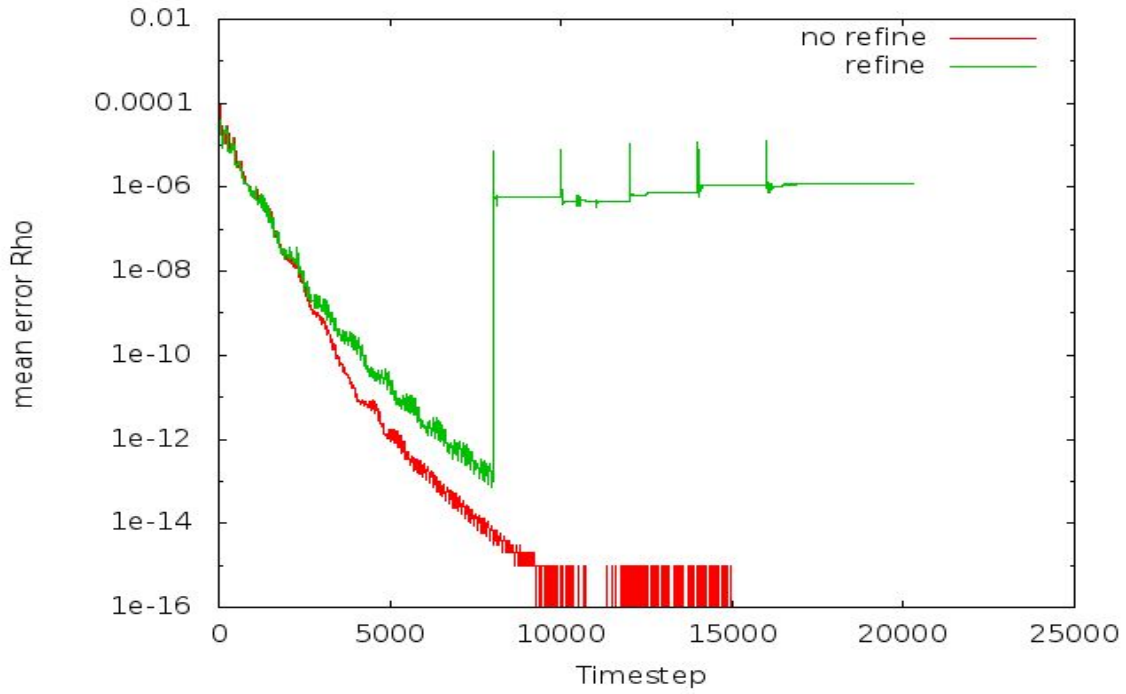


Fig.28b: Mean error of density through convergence history, case Ma_∞ =0.7, angle =2.0˚.

4.2 Transonic flow, Ma_∞= 0.85, angle of inflow = 1.25°.

The criteria for the refinement were the absolute difference of mach number (equal to 0.13) and the mean energy error at each cell (equal here to 1e-01). The final CFL was also a parameter that varied. At the case was equal to 5. Number of cells at the initial unstructured mesh = 3658.
In this case we have inviscid flow around Naca0012 with Mach number 0.85 and angle of flow 1.25˚. Two shock waves appear (fig. 29a-b) almost perpendicular to the contour of the airfoil. The initial coarse grid is refined in two different areas where the criterion points to.

Looking the levels of refinement we may observe that the majority of the new cells is fixed at the strong shock wave. This seems fair as both to shocks are refined by the same quantity-criterion.

In order to avoid the large scale refinement, a stricter value for the bandwidth of refinement was used. This is about the portion of the cells that will be divided by two and is also given as input at the solver. In order to avoid the large scale refinement, a stricter value for the bandwidth of refinement was used. This is about the portion of the cells that will be divided by two and is also given as input at the solver.

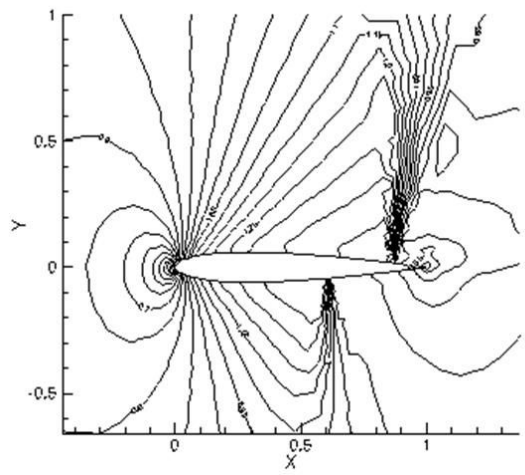


Fig.29a: Mach lines for inviscid flow Ma =0.85.

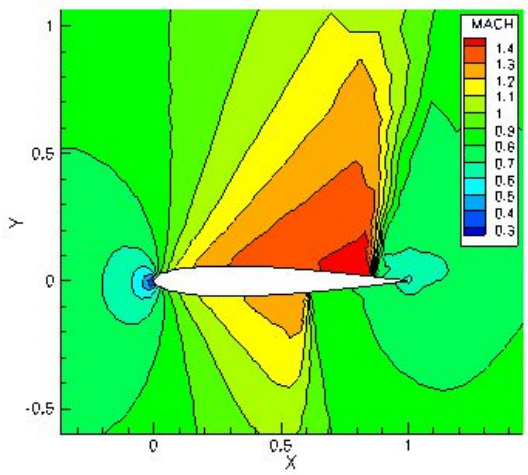


Fig.29b: Mach distribution over Naca0012.

The mesh changes follow below. The refinement is accomplished both on the pressure and suction side, where the absolute difference of mach number is quite large. This limit for the criterion was defined after testing multiple values. As seen at figures 31-35 the transition from the coarse to the refined part is a serious issue. Some geometric aspects were fixed by the solver's ability to check the neighbor's angles and volumes as mentioned at chapter 3 and the limits of the number of cells at the two new cells area and the number of the four new cells area. So the smoothness from coarse to finer grid is also taken care.

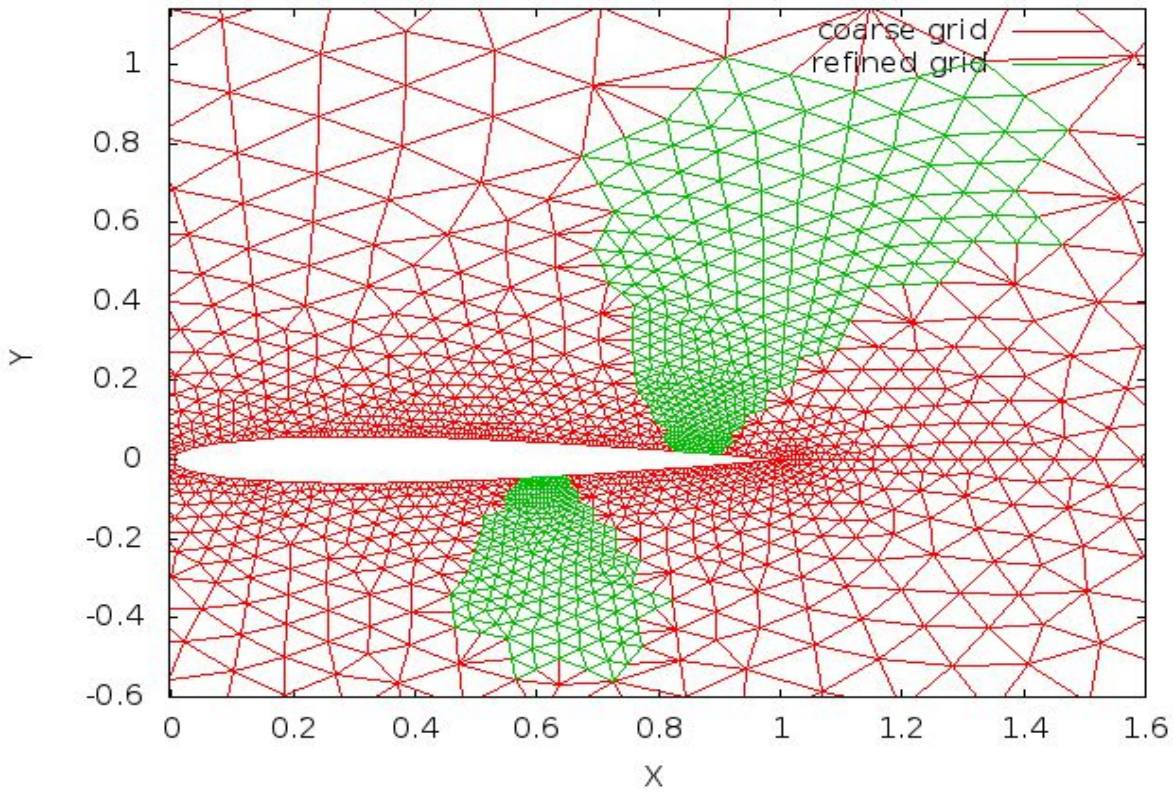


Fig. 30: 1st refinement, 4508 elements, Ma_∞ =0.85, angle =0.125˚.

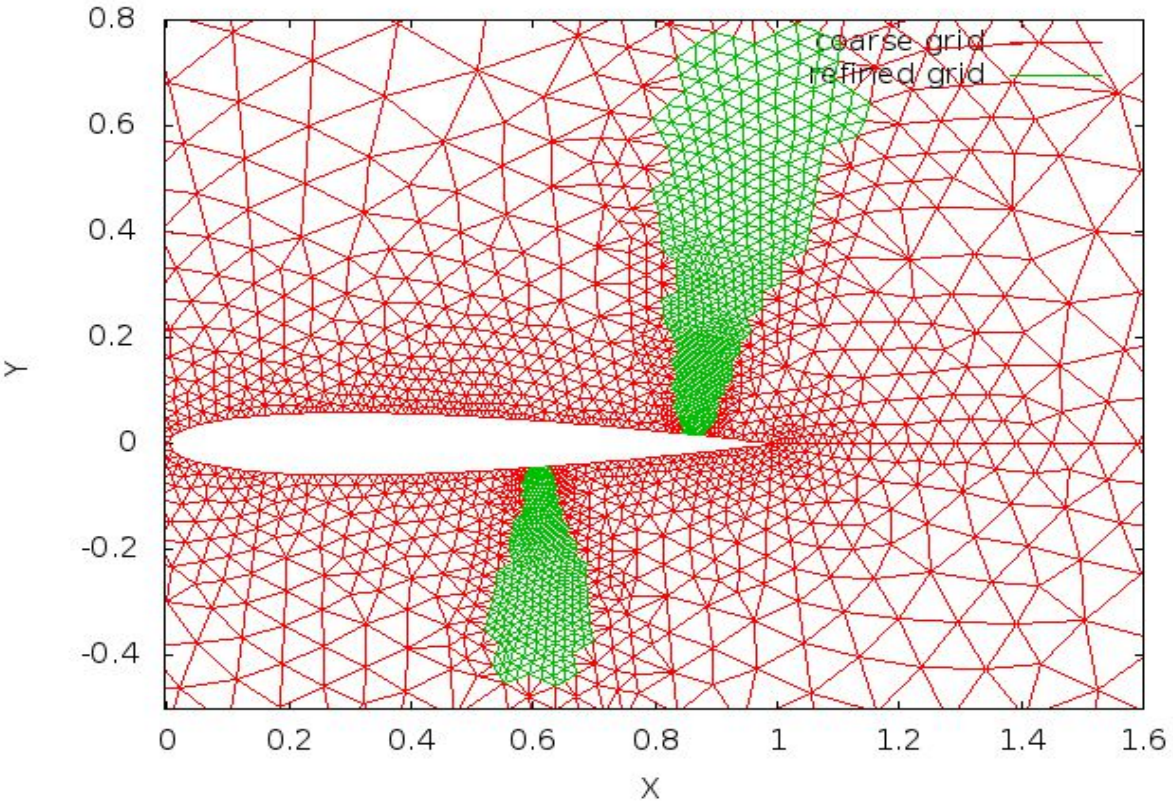


Fig 31: 2nd refinement, 6087 elements, Ma_∞ =0.85, angle =1.25˚.

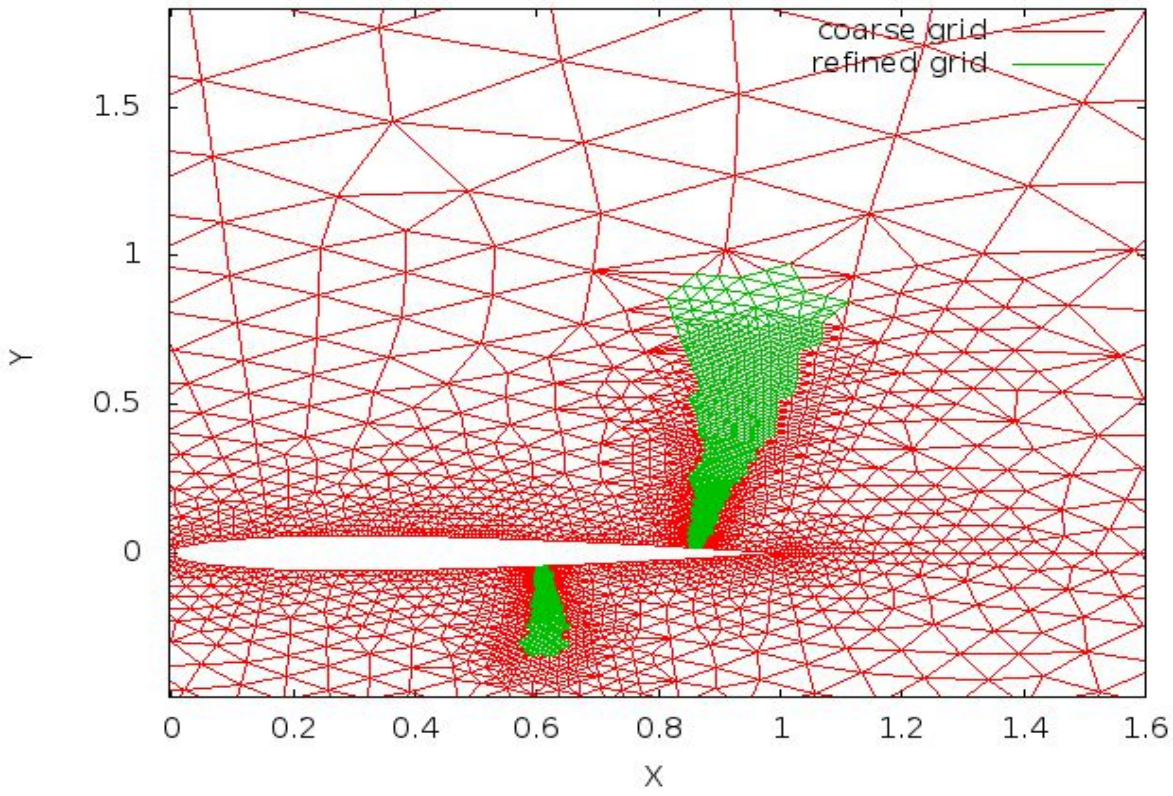


Fig 32: 3rd refinement, 9131 elements, Ma_∞ =0.85, angle =1.25˚.

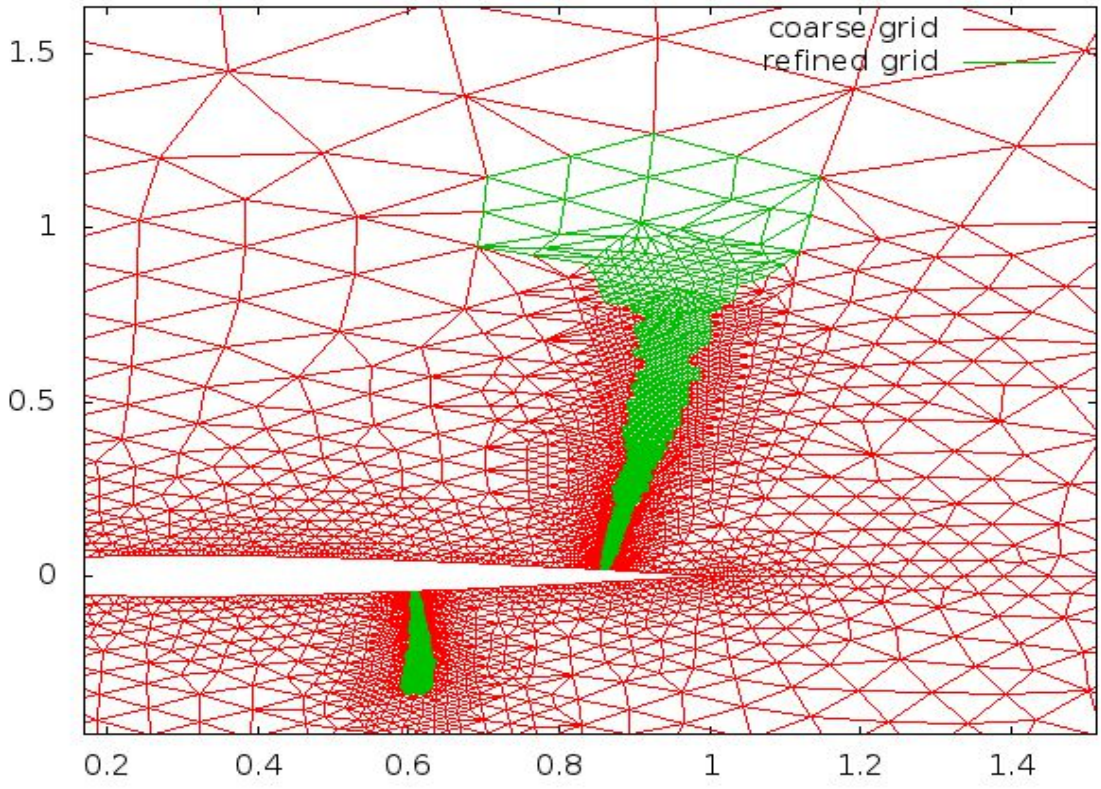


Fig. 33: 4th refinement, 15264 elements, Ma_∞ = 0.85, angle =1.25˚.

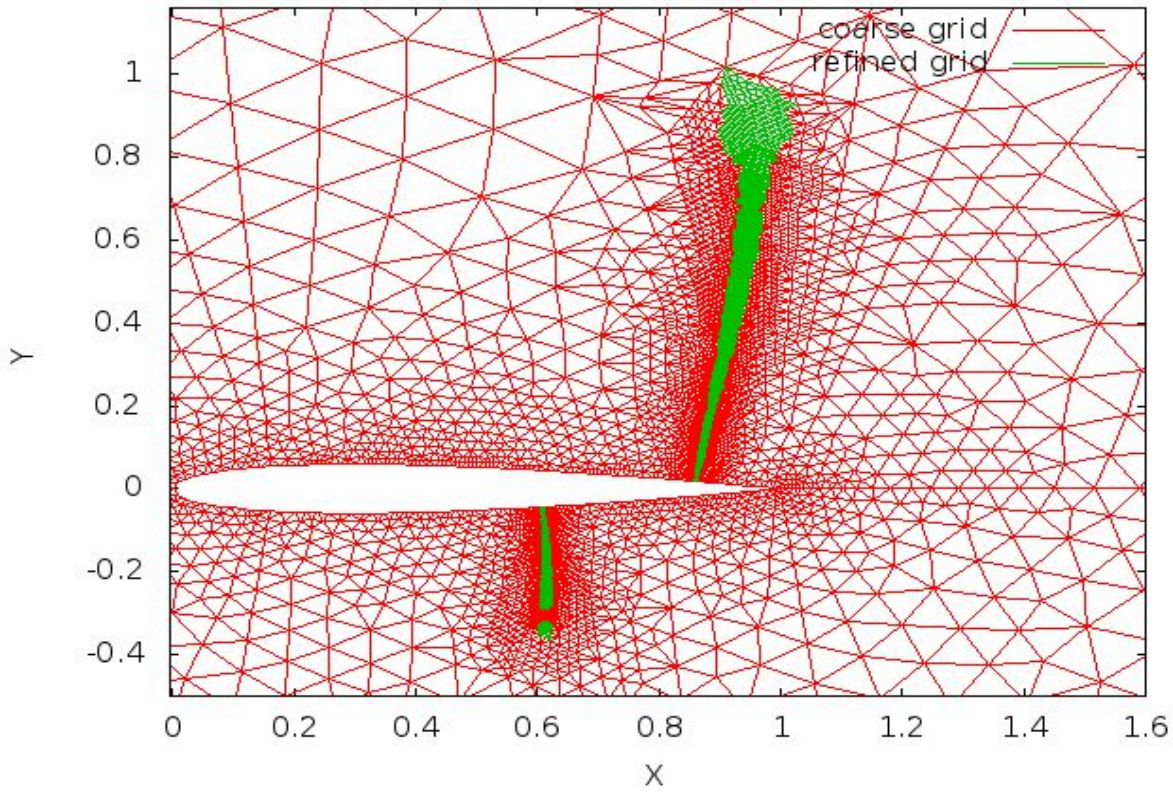


Fig. 34: 5th refinement, 27067 elements, Ma_∞ =0.85, angle =1.25˚.

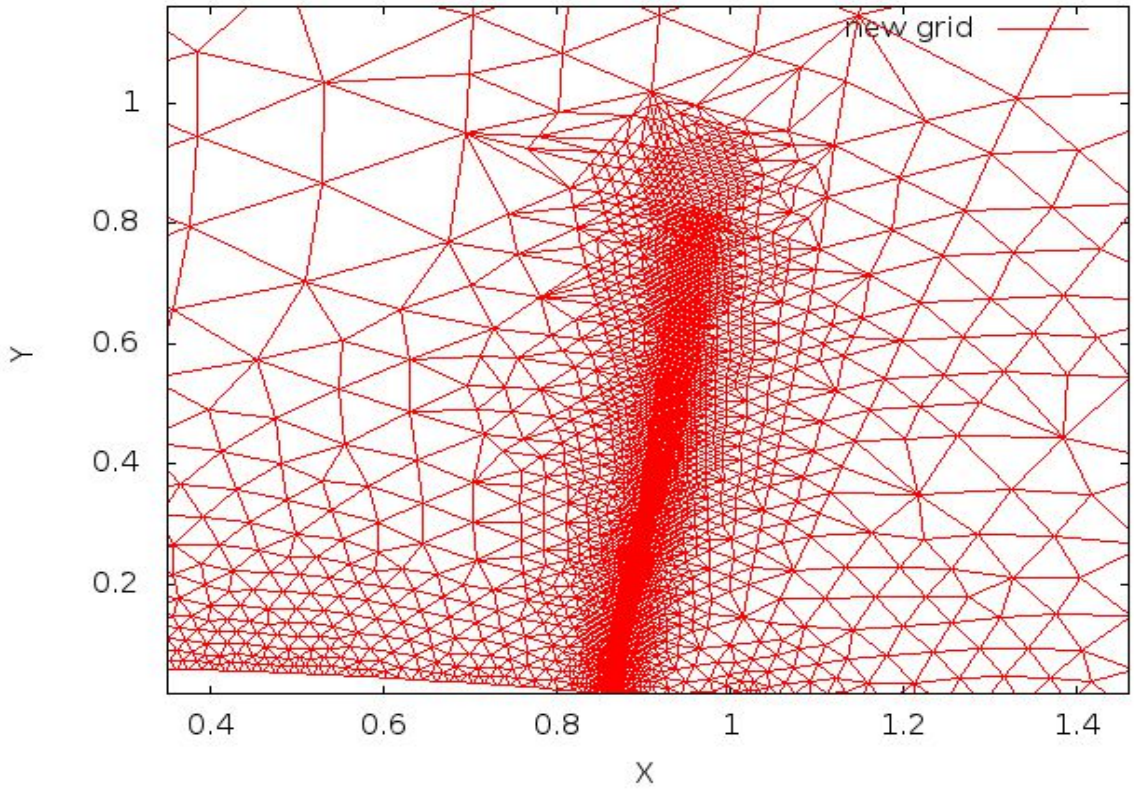


Fig.35: Area of refinement at big shock wave, Ma_∞ =0.85, angle =1.25˚.

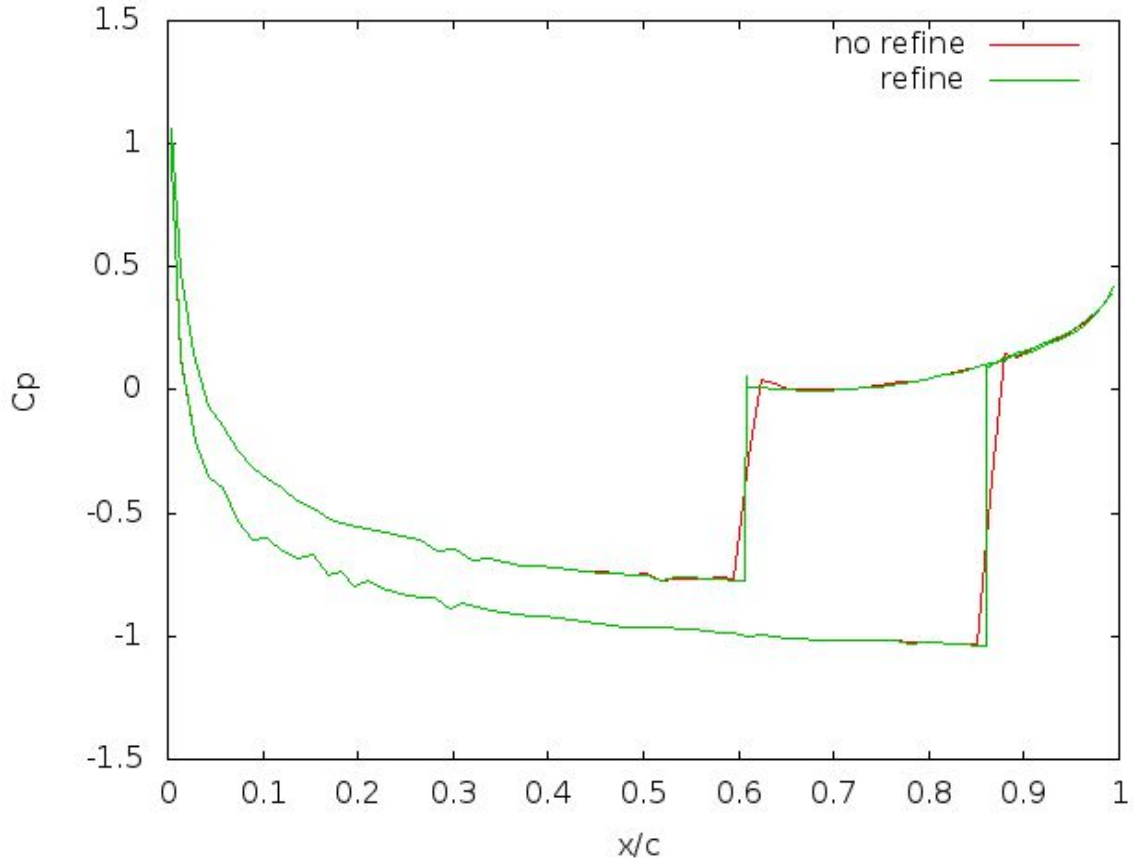


Fig. 36: Pressure coefficient over Naca0012, Ma_∞ =0.85, angle =1.25˚.

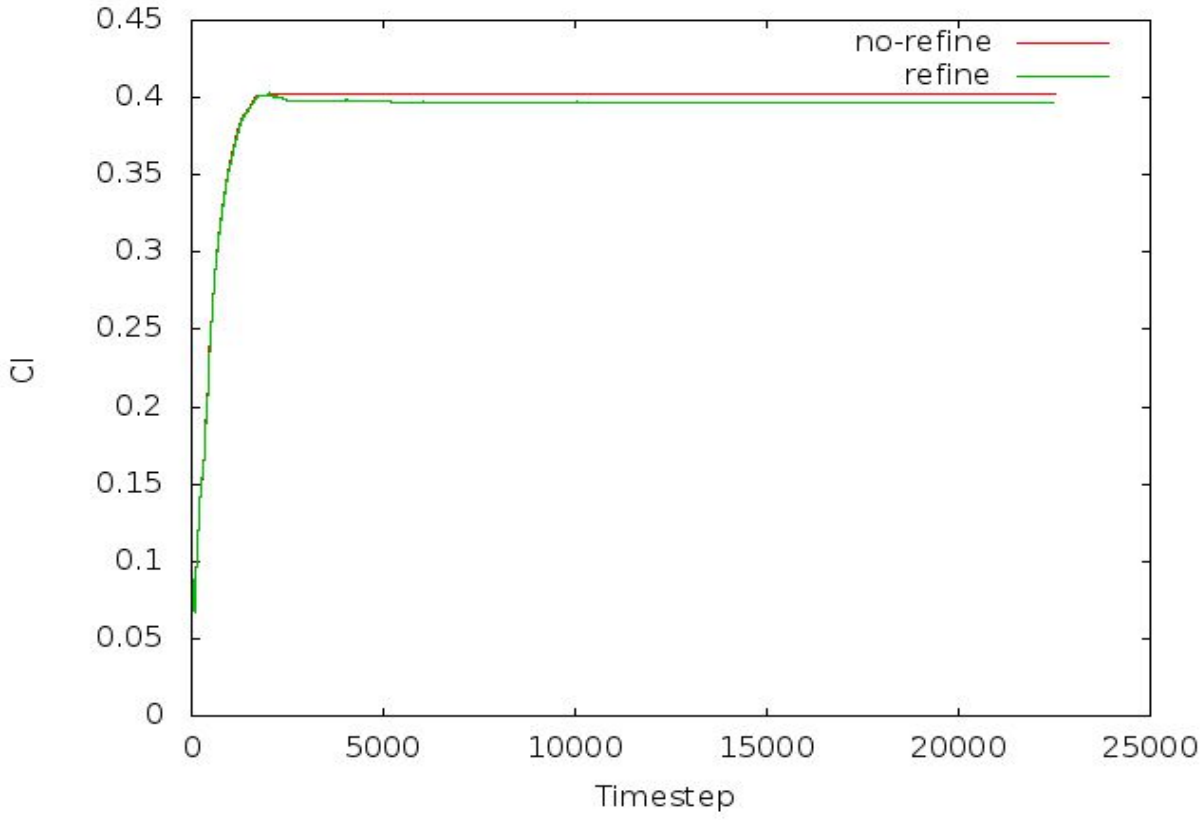


Fig. 37: Lift coefficient over Naca0012, through convergence history. Ma_∞ =0.85, angle =1.25˚.

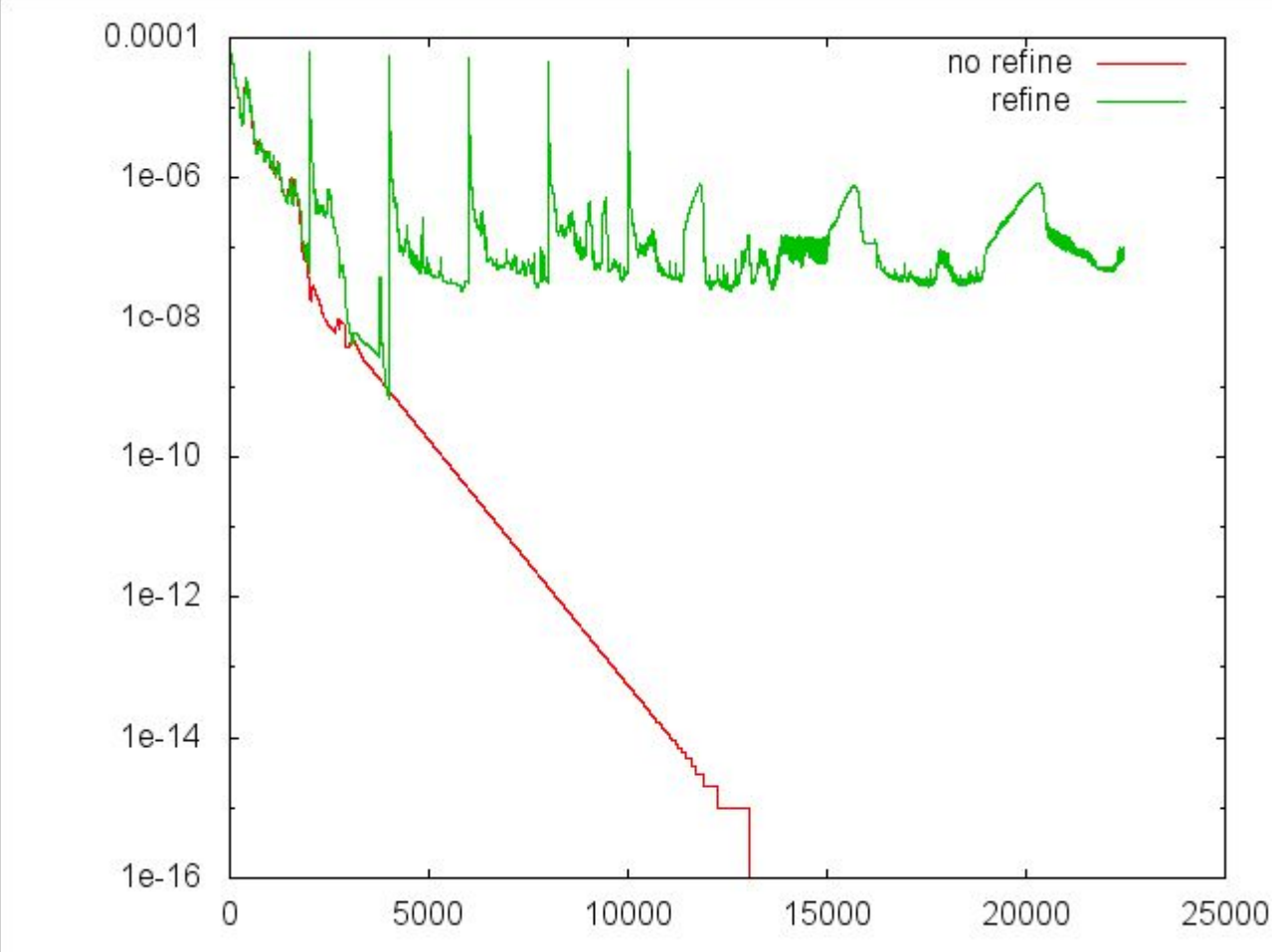


Fig. 38: Mean error of density, Ma_∞ =0.85, angle =1.25˚.

OBSARVATIONS

– For the pressure, the wiggles that occur at the triangle meshes, still exist here (fig. 36) and as mentioned before this (unpleasant) behavior is common at unstructured cases due to the cell's shape. Pressure coefficient C_p seems to improve after the refinement and shock waves are perfectly mapped at the pressure plot.

– Lift coefficient C_L diagram normally changes and convergences at a value a little lower than 0.4029 of the non refined mesh.

– As for the mean error of density, does not keep the low levels of congruence in the non refined case (fig. 38) and approximates a mean value around 5 to 8 10e-08. This big mean error is the result of cells with high error. Moreover an extra limitation was the number of cells for the next refinement of a mesh (less than 20.000 cells). The final mesh has 27067 cells (7.38 times larger than the initial). In figure 39 one can see where the mesh changes, and how "the criterion draws the path of refinement". At figure 40 there some captions of the shock wave over the upper side with the 6 (1+5) meshes. There exist regions where the mean error of energy is significantly big compared to the neighboring. The cells in these regions contribute to the high maximum error at both (fig. 41a,b,c,d). Comparing a caption before the forth refinement and afterwards (fig. 41 d) we can observe the revealed detail. At the next caption (every caption is in 1000 time steps) the error is smaller. Figure 41 is repeated in a way similar to the behavior the mean error of density.

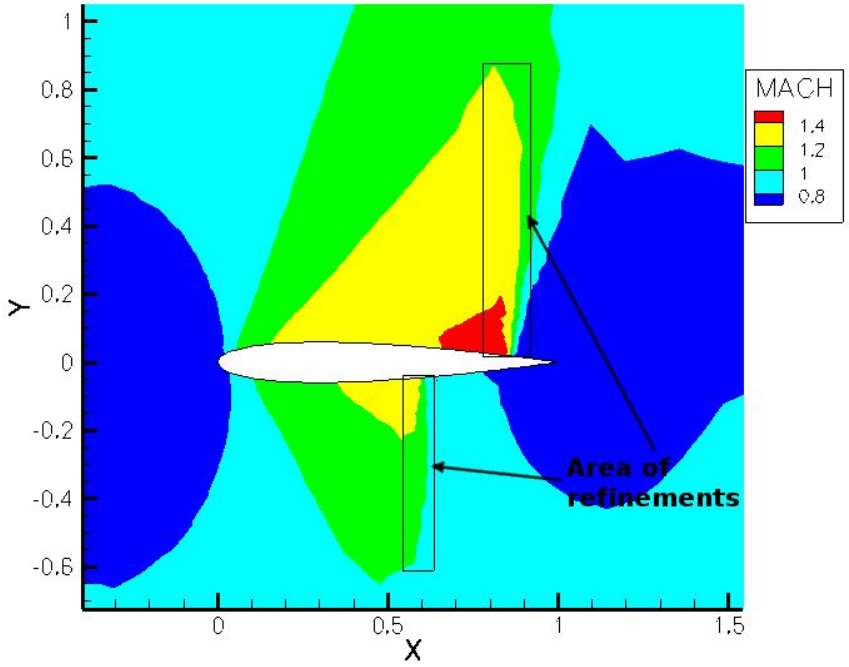


Fig.39:Area of refinement for Ma∞ =0.85, angle 1.25˚

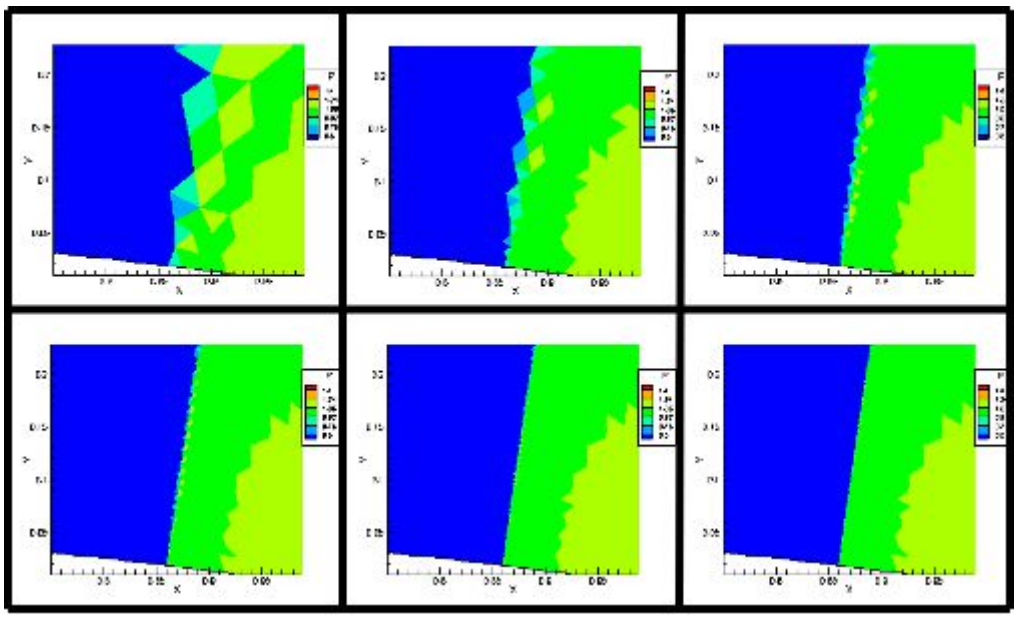


Fig. 40 : Pressure capture through the refinements.

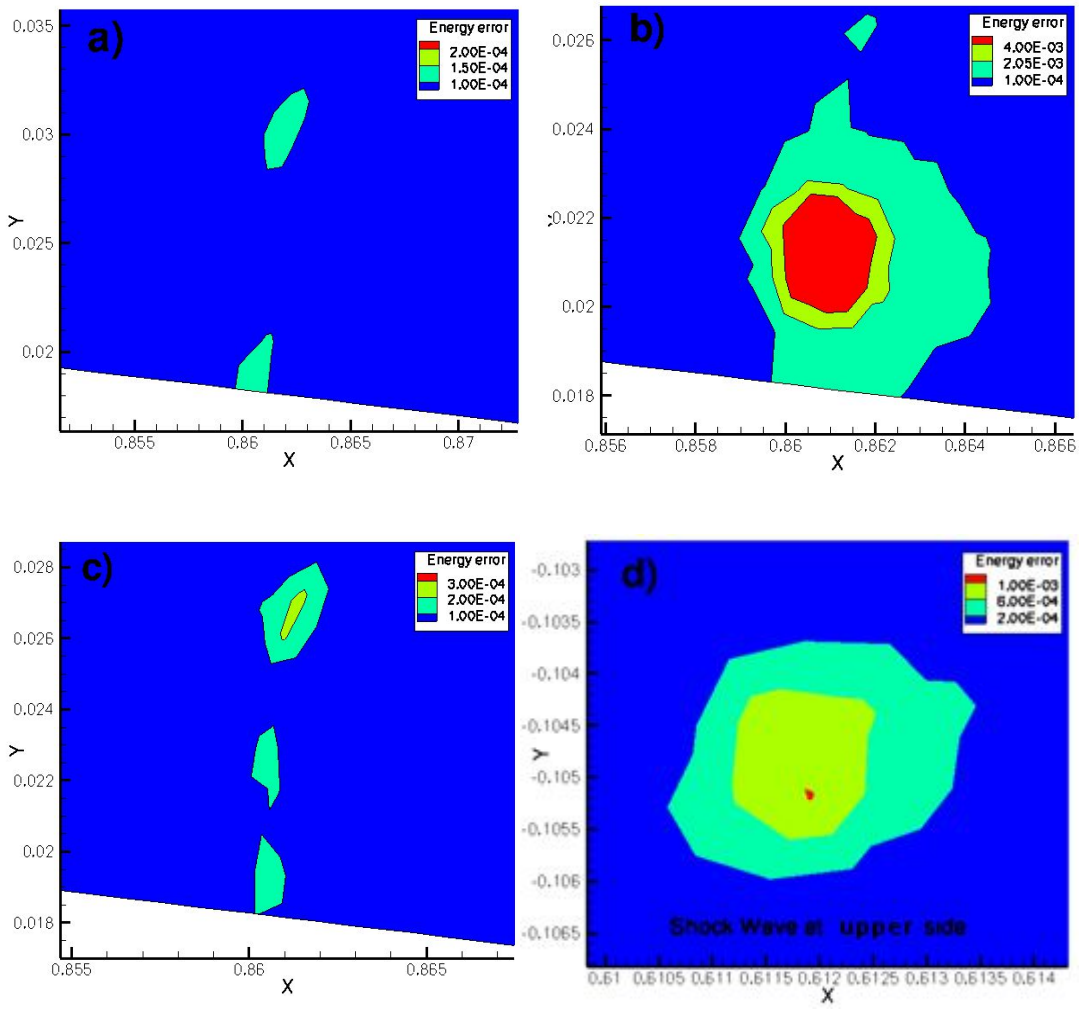


Fig.41: Mean error of energy per cell a) before 4th refinement ,b)after 1000 time steps and c)after 2000 time steps, d) after 3000 time steps Ma∞ =0.85 ,angle =1.25.˚

Above are figured the captions of the maximum energy error of energy. Small regions across the shock waves may have these variations, and this is just a typical example. As before after the repeated refinements the maximum error which is revealed in these areas, increases the error levels of the whole mesh. These refinement areas have large mean error of energy with respect to the rest mesh.

This observation leads us to the question: how is the number of cells can affect the convergence levels? A stricter criterion for the number of cells of the coarse mesh each time the refinement solver was called. Decreasing this number and applying one level of refinement less each time we can see the congruence behavior. The relation of the number of cells and the mean error of density for the same criterion of refinement is figured below (fig. 42):

Number of Refinements	Final Number of tri-elements
3	9131
4	15264
5	27067

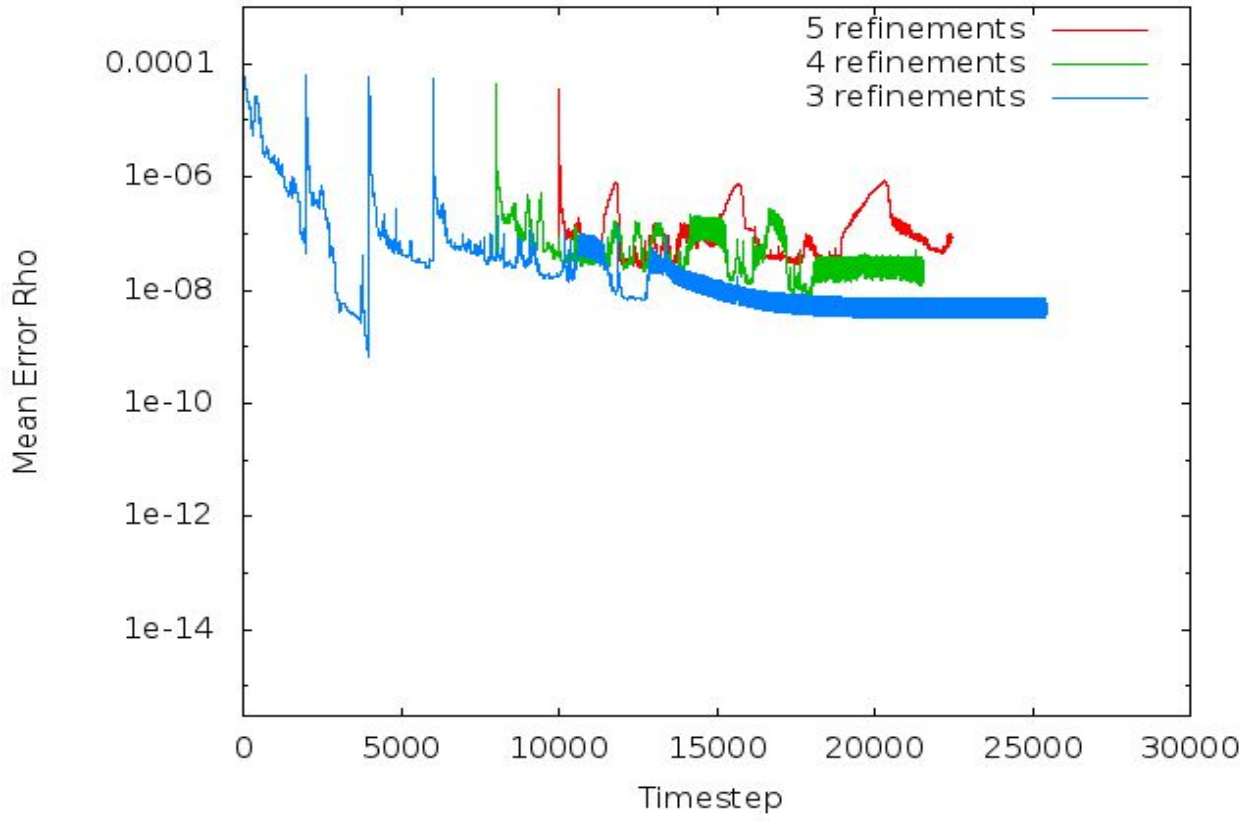


Fig. 42: Same error indicator and Mach criterion with 3, 4 and 5 refinements respectively, Ma_∞ =0.85, angle =1.25˚.

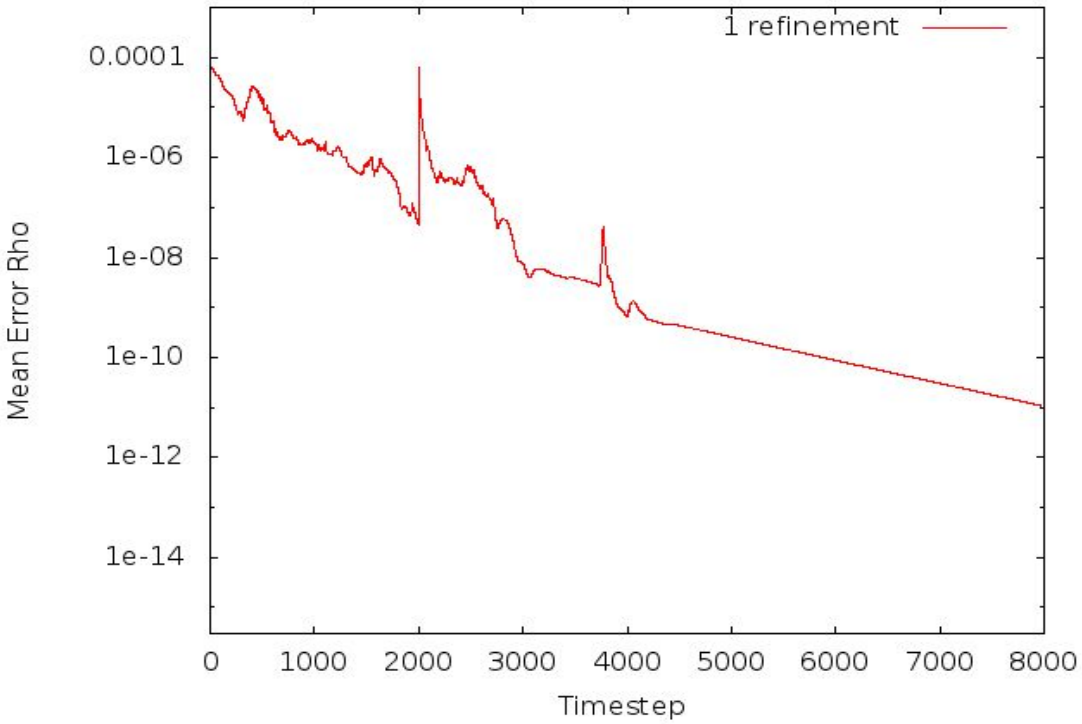


Fig. 43: 1 refinement, mean error of density is significantly small. This is an example of one Single case not mentioned further.

As we can see in picture 42, applying 2 refinements less we can reduce the mean density error 10 times smaller, and with one refinement we can reach at the same levels with the non refined mesh (fig 43). The same happens is we relax the criterion by increasing the mach number from 0.13 to 0.12 (fig. 44).

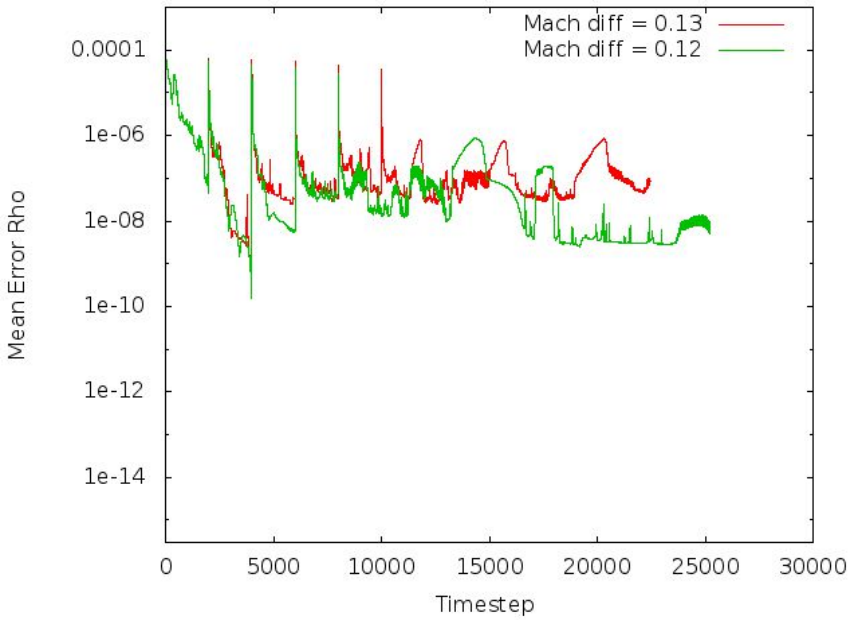


Fig. 44: Changing Mach number limit gives 4 refinements and even lower mean rho error ,Ma_∞ =0.85 ,Angle =1.25˚.

TESTING VARIUS REFINEMENT CRITERIA

Generally, the criterion of the absolute difference of mach number turned to be the most efficient for shock waves helping us to determine where the solver must be apply the mesh changes. Although others can also be tested. One can use the square root of the divergence $div\vec{p}$ where except for the shock's areas, some other sub regions are selected too like the stagnation point (fig. 45).

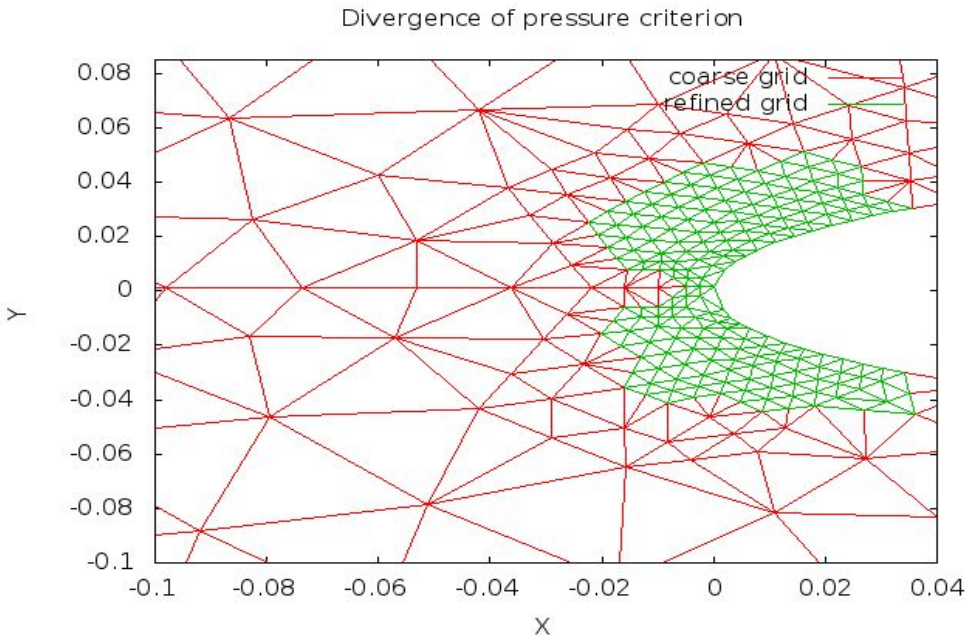


Fig. 45: 2nd refined grid, divergence of pressure criterion, at stagnation point, Ma_∞ =0.85, angle =1.25˚.

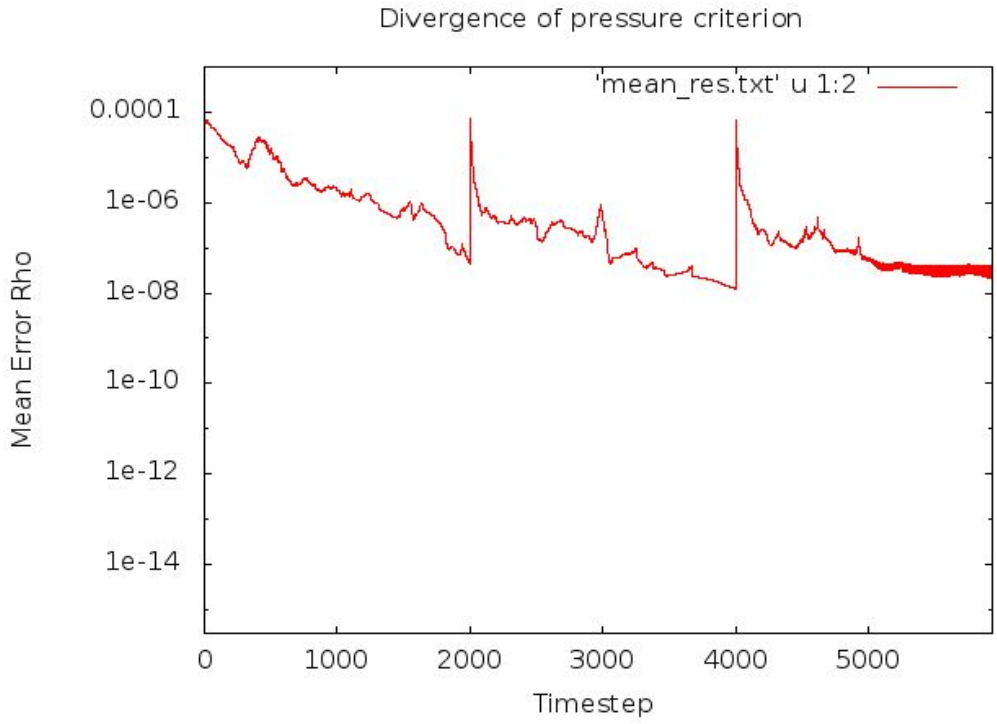


Fig.46: Mean error of density after 2 refinements, Ma_∞=0.85, angle =1.25˚, Naca0012.

– **Inviscid flow, Ma_∞=0.85, angle of inflow =1.25˚, with finer initial grid.**

The test case of mach 0.85, was promising as concerned the convergence so a initial coarse unstructured mesh with more cells was used. This mesh has 13586 triangle cells and the refinements are below (fig.47-48-49-50).

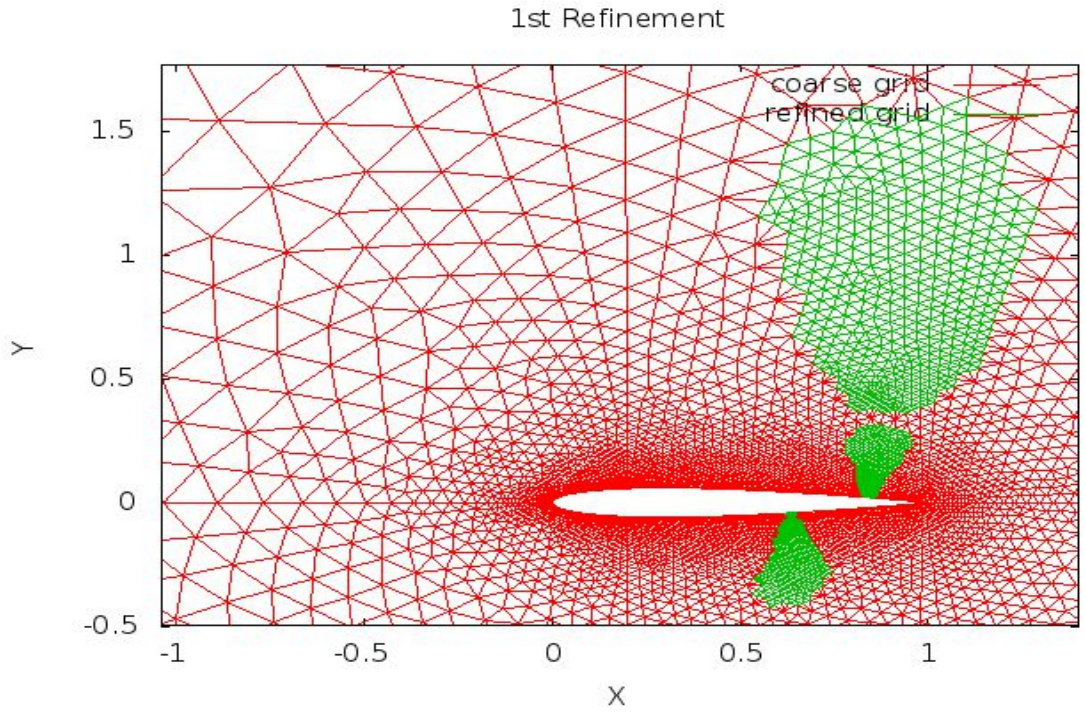


Fig. 47: 1st Refinement, grid with 15828 tri elements, Naca0012, Ma_∞ =0.85, angle =1.25˚.

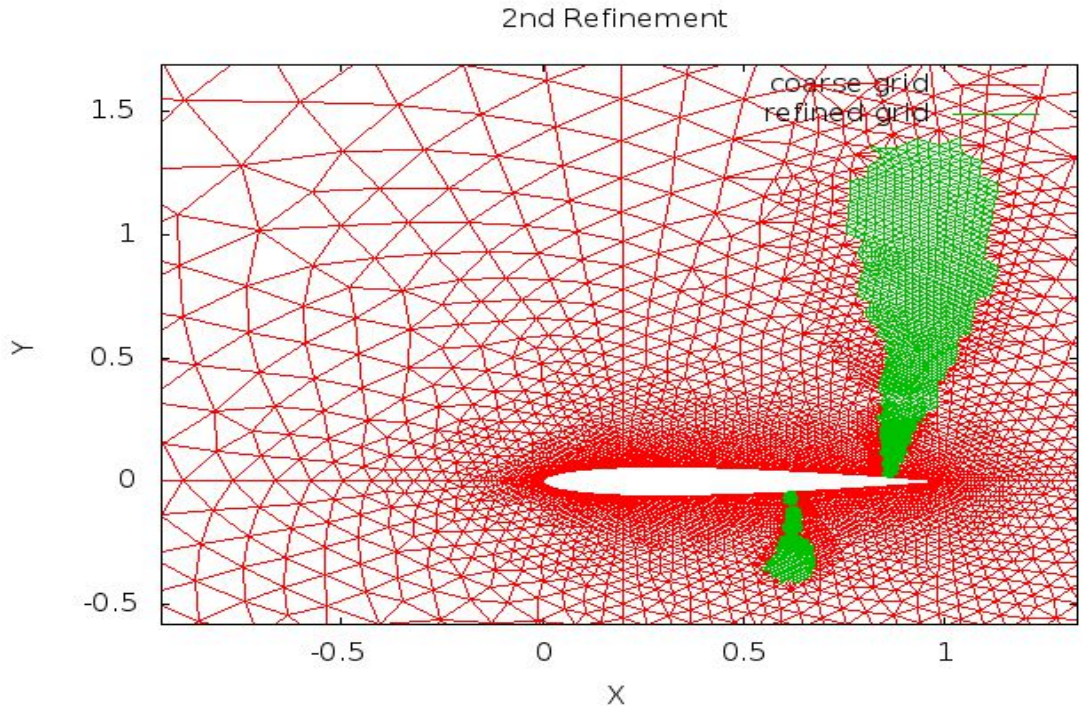


Fig.48: 2nd Refinement, grid with 20245 tri elements, Naca0012, Ma_∞=0.85, angle =1.25˚.

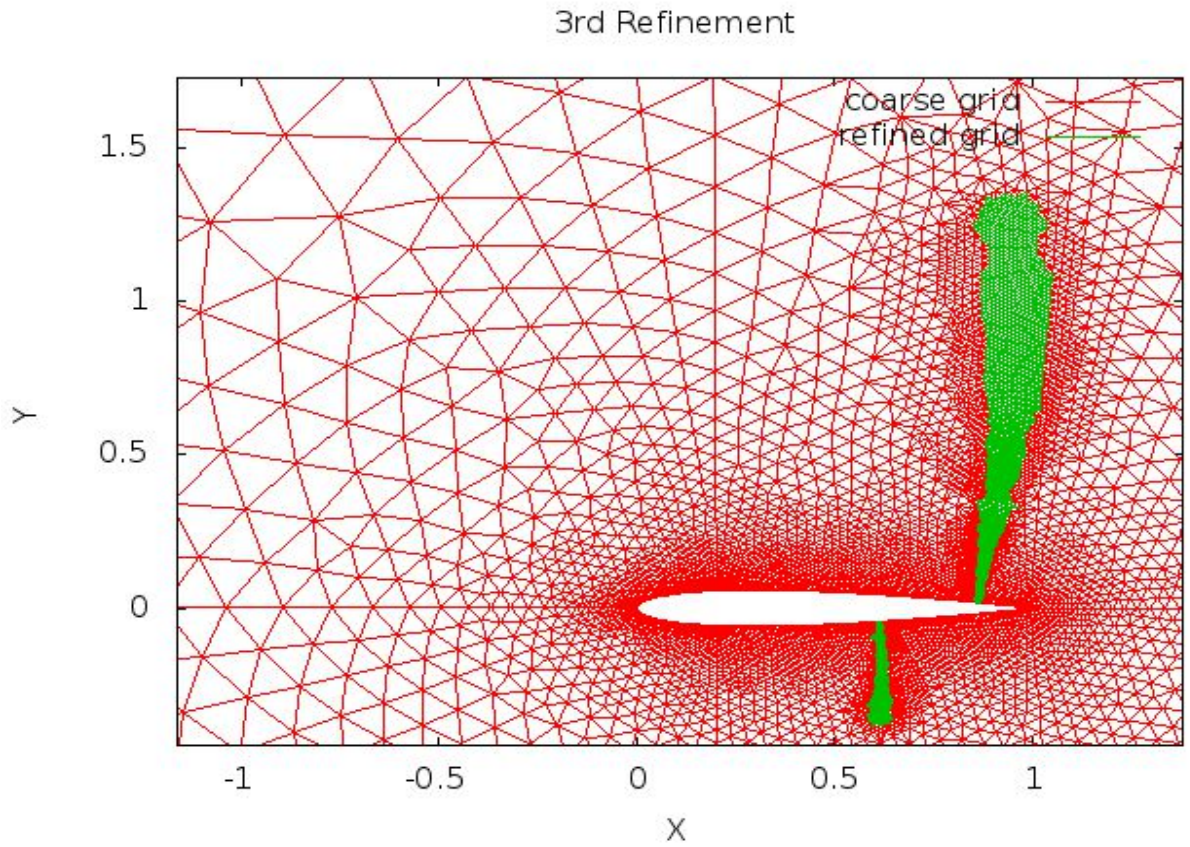


Fig.49: 3rd Refinement, grid with 28891 tri elements, Naca0012, Ma_∞ =0.85, angle =1.25˚.

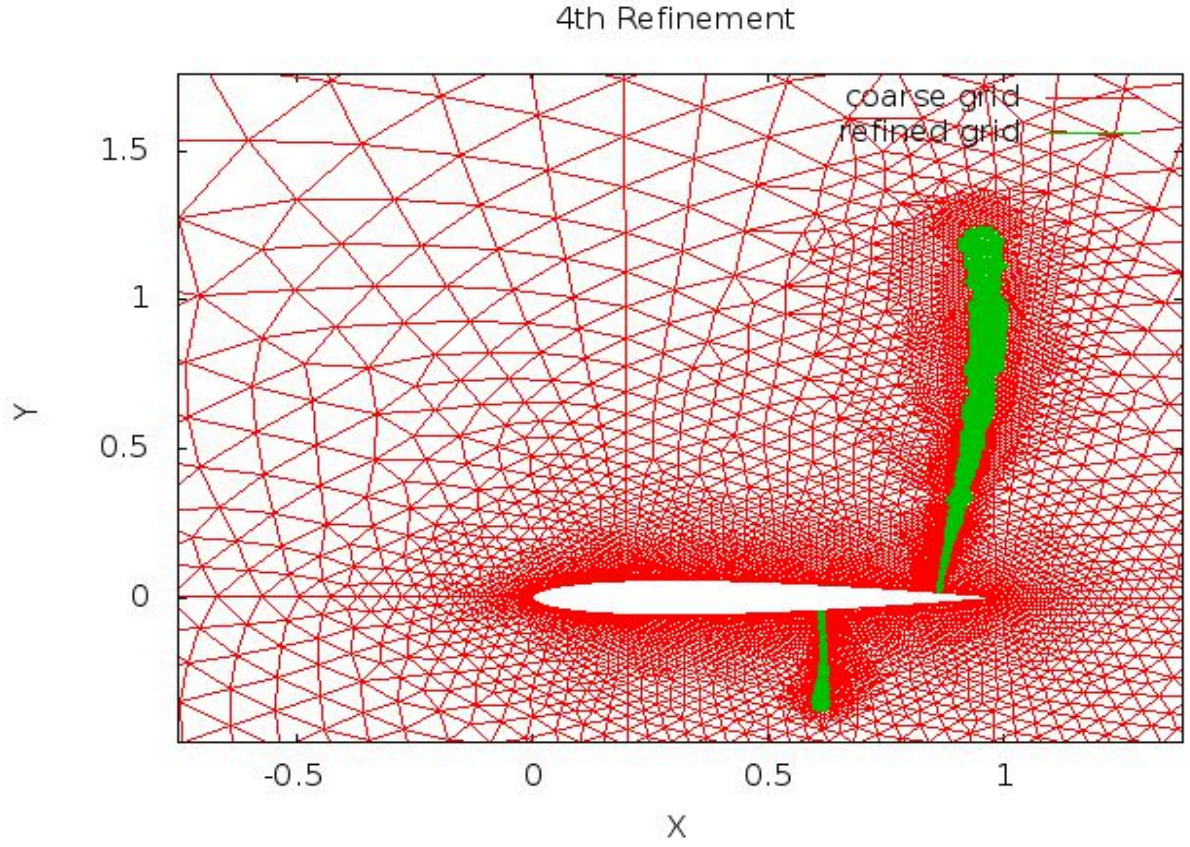


Fig 50: 4th Refinement, grid with 45955 tri elements, Naca0012, Ma_∞ =0.85, angle =1.25˚.

The pressure and lift coefficient and mean error of density with respect to time are the following:

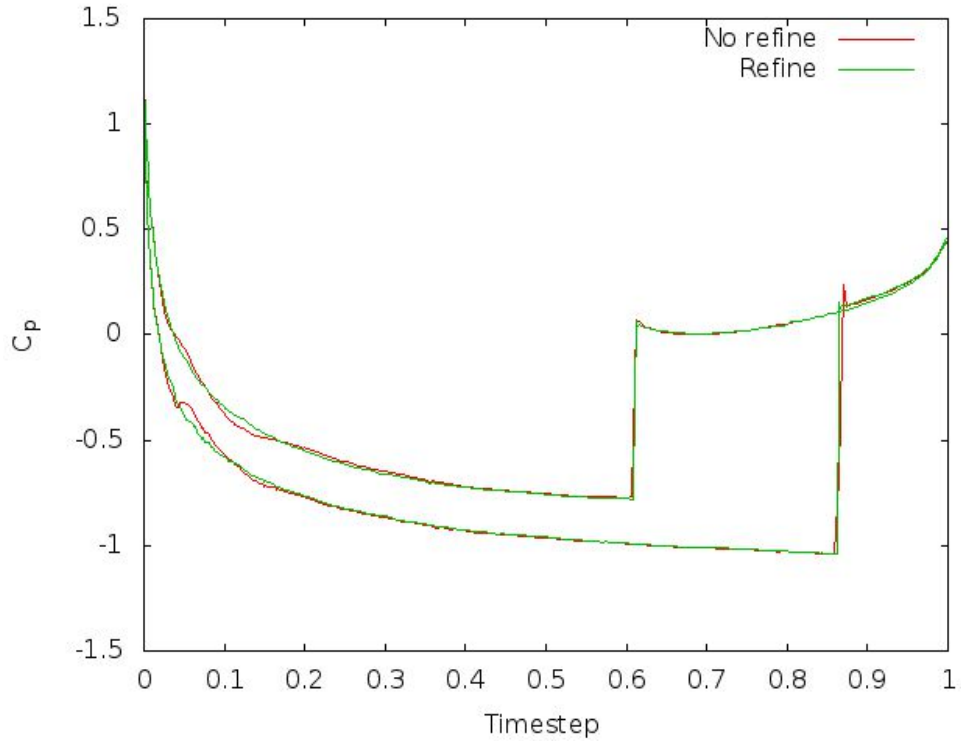


Fig.51: Pressure coefficient for 4 refinements, with an initial grid with 13586 tri-elements, Ma_∞ =0.85, angle = 1.25˚.

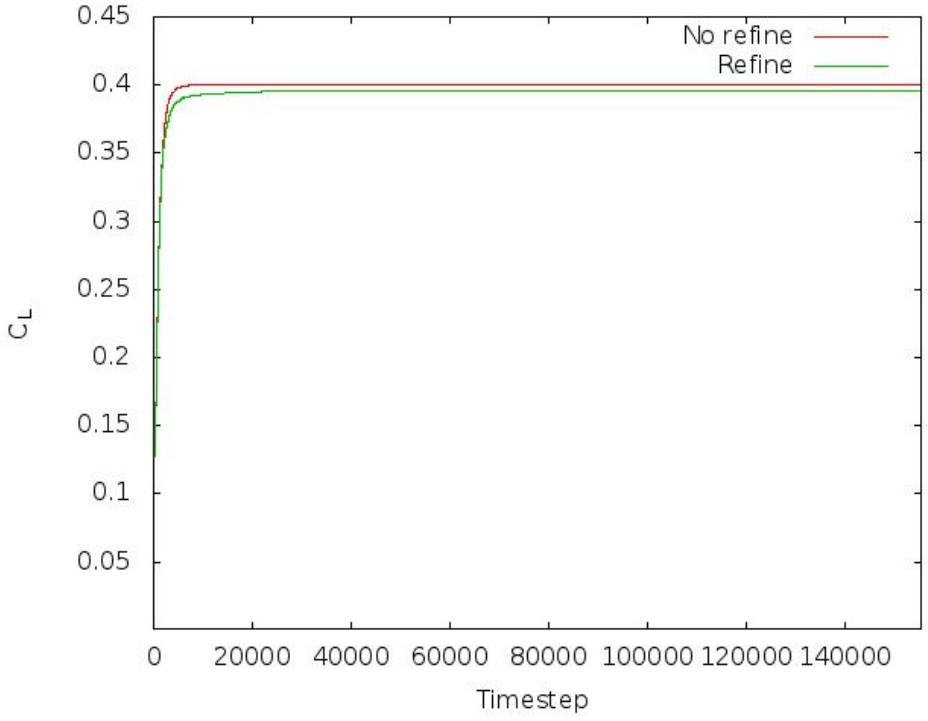


Fig. 52: Lift coefficient for 4 refinements with an initial grid with 13586 tri-elements, Ma_∞=0.85, angle =1.25˚, Naca0012.

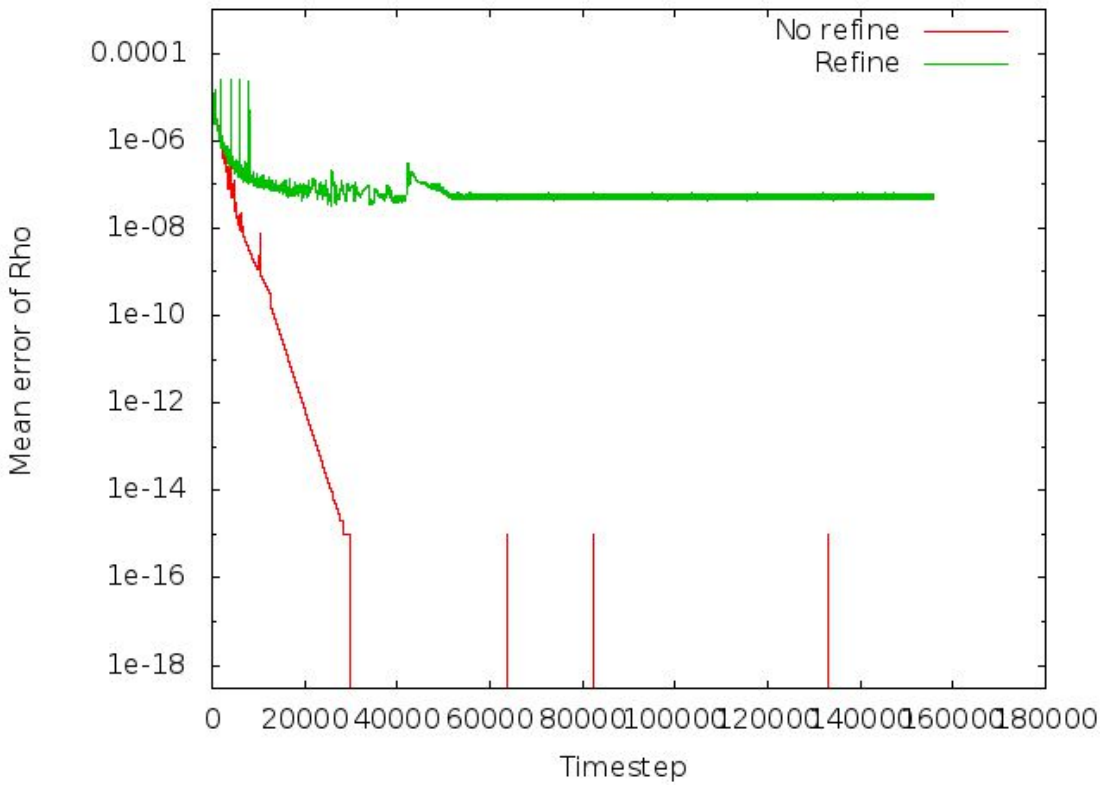


Fig. 53: Mean error of density through time evolution after 4 refinements with a coarser initial grid, Ma_∞ =0.85, angle =1.25˚, Naca0012.

– As shown in picture 51 the graph of C_p is quite smooth, and every peak corresponds to a shock wave like before. This pressure profile is satisfactory and the sharp slopes at the non refined case are eliminated.

– Similarly, at the lift coefficient graph looks different than the non refined case. In this case for the first time the lift takes lower values after the refinement: The lift is reduced with the grid changes. The reason is because of the pressure coefficient around the airfoil is changed with the new mesh and so does the integral of the pressure (as a result the lift coefficient).

– Mean error of density is slightly improved with respect to the less populated initial grid in the previous case. Generally, the run cases proved that an initial grid with much more cells may give the same convergence, with an initial one with fewer cells at the same flow conditions and refinement criteria at Mach =0.85, flow angle = 1.25°. This means that starting with 13586 cells, and applying some refinements, we may expect the solution to convergence, with a cut-off error at 1e-07 with a grid with more cells than before.

The mean error in the less populated mesh is distributed at big computational cells. A mesh like this consists of cells with mach number less than 1 that share faces with others with mach greater than 1. When applying a refinement, cells like these still exist across the refinement regions. The number of these elements increases and the shock wave is captured with more details after each refinement level. This can be shown below (fig.54). The supersonic and subsonic areas are figured and in the middle the shock wave.

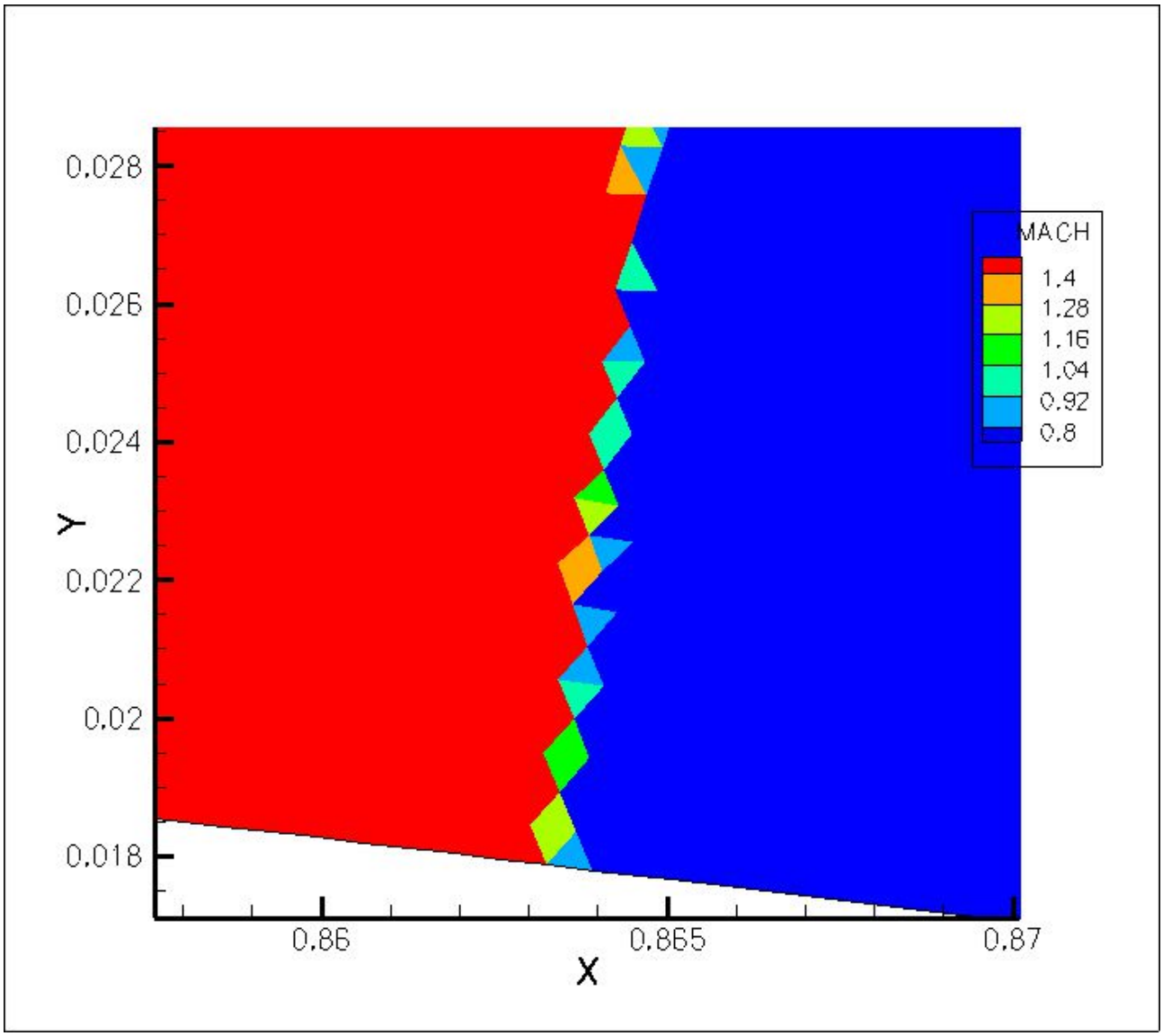


Fig. 54: Shock wave over Naca0012, (suction side) capturing primary value per cell , after 4 refinements.

4.3 Inviscid flow, $Ma_\infty = 0.6$, angle of inflow = 2.0°.

This test case is theoretically easier than the previous ones because of the weak shock wave that is developed over the airfoil Naca0012. The refinement criterion was the error and the absolute difference of mach number with smaller values than above test cases. Other criteria were also applied such that the divergence of pressure and the density combined with the error indicator. These all criteria although have the negative feature: the dispersion of the quantity of interest, after every refinement. For example the mean and max value for the pressure is double after the first level. Consequently the solver selects and refines not only the shock wave area but more regions with the Mach criterion (fig. 55).

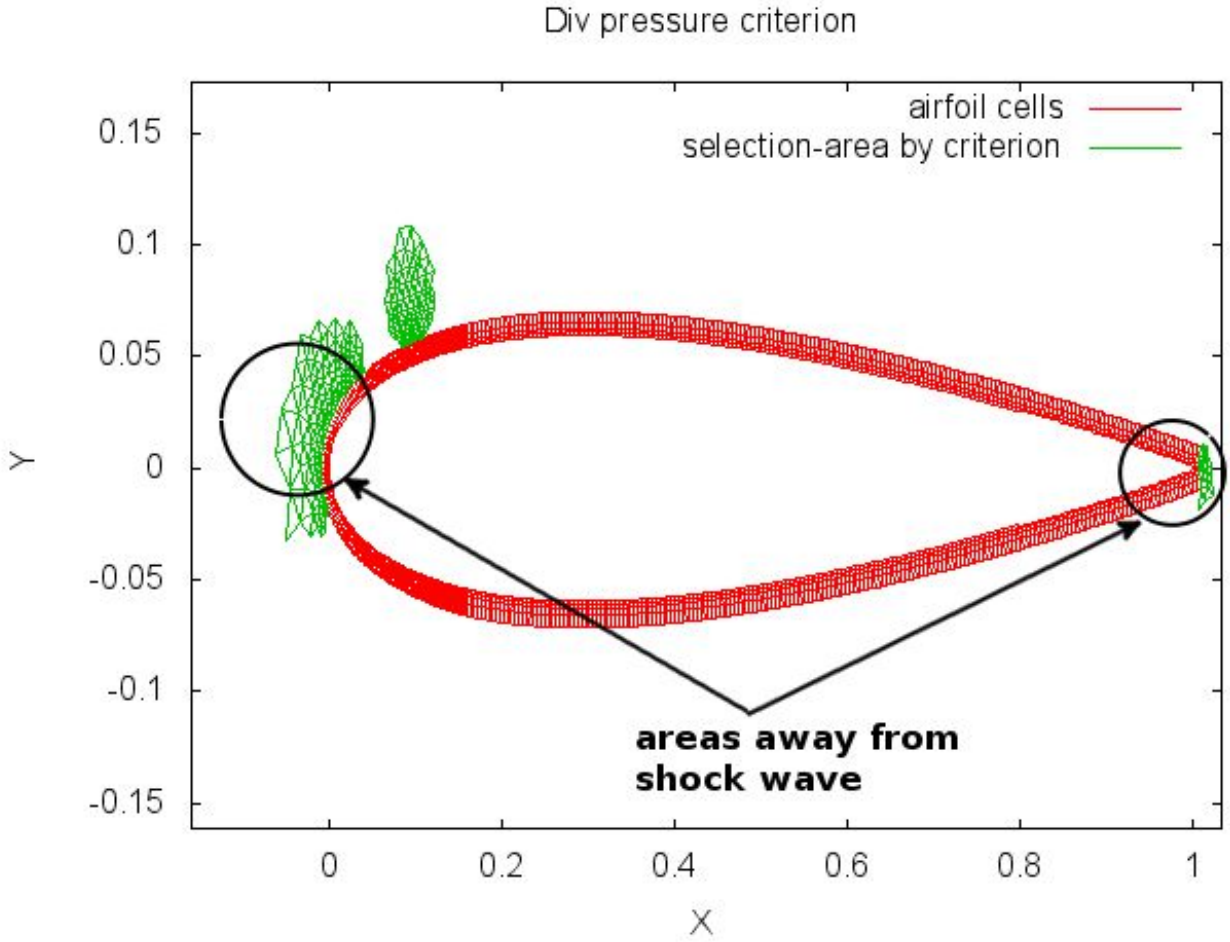


Fig. 55: Selected area, for Ma_∞ =0.85, angle of inflow 4.0˚.

Every function criterion can be eventually be applied using an appropriate value as limit.
Initial unstructured mesh with only triangle cells:
Coarse grid with 3658 triangles and refinement selection with the following quantity:

$$\sqrt{\left(\frac{\partial p}{\partial x}\right)^2 + \left(\frac{\partial p}{\partial y}\right)^2}$$

And the mean error per cell equal to 10d-01 (higher than before). An extra limitation for the coarse grid to have less than 6000 cells. As a result 4 levels of refinement were applied (fig. 56-57-58-59).

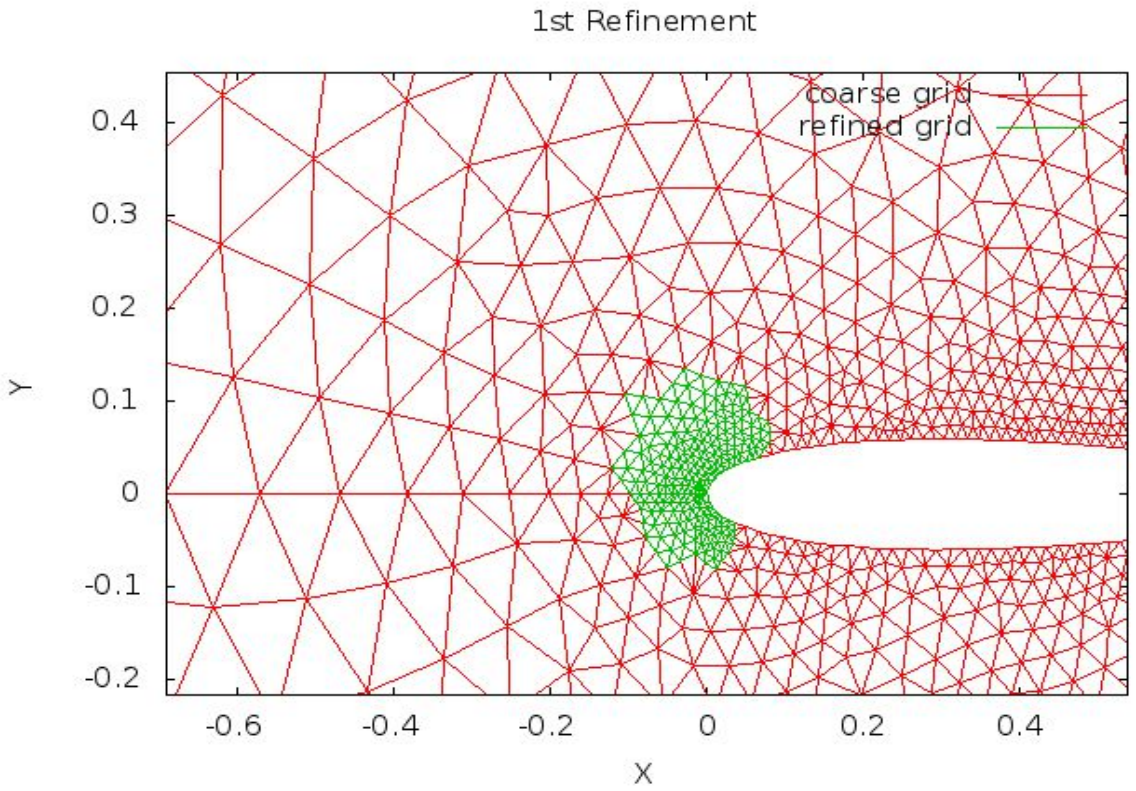


Fig. 56: 1st refinement, 3956 triangle elements, Mach =0.6, angle =4.0˚, Naca0012.

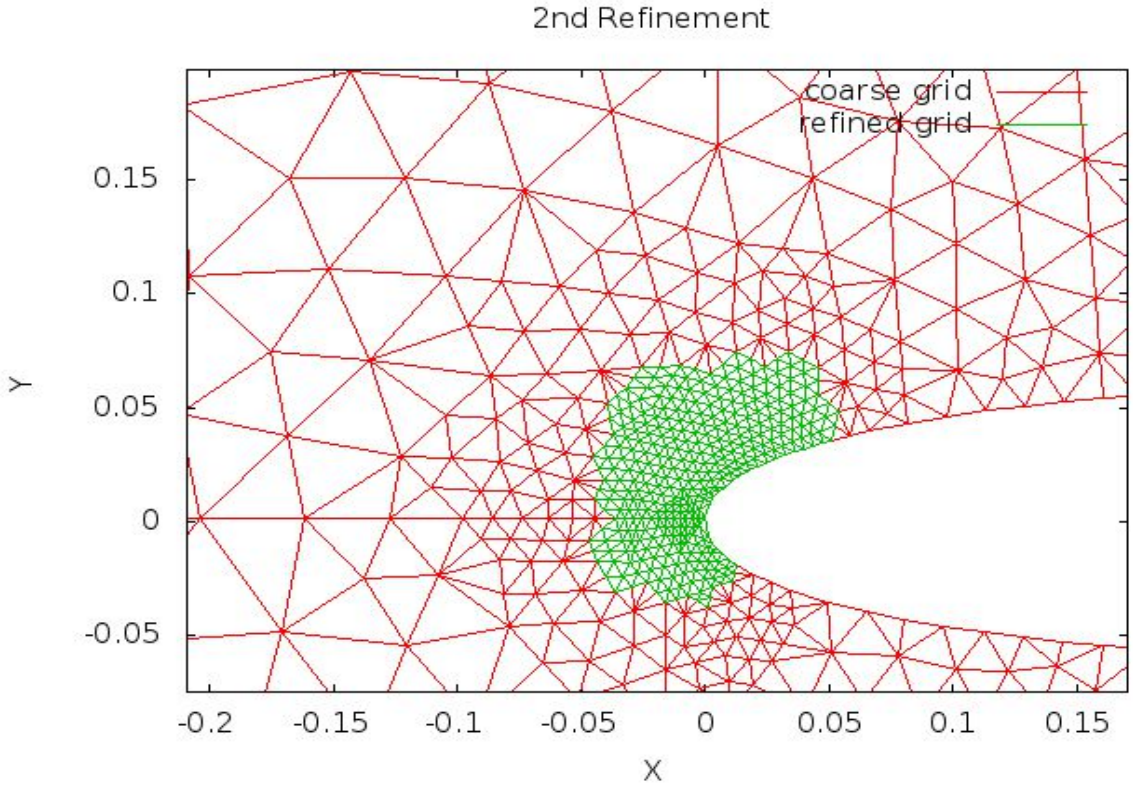


Fig. 57: 2nd refinement, 4441 tri-elements, Ma_∞ =0.6 ,angle =4.0˚, Naca0012.

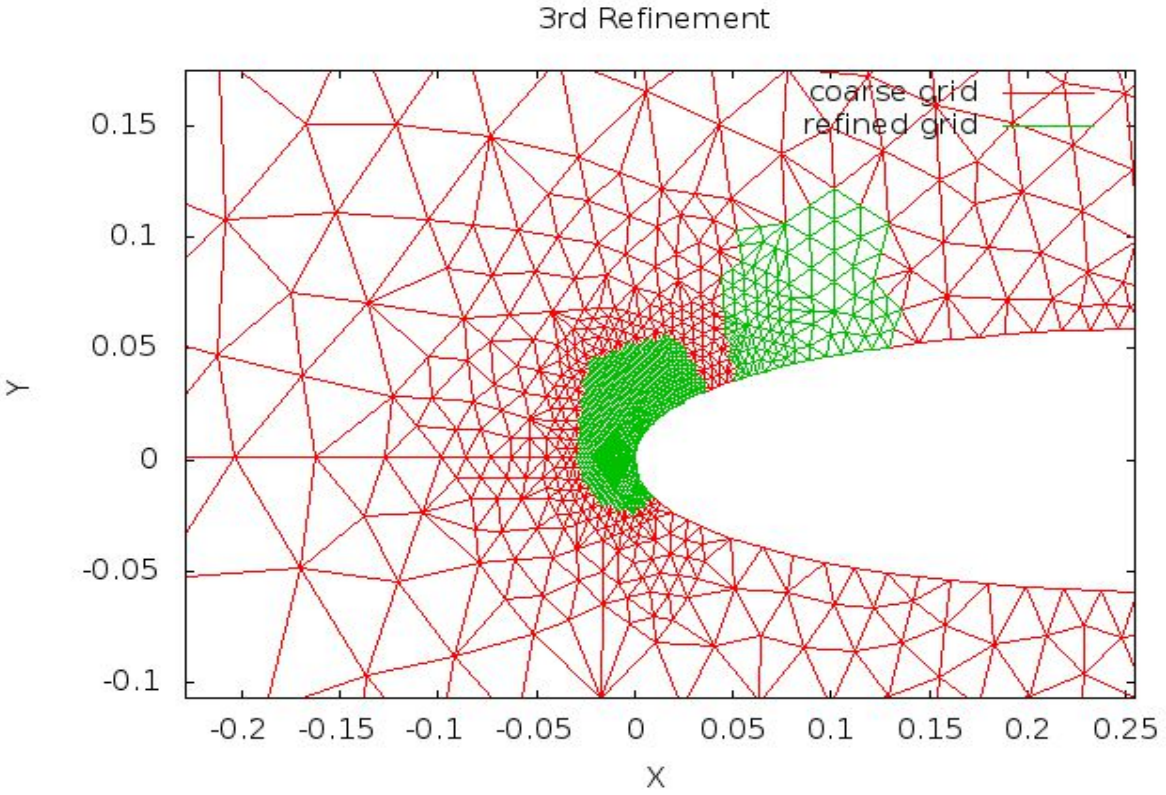


Fig. 58: 3rd refinement 5656 tri-elements, Mach =0.6, angle =4.0˚, Naca0012.

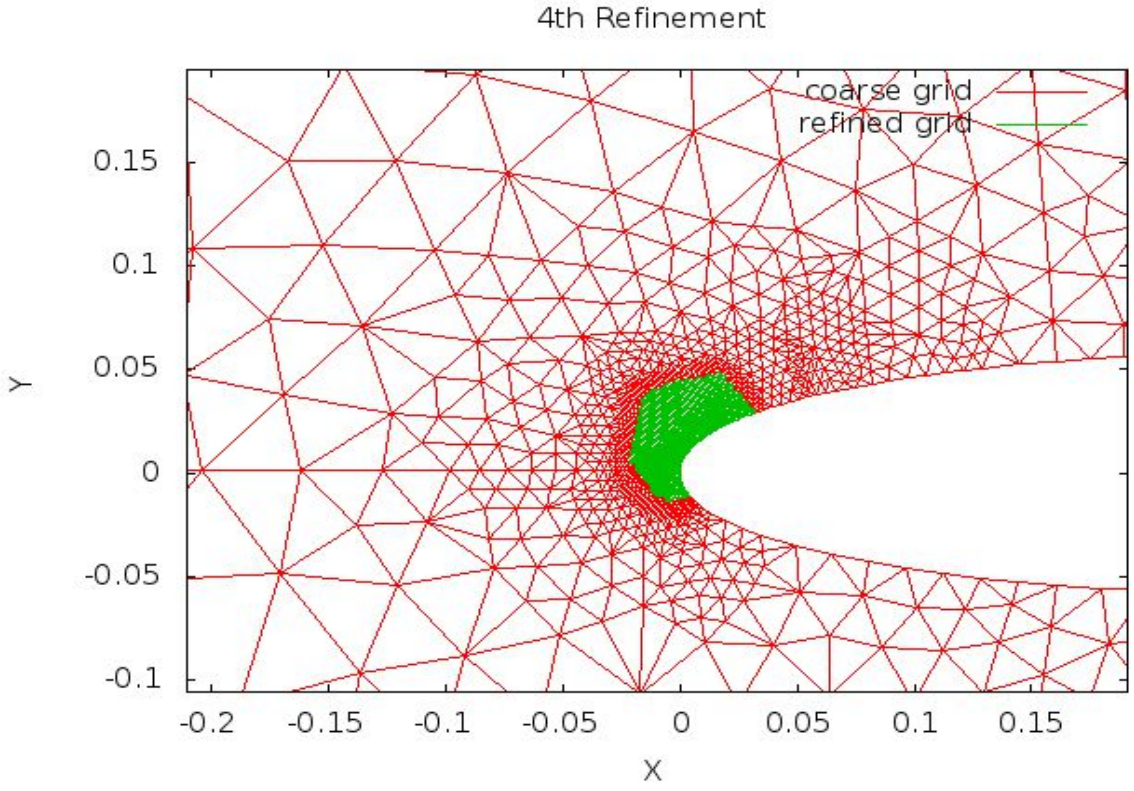


Fig. 59: 4th refinement, 8351 tri-elements, Mach =0.6, angle =4.0˚, Naca0012.

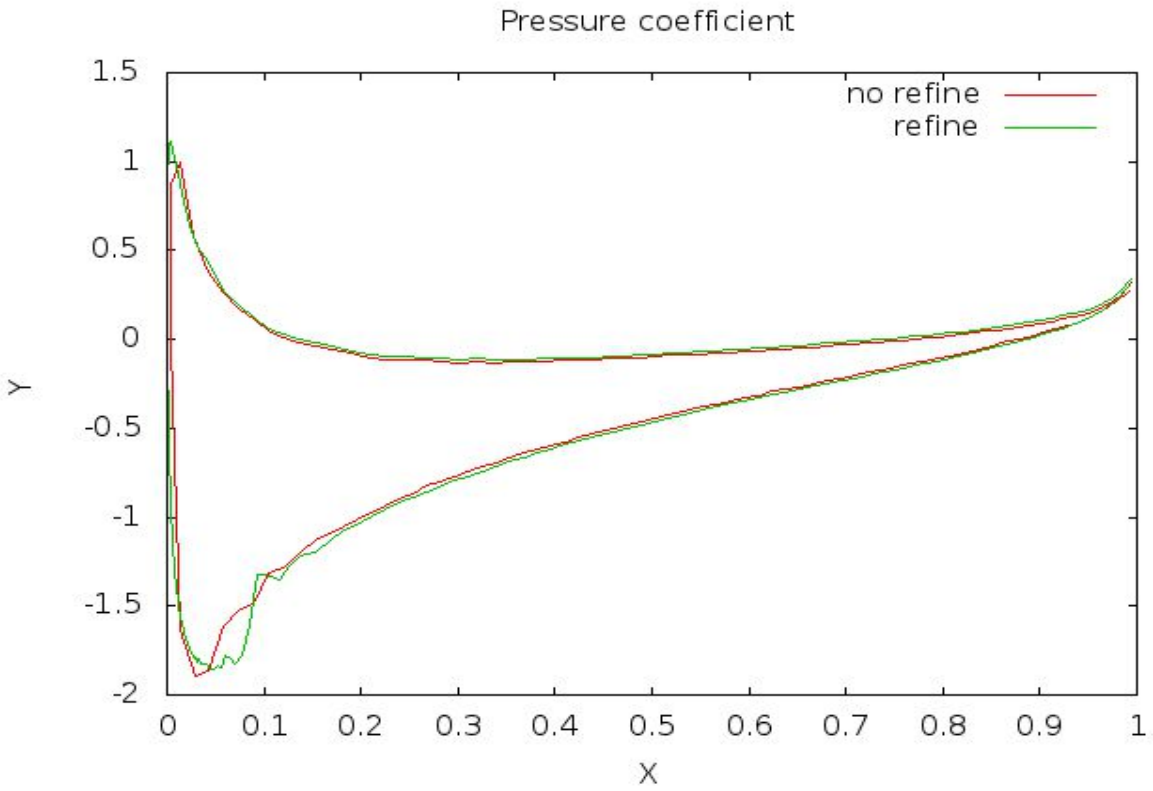


Fig. 60: Cp for Naca0012, refined and non-refined cases for Ma_∞ =0.6, angle =4.0˚.

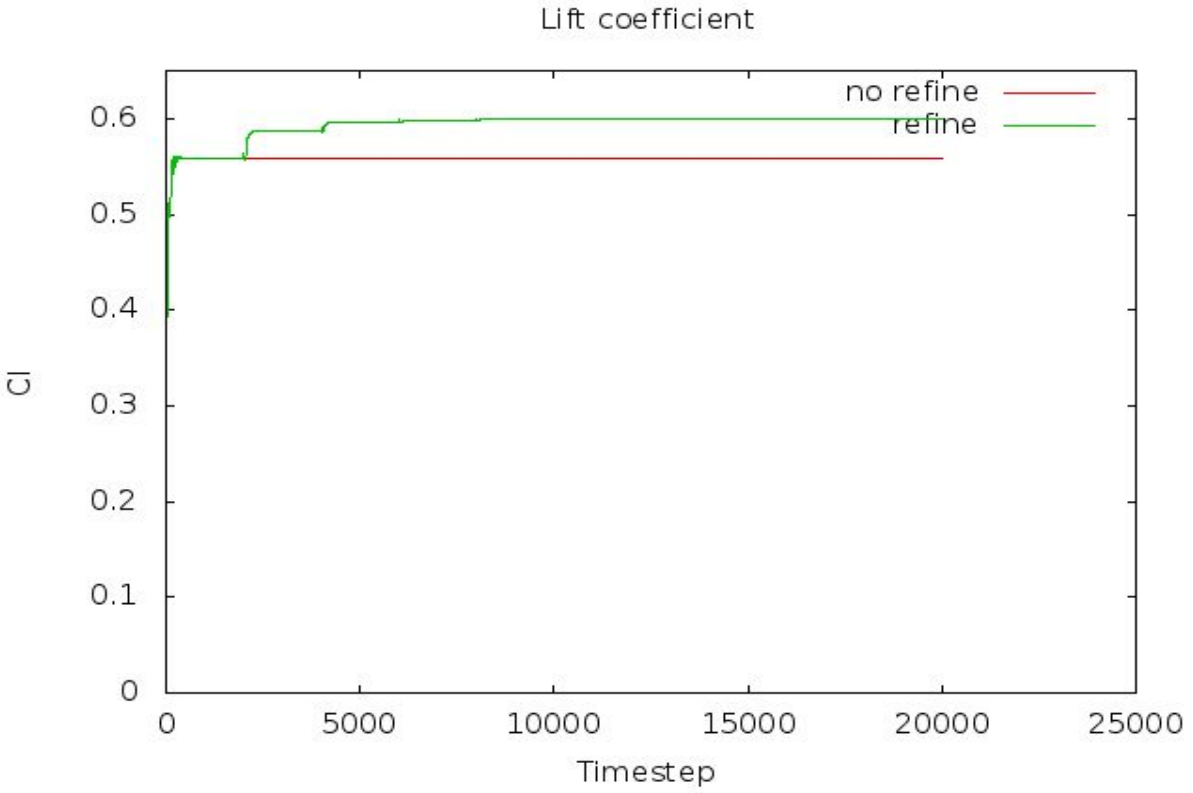


Fig. 61: Lift coefficient through convergence history for refined and non-refined cases, Ma_∞ =0.6, angle =4.0˚.

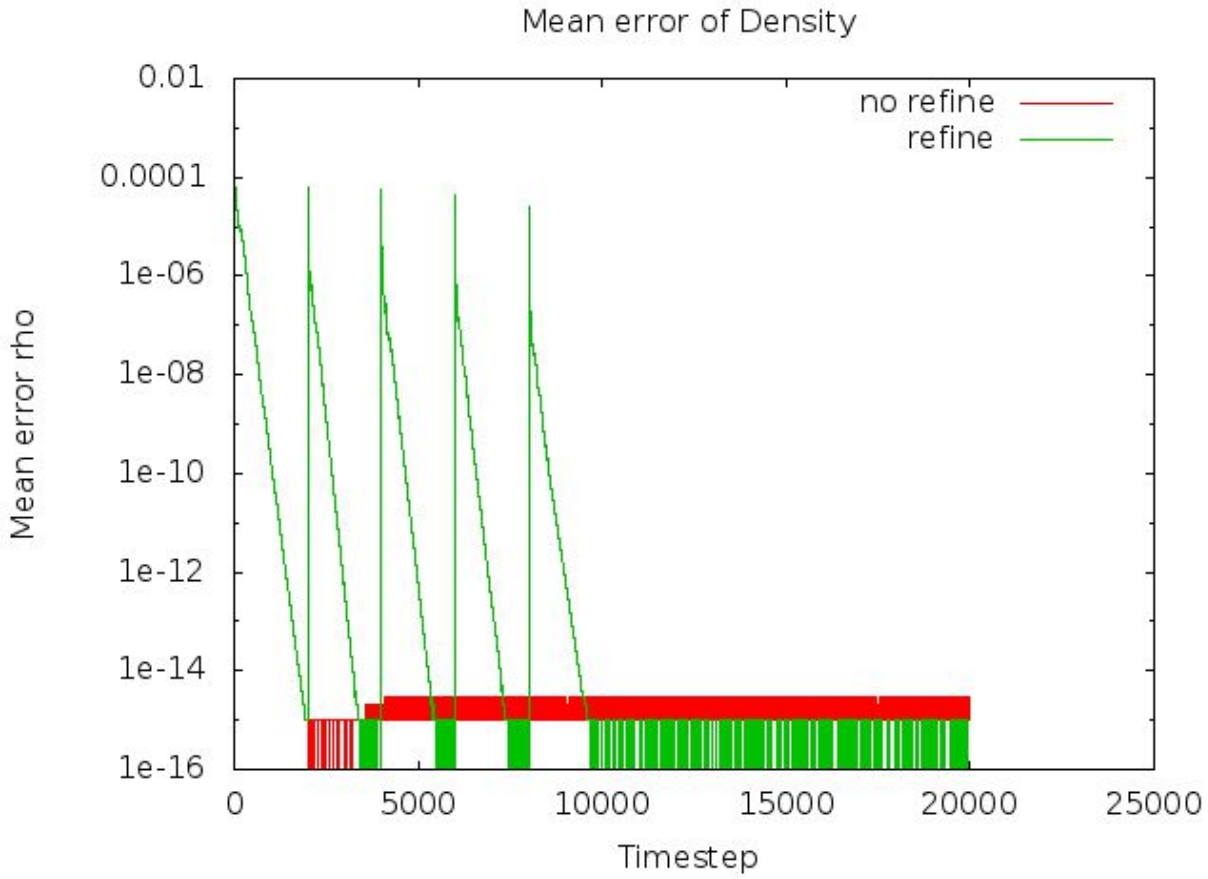


Fig. 62: Mean error of density for Naca0012, refined and non-refined cases, Ma∞ =0.6, angle =4.0˚.

– For a small rate of refinement ~2.28 times the initial number of cells, the mean error of density drops at the same and even lower levels of non refined case. The criterion we chose represents the maximum variation of the pressure in the computational domain. The solver selects areas with large mean pressure changes (fig. 63). As we can see one of these areas is the one that the shock shows up (fig.64). The stagnation point is chosen too and more nodes in the boundary are constructed with the numerical methods mentioned at chapter 3.

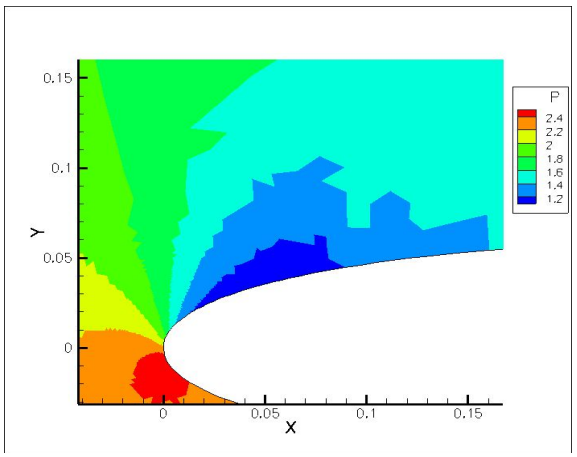


Fig. 63: Pressure distribution around Naca0012, Ma =0.6, angle =4˚.

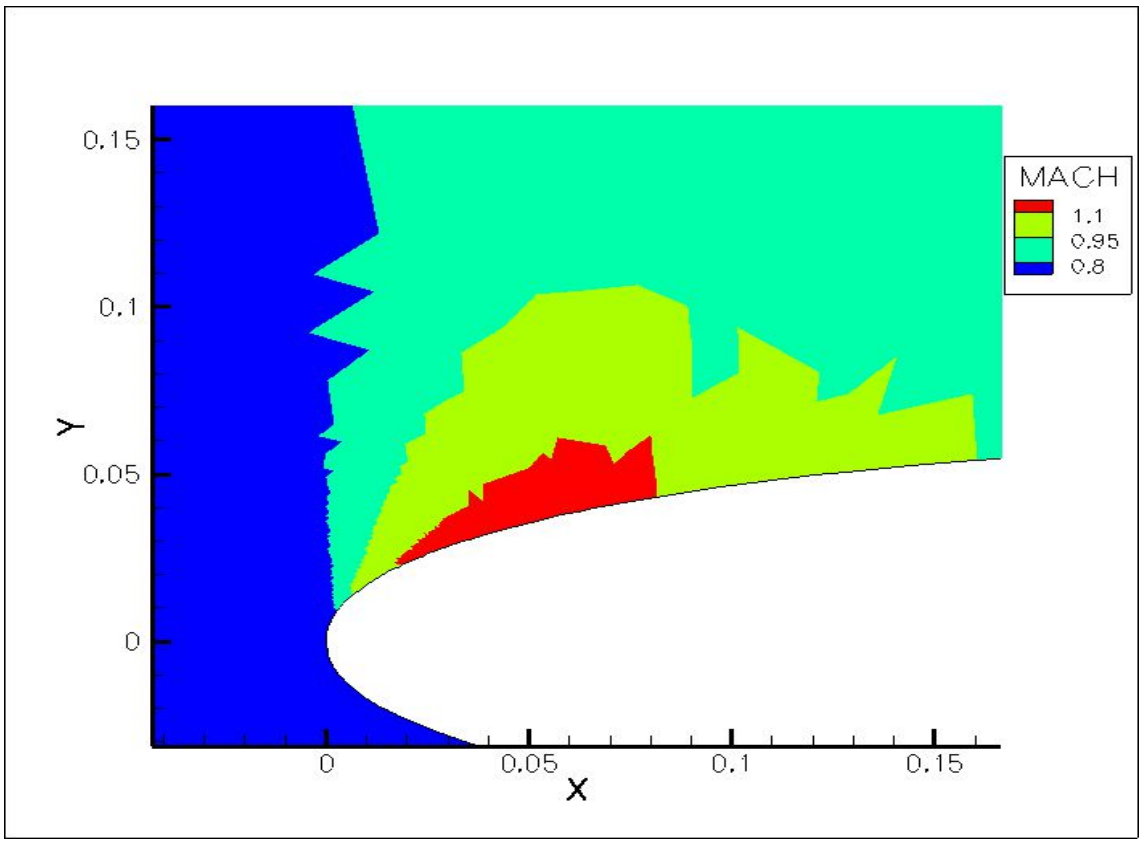


Fig.64: Mach distribution around Naca0012. Solver refines the area of the shock wave too.

Pressure and lift coefficients are presented in figures 60 and 61. Wiggles at pressure graph still occur due to unstructured mesh. This problem can be solved with combining triangle and quadrilateral cells in the initial grid. For the hybrid case that follows a limitation to the number of cells at the coarse grid at each refinement was applied (less than 15.000 cells).

- **Hybrid mesh with triangle and quad cells.**

We test this same flow (Ma_∞ =0.6, angle of flow = 4˚) using the mach difference criterion (10d-05) with the error indicator of energy per cell.

As we can see (fig. 65-71), the new grids are refined at the stagnation point and the around shock wave as it was expected.

Initial number of cells is 11463 cells. The captions of each refinement are following:

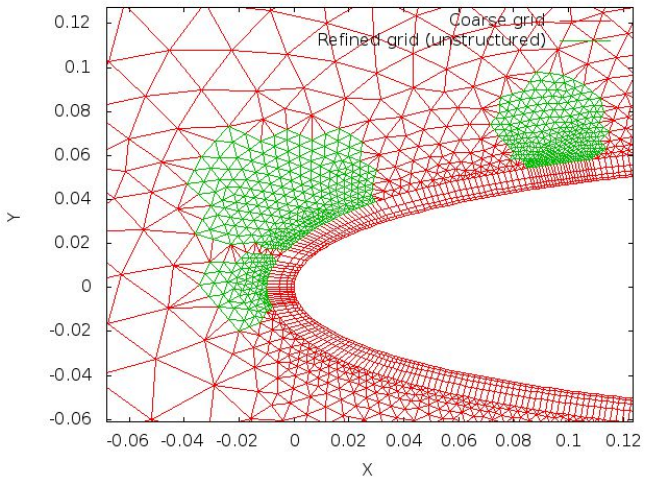


Fig.65a) : 1st refinement .Number of elements 11463 hybrid mesh .Ma_∞ =0.6 ,angle =4˚, Naca0012.

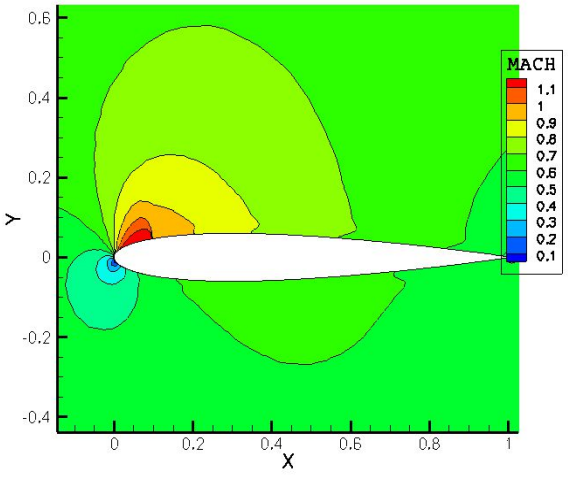


Fig.65b) : Mach distribution over Naca0012 for Ma_∞ =0.6 angle 4˚.

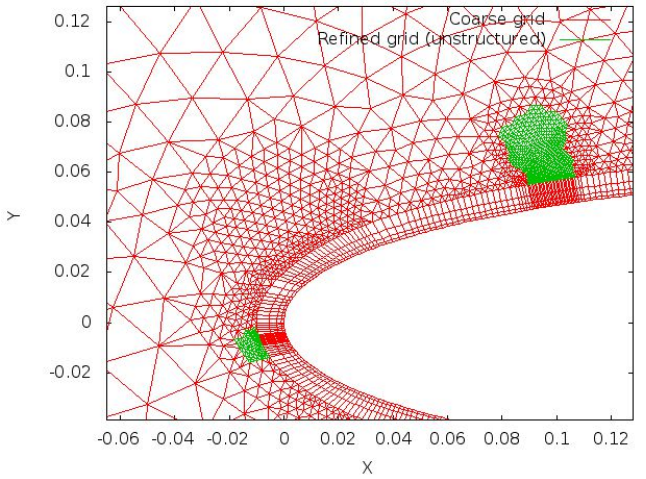


Fig. 66a) : 2nd refinement .Number of elements 12573 hybrid mesh. Ma_∞ =0.6 ,angle =4˚,Naca0012.

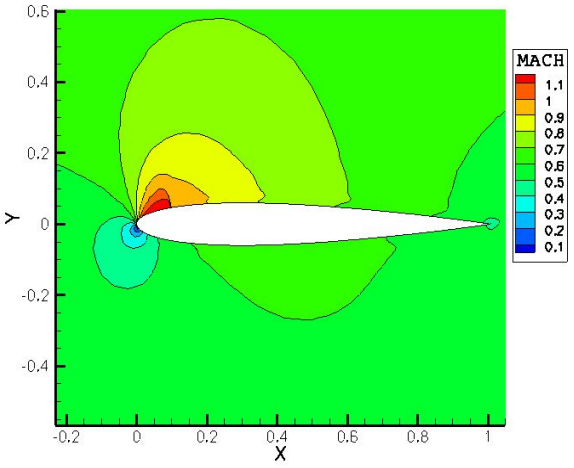


Fig.66.b) :Mach distribution for 2nd refinement.

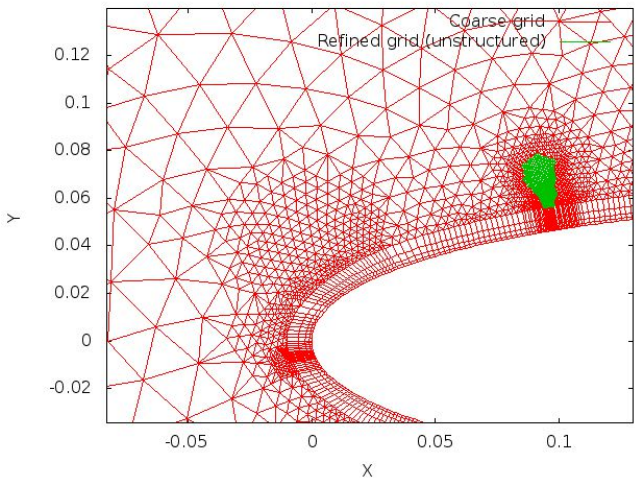


Fig.67a) :3rd refinement , 13899elements hybrid mesh. Ma_∞ =0.6 ,angle 4˚,Naca0012.

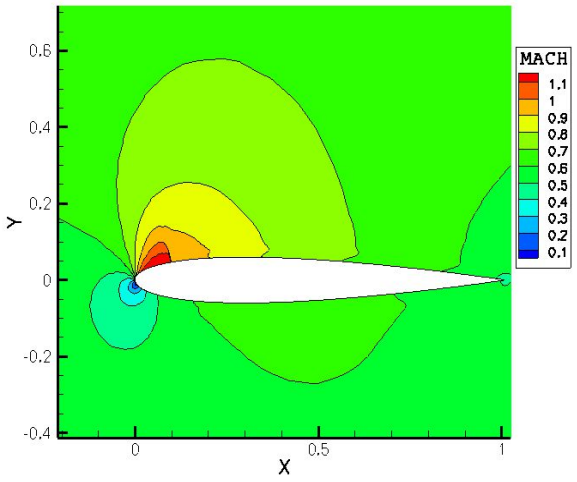


Fig.67b) :Mach distribution over Naca0012.

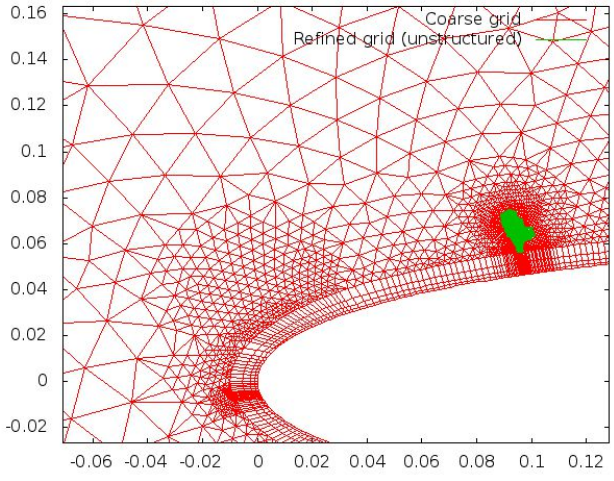


Fig. 68a) : 4th refinement .Number of elements 16872, hybrid mesh .Ma_∞ =0.6 ,angle =4˚, Naca0012.

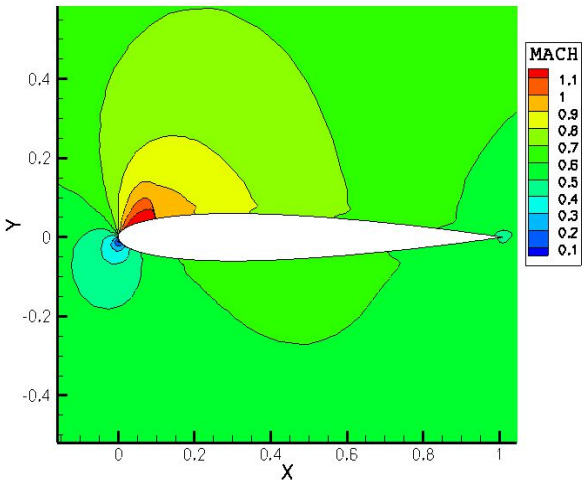


Fig.68b) :Mach distribution over Naca0012.

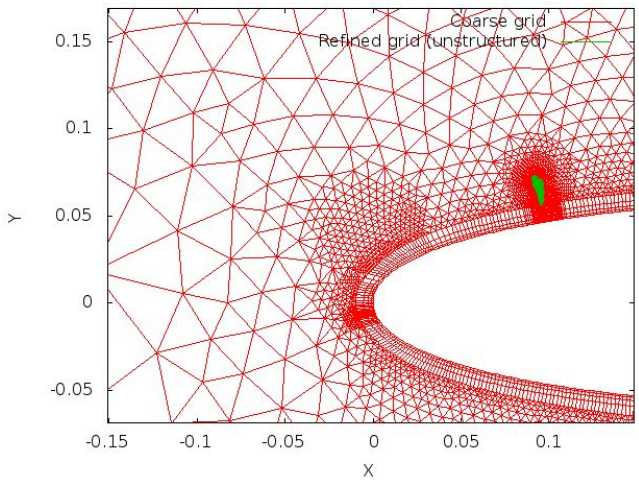


Fig.69a) : 5th refinement .Number of elements 23836 , hybrid mesh .Ma_∞ =0.6 ,angle 4˚ ,Naca0012.

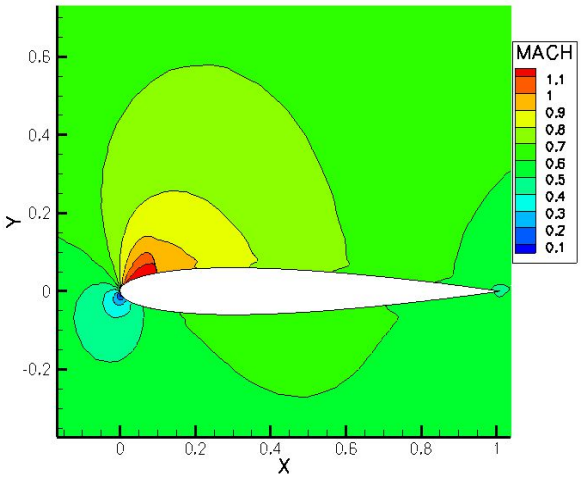


Fig.69.b) : Mach distribution for 5th refinement.

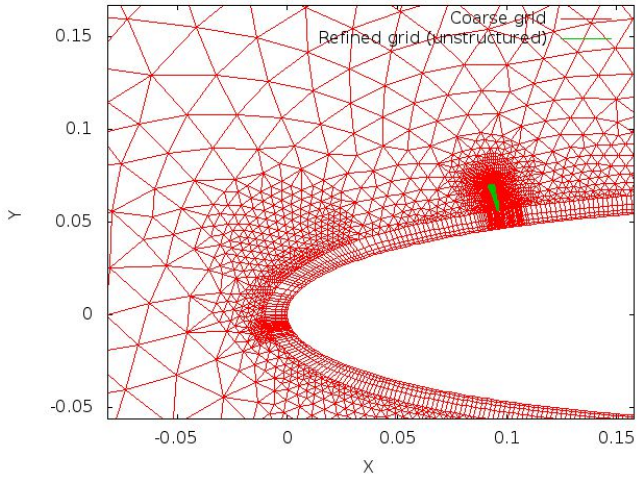


Fig.70a) : 6th refinement ,Number of elements 36168 , hybrid mesh .Ma_∞ =0.6 ,angle 4 ,Naca0012.

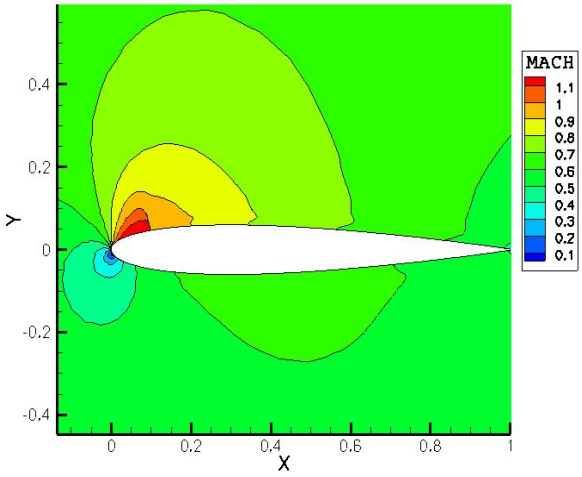


Fig 70b) : Mach distribution for 6th refinement.

The grid changes both at structured and unstructured part. Using the suitable values for the geometric smoothness we may obtain as figured at pictures *.α) finer meshes at the shock wave region around the airfoil.

Pressure and lift coefficient and mean error of density diagrams are following next compared to the non refined case (fig. 71-74):

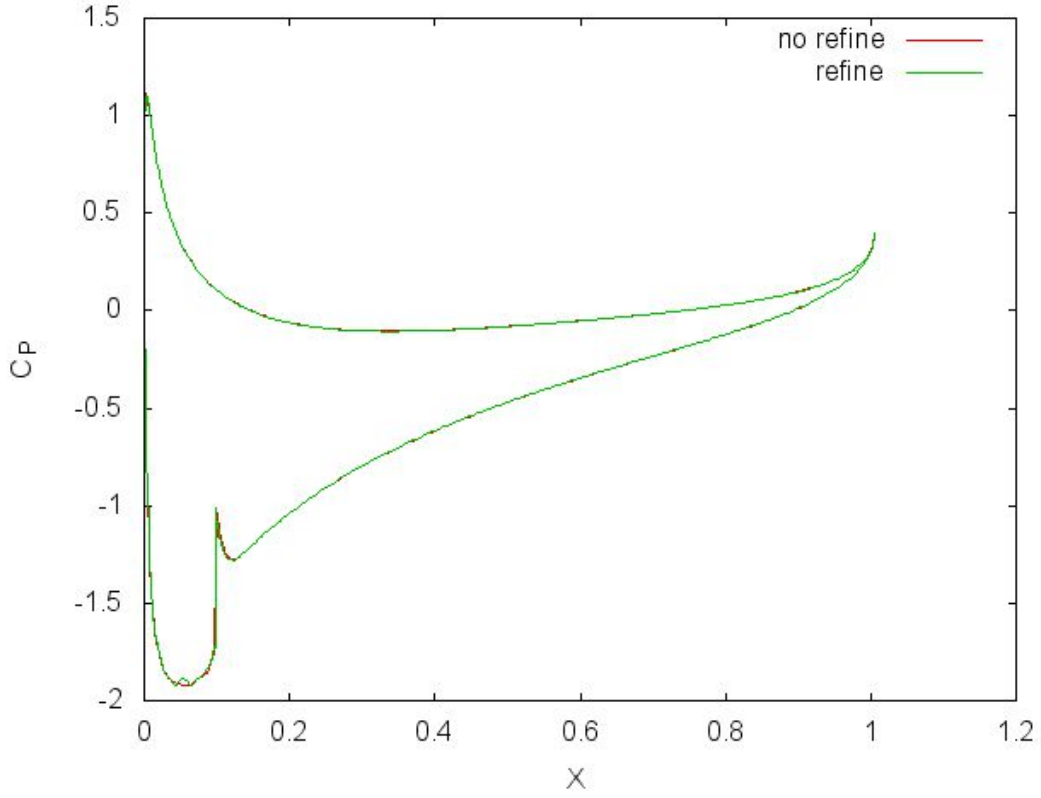


Fig.71) : Pressure coefficient for both cases of refined and non-refined grid , Ma_∞ =0.6 ,angle 4˚ over Naca0012.

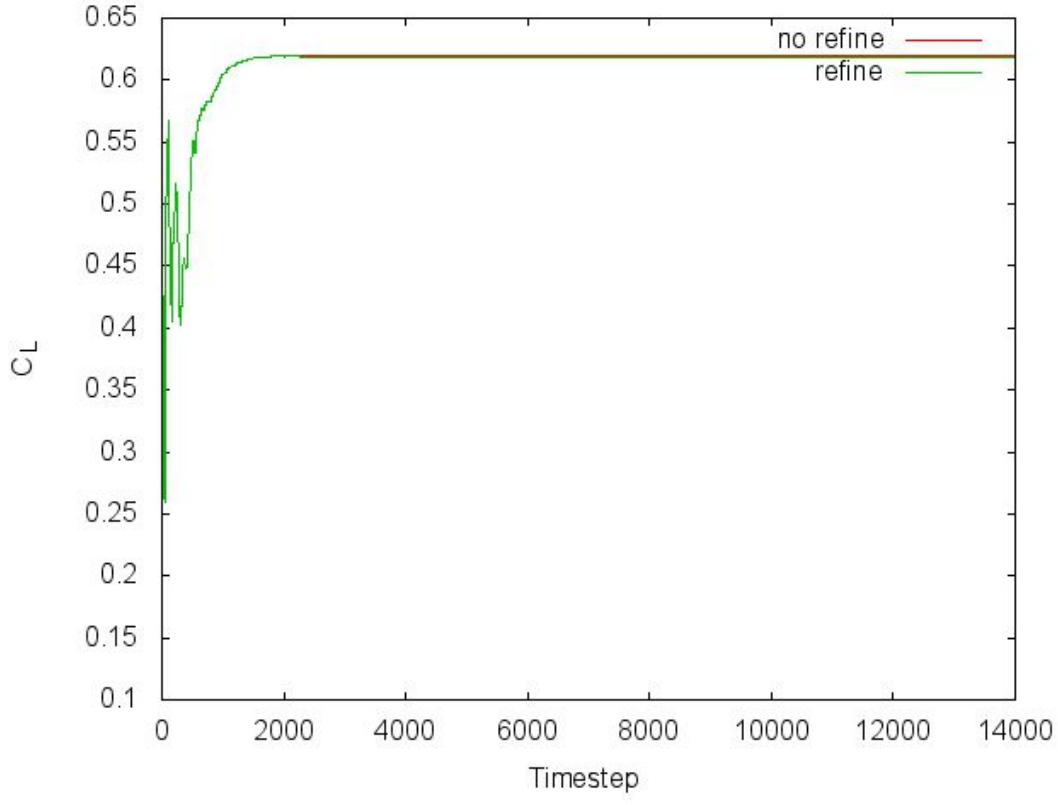


Fig.72) : Lift coefficient for the refined and non-refined grid ,cases .Ma_∞ =0.6 ,angle =4˚ ,over Naca0012.

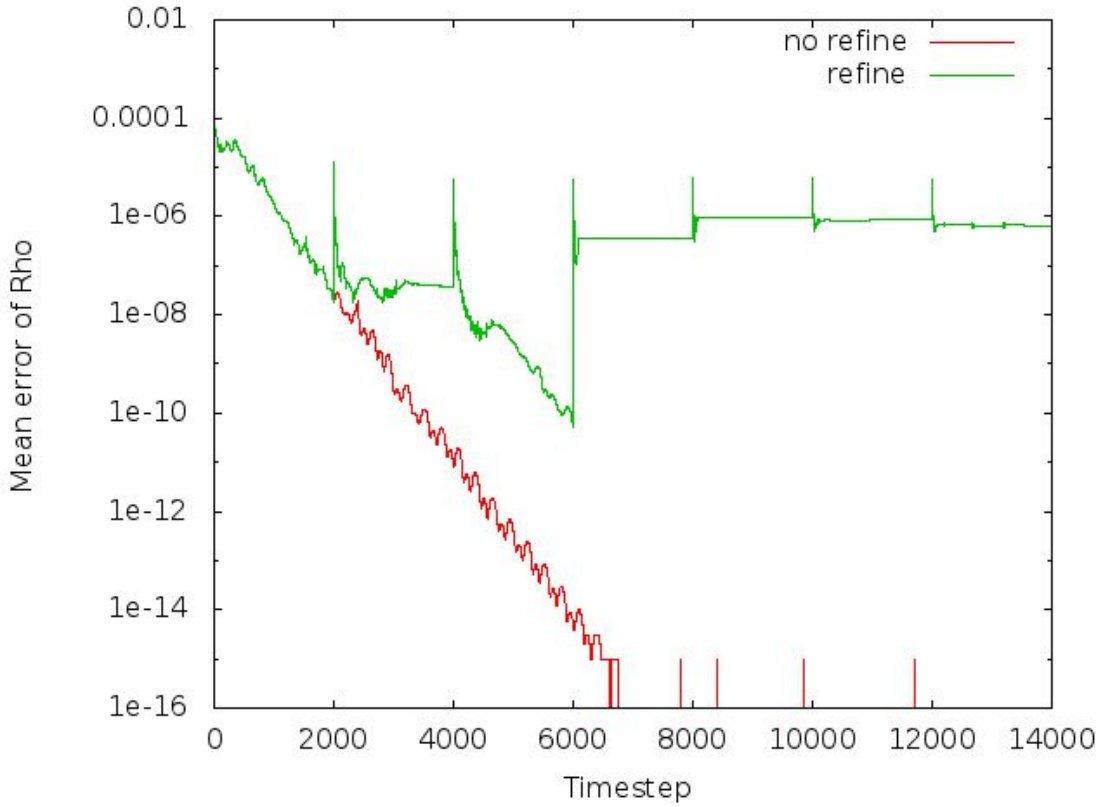


Fig.73) : Mean error of density for refined and non-refined cases. Every peak of the graph is assigned to a refinement .Ma_∞ =0.6, angle 4˚, Naca0012.

Pressure coefficient is properly simulated and wiggles are eliminated after the refinements (fig. 71). Lift coefficient stays the same as the non refined test case. The huge difference appears at the mean error of density. Sub regions with large energy error surrounded by ones with lower (fig. 74a) these areas may exist across the shock wave at the unstructured part near the structured one where the elements have high aspect ratio (fig74b). The final mesh has 36168 cells (3.32 times more populated than the initial).

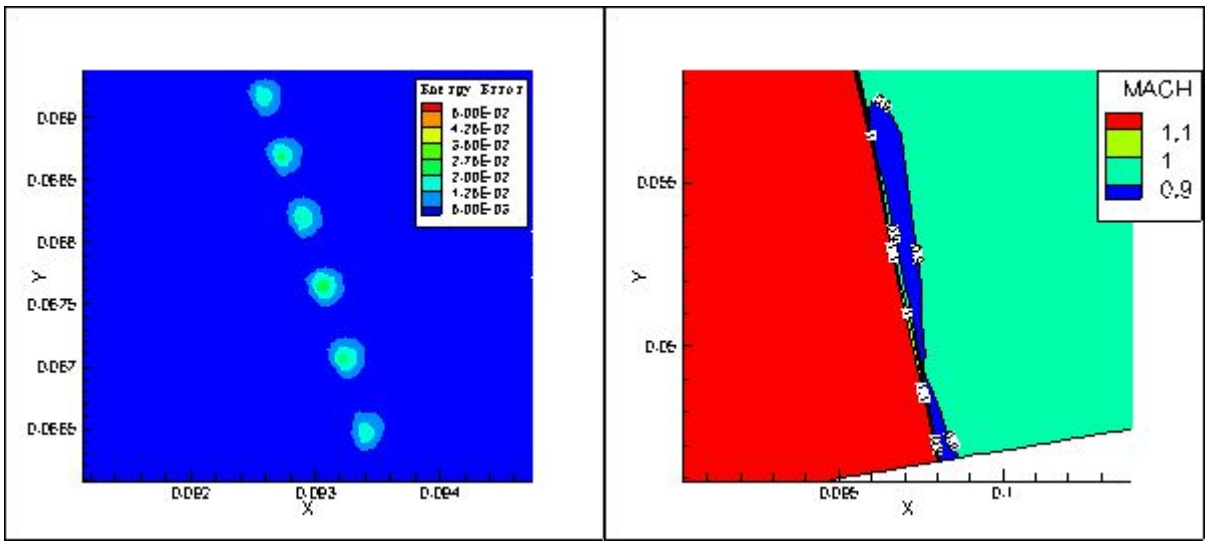


Fig.74a) : Areas with large error of energy (unstruct.). *Fig.74.b) :Elements of high aspect ratio (structured).*

4.4 Inviscid transonic flow, $Ma_\infty = 0.95$, angle of inflow = 0°

At this test case shock waves around Naca0012 form a triangle area (pressure-suction side-farfield). The criterion of absolute difference of Mach number is efficient tested here too for these shocks. These areas have also large variation of pressure (fig. 75b-76b-77b). As a result the selection was made with the divergence of pressure which was more suitable than one with mach number. The solver refines only cells that have pressure divergence larger than the input limit alongside with energy error per cell. Initial unstructured mesh with 11629 cells. Each refinement level is shown below (fig. 75a-76a-77a):

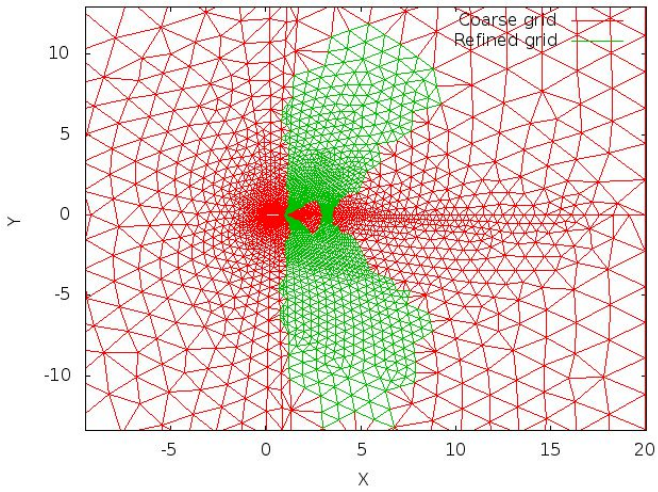


Fig. 75a) : 1st refinement , number of elements 16239 , all tris. Inviscid flow Ma_∞ =0.95 , angle 0°, Naca0012.

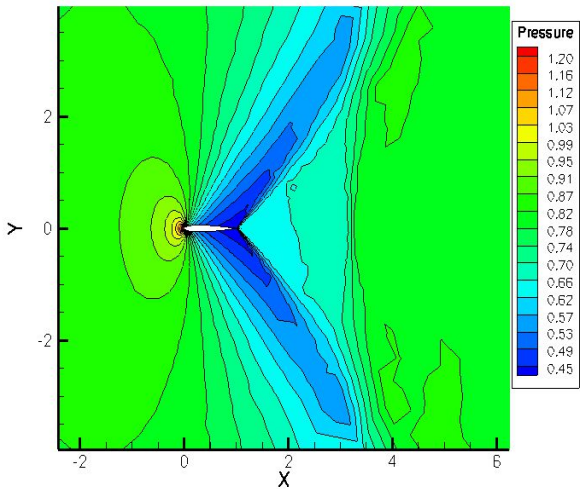


Fig. 75b) : Pressure around Naca0012.

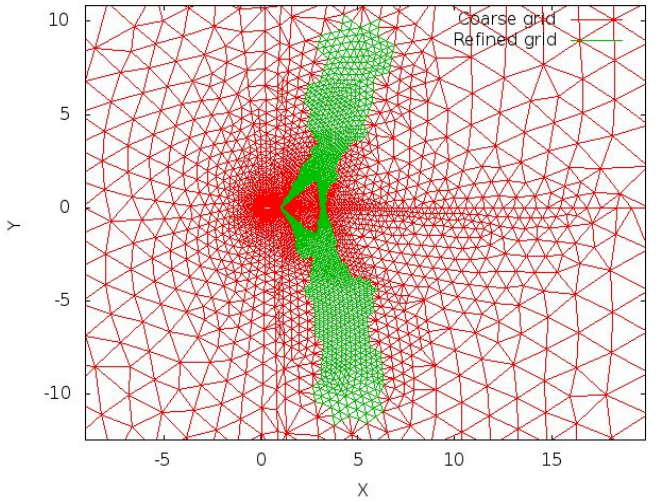


Fig. 76a) : 2nd refinement, number of tri elements 25127. Inviscid flow, Ma_∞ = 0.95, angle =0°, Naca0012.

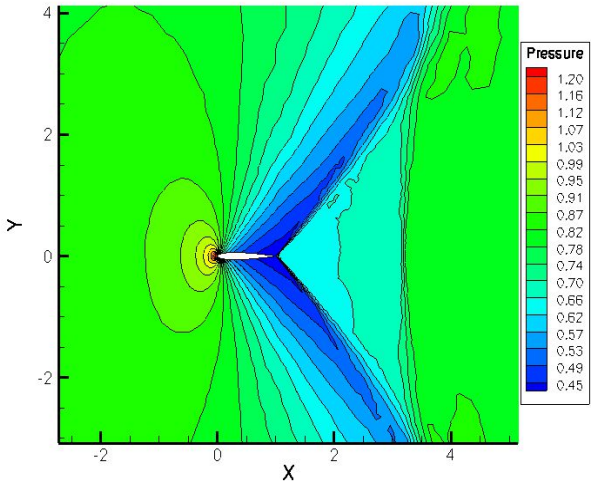


Fig. 76b) : Pressure lines for 2nd refinement.

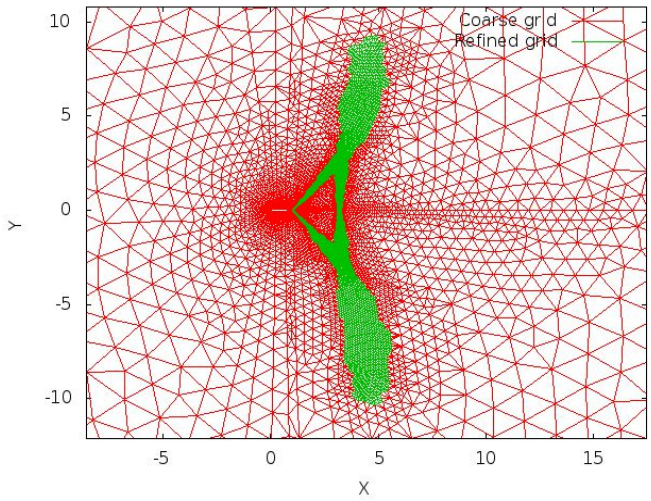


Fig. 77a) :3rd refinement ,number of tri-elements 44487. Inviscid flow over Naca0012 ,Ma_∞ =0.95 ,angle =0˚.

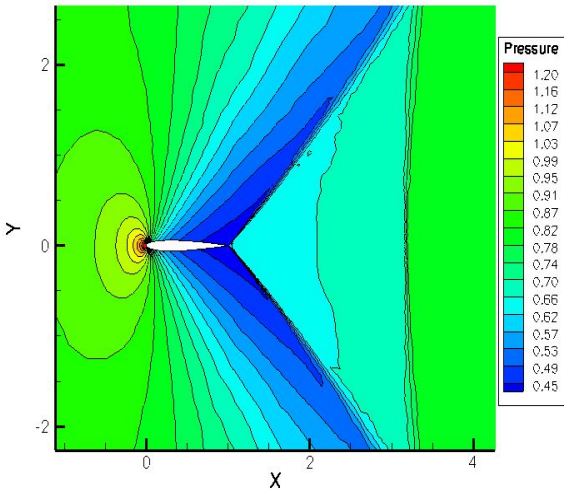


Fig.77b) :Pressure lines for 3rd refinement. Pressure value was the main criterion.

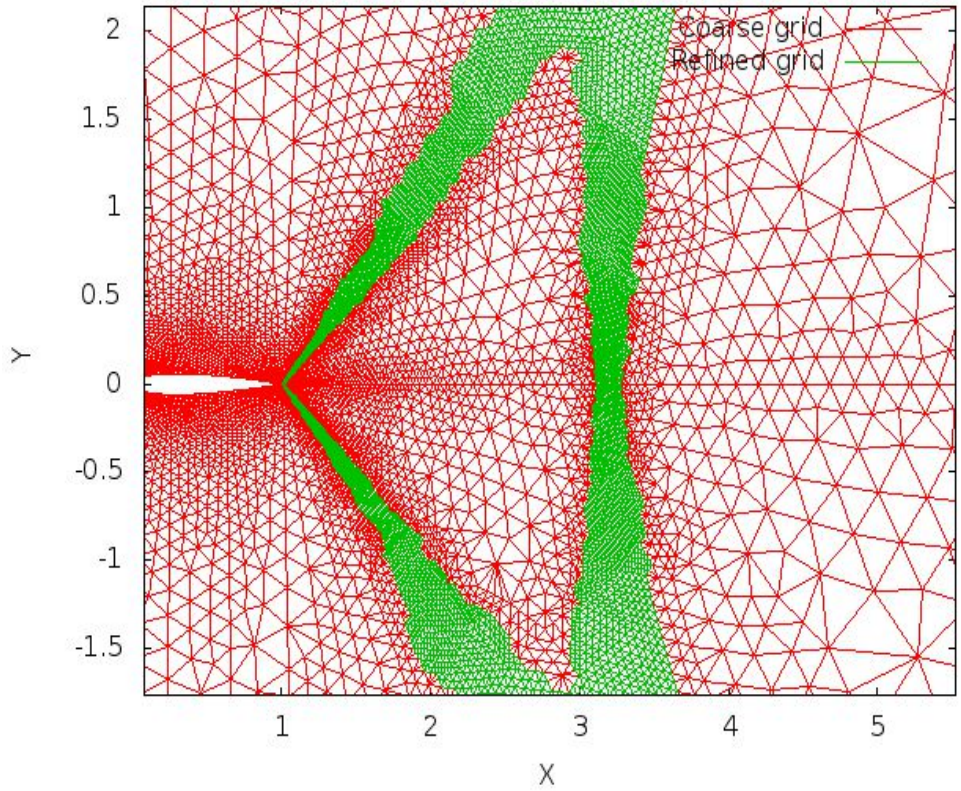


Fig.77c) : The 3rd refinement at a closer look. refinement over areas with large pressure and Mach differences (inviscid case).

Mach number and pressure changes need finer grid at the leading edge and the area of the shock waves. There exist 3 areas where the mesh changes and the rate of refinement was defined to the input to avoid extensive mesh population, and at the same time allowing the solver to select a larger amount of parent cells than previous test cases 8.2% of the initial mesh from 5% before). The number of layers that will be refined and neighbor to the ones selected from flow criteria, is also reduced.

The pressure coefficient, lift coefficient and mean error of density through time evolution compared to the non refined case are following (fig. 78-80):

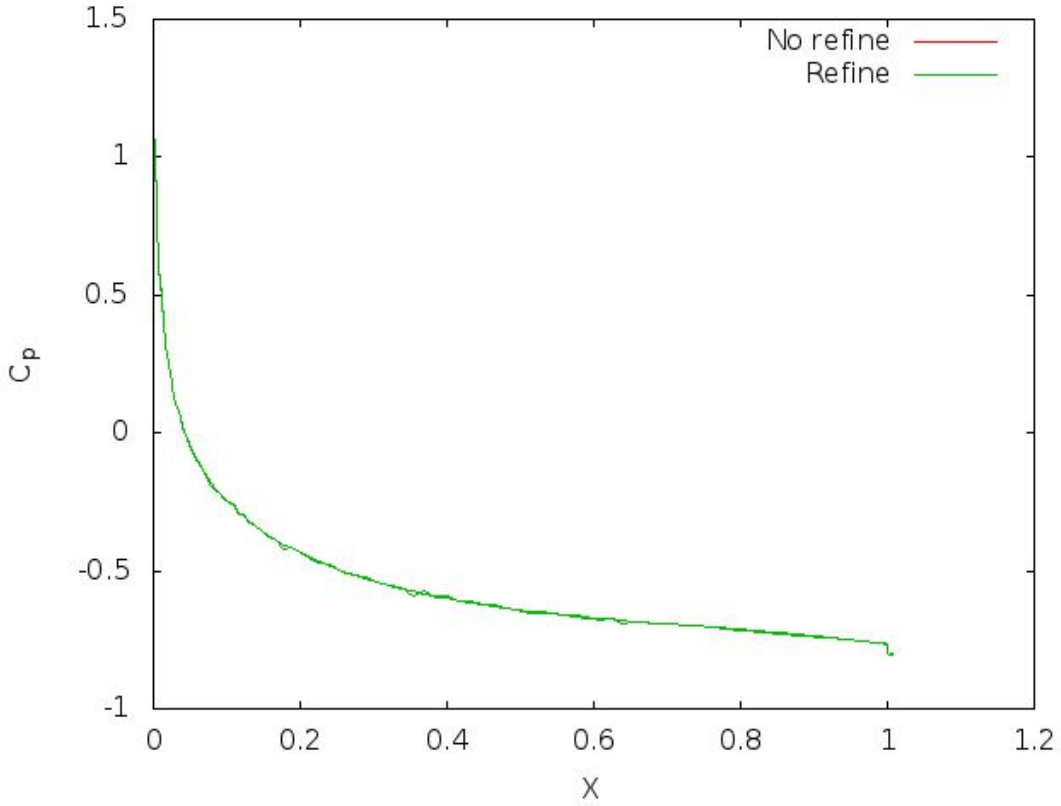


Fig. 78 :Pressure coefficient for inviscid flow over Naca0012 ,Ma_∞ =0.95 ,angle =0˚ (both refined and non-refined grid cases).

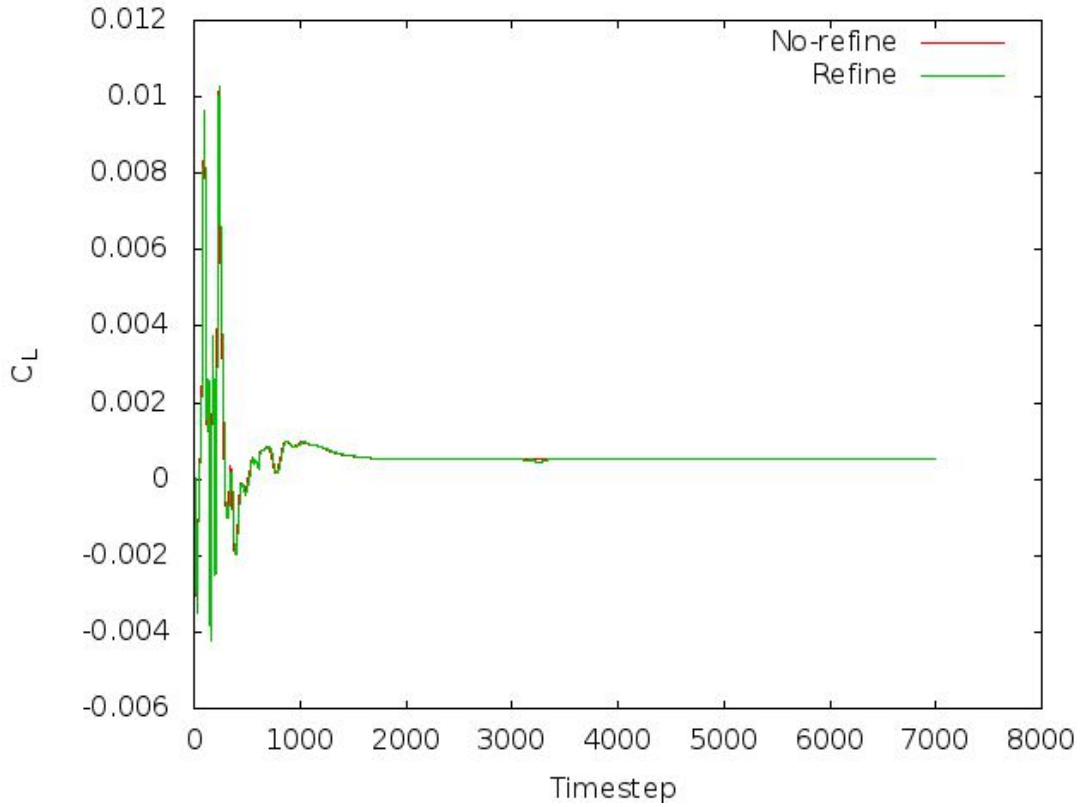


Fig. 79) :Lift coefficient through convergence history for inviscid flow over Naca0012, Ma_∞ =0.95, angle =0˚.

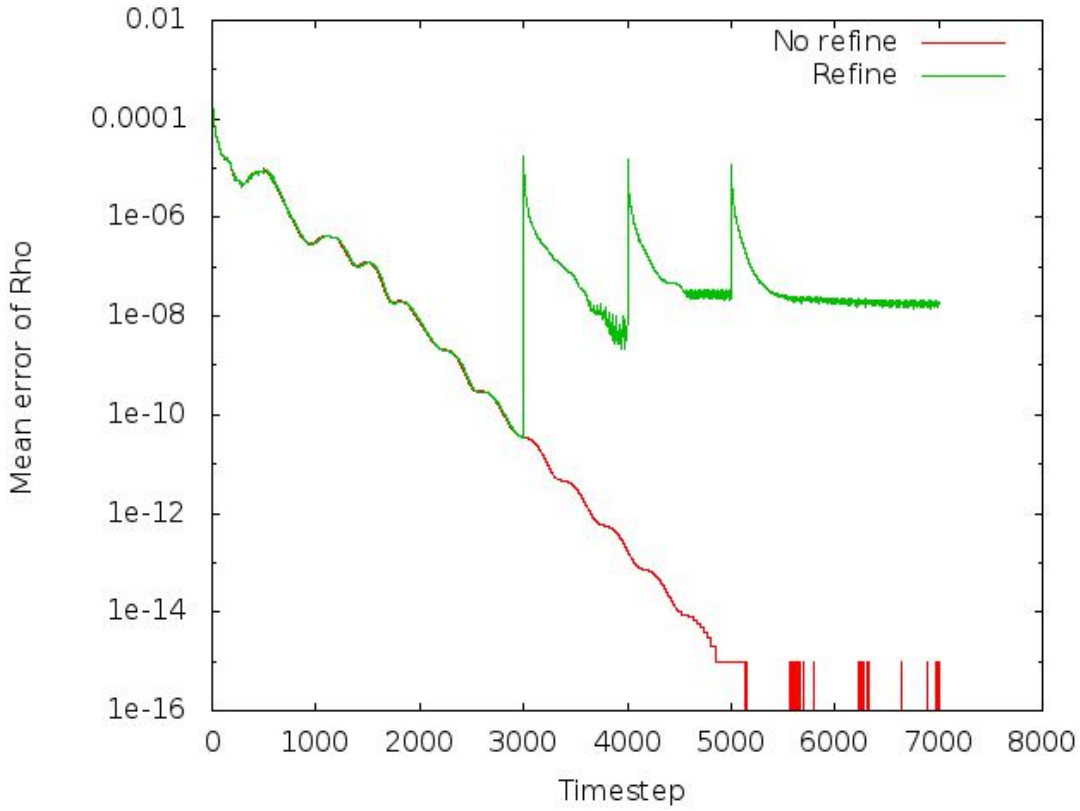


Fig.80) : Mean error of density through convergence history for inviscid flow over Naca0012 , Ma_∞ =0.95, angle =0˚.Error is almost equal after each one of the refinements.

Because of the zero angle the pressure distribution is as expected to remain the same (fig. 78). The same holds for the lift coefficient (fig. 79).

Differentiations appear only at the diagram of mean error of density. The convergence level is satisfying and approximates the value of 1e-08. The final grid has 44.487 triangle cells giving us a suitable flow simulation. In this test case convergence is affected by the CFL number, (final CFL), and smaller value was selected to achieve better convergence. The CFL_{final} used was 5 here.

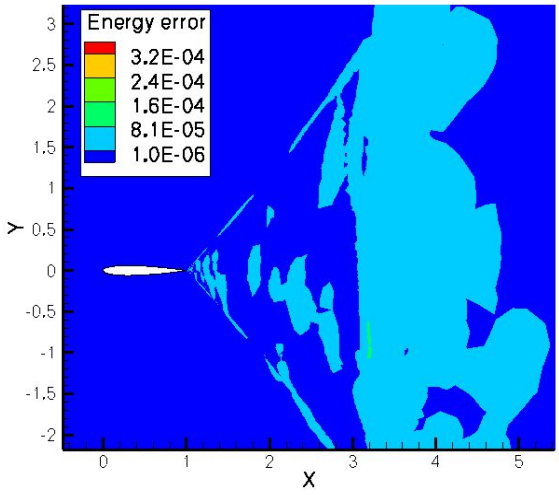


Fig.80a: Error of energy for inviscid flow over Naca0012, Mach=0.95, angle =0.

4.5 Inviscid transonic flow, around two-elements airfoil, $Ma_\infty = 0.185$, angle of inflow = 6°.

At this test case the inviscid flow over the two-elements airfoil NLR -7301, was studied. There is no shock wave so the absolute difference of mach number will not be efficient here. Either the pressure or the changes at velocity were rejected to. The refinement criterion was the error indicator of energy of each cell. The cut-off for the selection was a small number around 10d-08, because of the small Ma_∞ number of the flow. The initial grid was firstly designed and then refined has 9016 triangle cells.

The new grids are presented below, for each refinement (fig. 81a-82a-83a-84a). Grid changes are applied where the energy error is larger (fig. 81b-82b-83b-84b) and is mostly between the main wing and the flap.

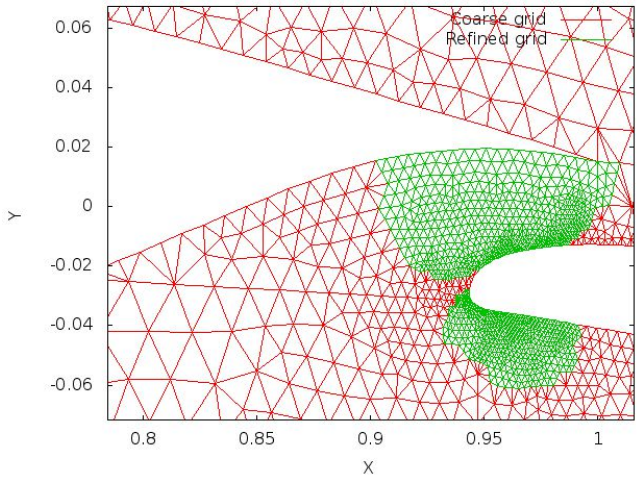


Fig. 81a): 1st refinement ,number of elements 10578 (all tris). Inviscid flow over NLR-7301 airfoil ,Ma_∞ =0.185 ,angle =6˚.

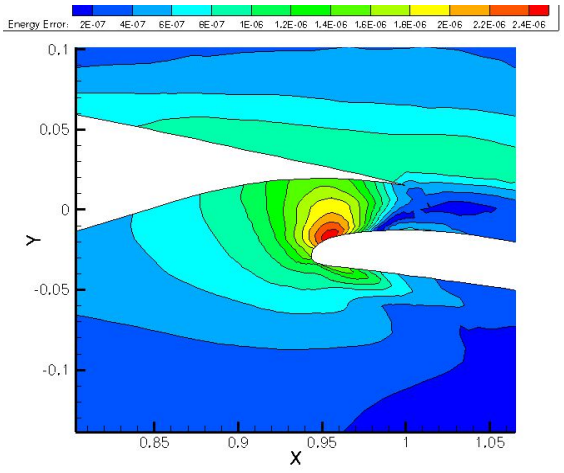


Fig.81b) :Energy error distribution between The 2 elements (error criterion).

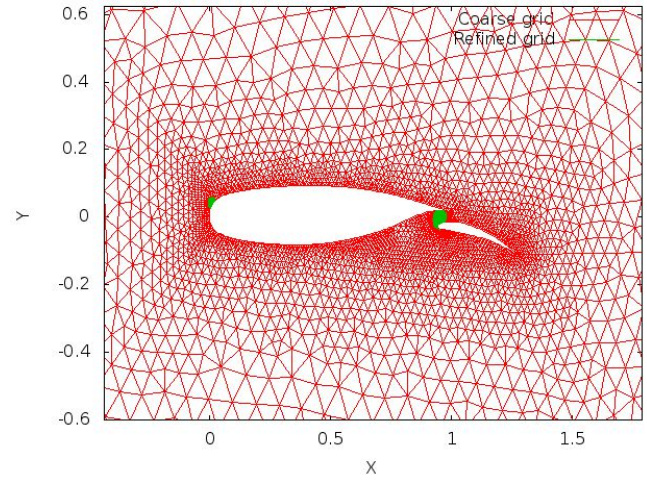


Fig.82a) : 2nd refinement for inviscid flow over NLR-7301 , number of elements 12929 (all tris). Ma_∞ =0.185 ,angle 6˚.

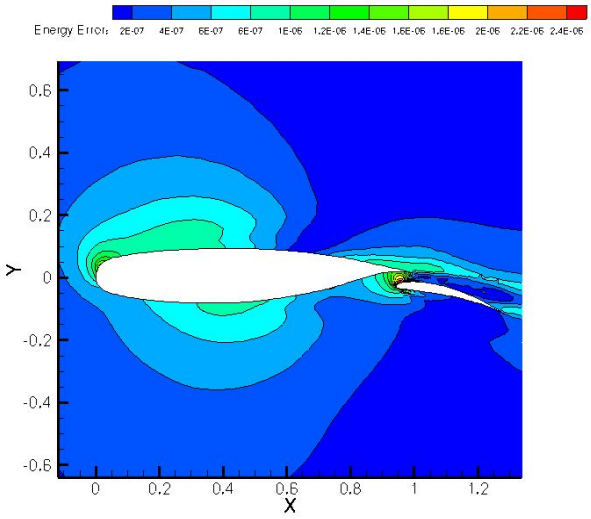
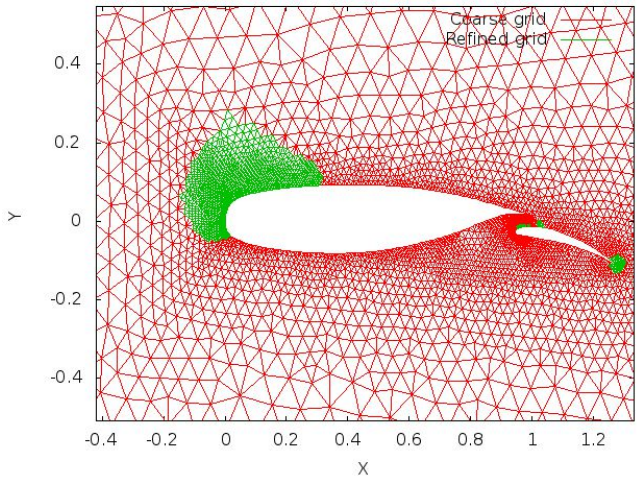


Fig.82.b) Error of energy distribution around The airfoil for 2nd refinement.



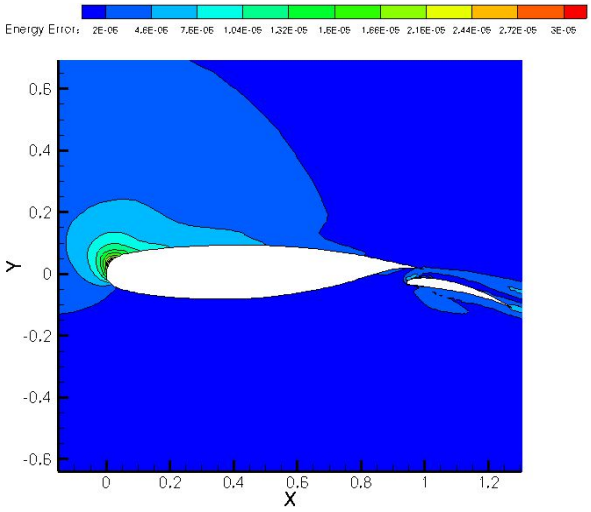
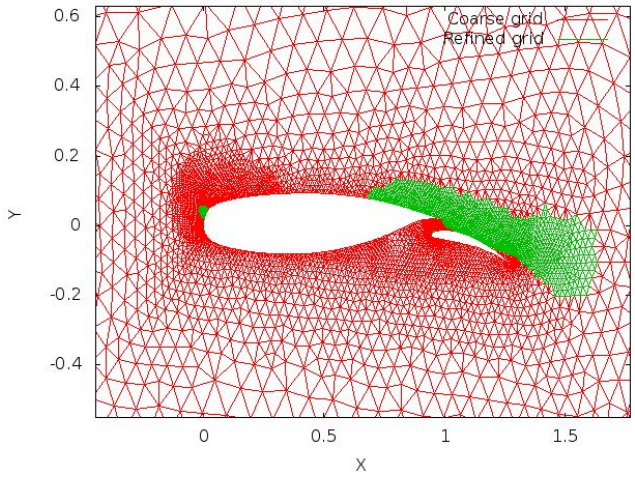


Fig.83a): 3rd refinement, number of elements 17906 ,all tris. Inviscid flow over NLR-7301 ,Ma_∞ =0.185 ,angle=6˚.

Fig.83b): Error of energy (3rd refinement).



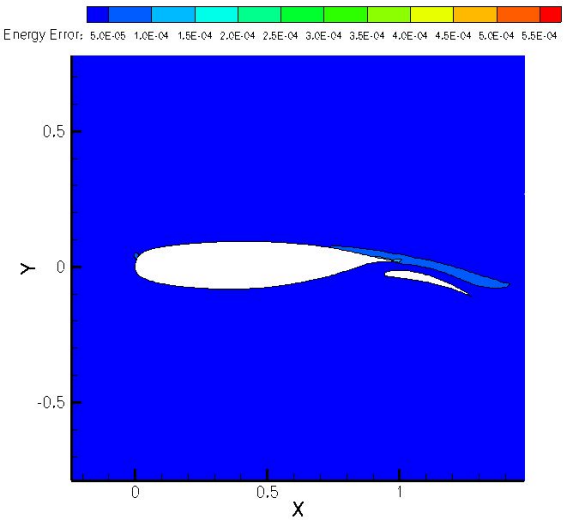


Fig. 84a)：4th refinement ,number of elements 22678 (all tris). Inviscid flow Ma_∞ =0.185, angle =6˚, NLR-7301.

Fig.84.b): Energy error distribution at 4th refinement, around NLR-7301.

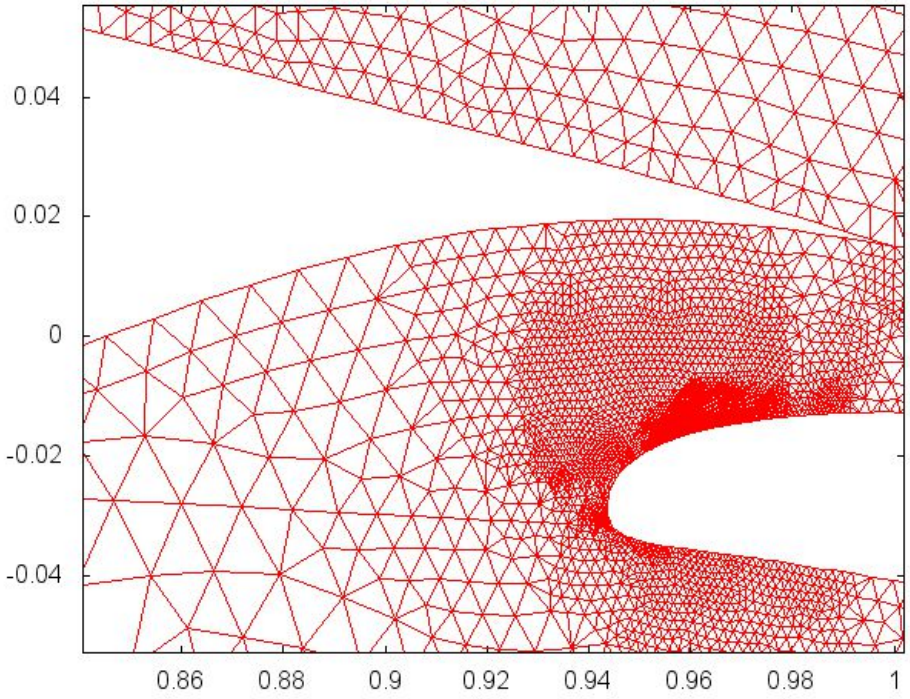


Fig.84 c): Area between the main wing and flap after the 4 refinements. Inviscid flow Ma_∞ =0.185, angle 6˚.

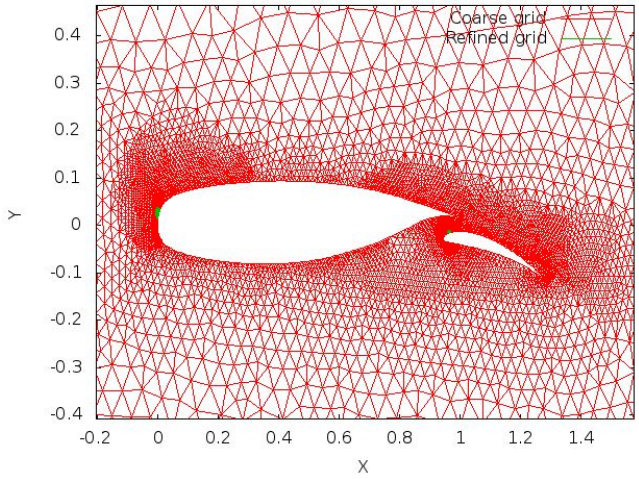


Fig. 85a): 5th refinement, number of 25931 triangle Elements. Refining stagnation point areas, Inviscid flow Ma_∞=0.185, angle=6˚.

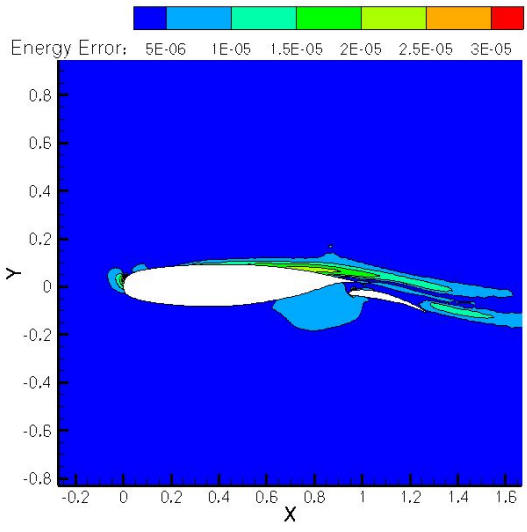


Fig.85.b): Error energy around NLR-7301.

The pressure, lift coefficient and the mean error of density through time evolution compared to the non refined case, are the following (fig. 86,87,88):

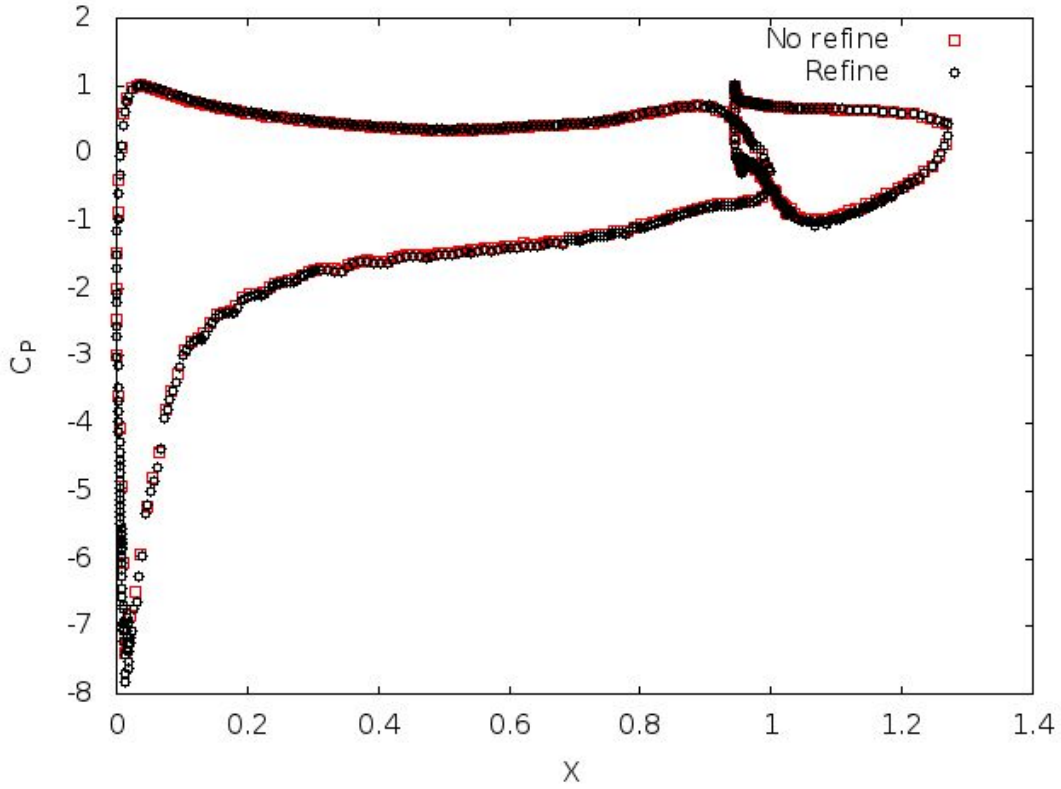


Fig.86) : Pressure coefficient around NLR-7301 airfoil with flap. Inviscid flow, Ma_∞ =0.185 ,angle=6˚.

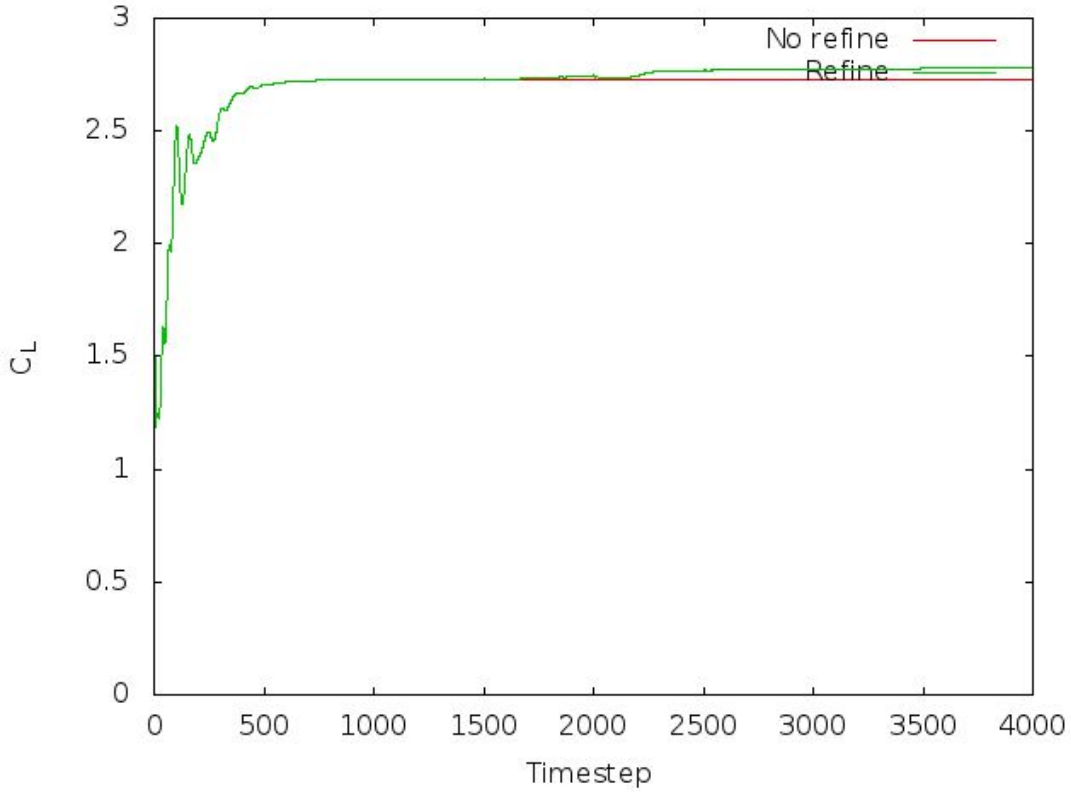


Fig.87) :Lift coefficient for refined ,non-refined grid cases .Inviscid flow, Ma_∞ =0.185, angle=6˚, NLR-7301.

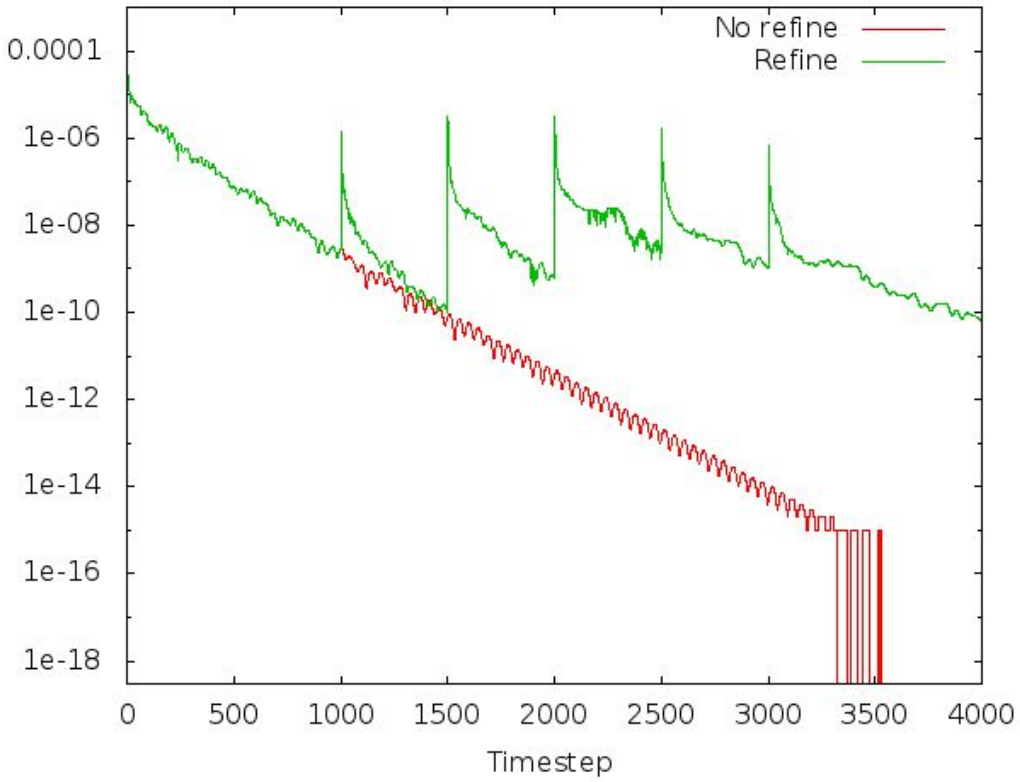


Fig.88): Mean error of density for inviscid flow around NLR-7301 airfoil at Ma_∞ =0.185, angle =6˚.

The pressure distribution (fig. 86) does not change after the refinements (changes happen only momentarily until it convergences again). Moreover the wiggles of previous unstructured cases do not occur. The lift at NLR-7301 (fig. 87) increases after the refinement with respect to the non refined case. It also increases compared to the lift of Naca0012, something that is expected as far NLR-7301 concerned, because it has two elements. The mean error of density convergence in quite acceptable levels after each refinement and at the end it is practically zero after 4000 time steps (1e-10). The local resolution is increased and the accuracy of the solution is also high.

5. Summary – Conclusions

5.1 Resume of the thesis

Summing up, the purpose of this thesis was the design (development and programming) of methods for the refinement at meshes that are used to solve adapted to the flow solver of the laboratory of fluid mechanics of the department of mechanical engineering of NTUA. The flow solver calls the refinement solver that decides if the grid needs changes, selects these areas and improves the grid locally all based on some criteria. These criteria take under consideration the flow parameters and the geometric features that need to fix to smoothly adapt the new mesh (refined) to the old one (coarse). All these features as defined by the user as input. The refinement solver's code

is added to the main program as a subroutine. The speed of the solver's calculations was also a problem that was taken care in order to be fast. A solver that based to a sorting algorithm that was able to fix new nodes and cells was abandoned. The reason was that the method was efficient but too slow.

The code was mainly tested to the airfoil Naca0012 both for structured and structured meshes with triangle and quadrilateral cells and for inviscid flows. It was afterwards extended to any other airfoil and for viscous flows too. The quantities that are used as criteria of refinement may vary for each flow and were studied at different test cases.

5.2 Conclusions

A method for mesh refinement in multiple steps with various criteria was developed and tested. The method applied at triangle of hybrid grids.

Difference flow cases were studied. Behavior of flow simulation with respect to the criteria was also a part of our study.

The procedure of refinement for each level, considering the fact that it is determined and locally applied, differentiates the computational grid at some regions leaving the rest untouched. This may lead to a limited convergence level after a specific value. Either for the triangle or the quadrilateral mesh, applying one or more levels of refinement may provoke changes at the grid smoothness that can be small or large, depending on the flow conditions. In that way, a coarse grid that has Delaunay triangulation is featured with smooth transition from the smaller cells area, to the bigger ones, and as a result so does the smoothness of the solution and consequently eliminating the error during the convergence history.

For the interpolation of the variables at the new cells we applied the simplest interpolation scheme.

Mesh refinement is most of the times necessary for better flow simulation. An important factor is the suitable criterion. Some criteria can be pretty efficient and their variations describe perfectly the phenomena occurring. In test-cases with shock waves differences of mach number and pressure are properly applied. In cases with subsonic flows criteria like mean error may be useful too. for every flow there can be found a quantity that corresponds to the appropriate criterion for the refinement. The combination with two or more of them turns to be pretty efficient.

Refinement equals to higher resolution and detailed simulation in regions of interest. If these areas include shock waves, abrupt increment of temperature, density and pressure occur, non continuously and the energy is conserved. These discontinuities are properly revealed as more cells are developed there. These small finite volumes obtain large error as expected from the physics of shocks. This detail I the flow is not simulated before the current level of refinement and affects the convergence level. Convergence can be reduced at the non refined grid levels depending both on the level of refinement and the flow.

Mesh refinement may affect pressure distribution over the airfoil. In order to prevent unpleasant results the initial grid must be properly constructed. Boundary elements around airfoil body need special treatment.

5.3 Suggestions for future research and improvement

After managing the basic development of the refinement, some much features could be applied and some others be improved. These are the following:

1) Interpolation with higher order schemes for example wider stencil methods.
2) Development of methods that will ensure smooth transition to the regions with new mesh. The computational direction from large computational cells to smaller ones is an important issue. This can be achieved with constructing transition layers at specific regions of using the whole refinement area.
3) Development of the opposite procedure of coarsening the grid. Combining such techniques with refinement could lead to less computational load using less cells.
4) Methods of decomposing the mesh to smaller parts and embedded to flow solver of parallel processing increasing the computational capabilities.
5) Some further modifications could be applied for the flow solver to handle meshes with hanging nodes, technique that would improve the smoothness of mesh geometry.
6) Using the ability of the flow solver of the laboratory to 3D finite volumes with various shapes the refinement some extra add-ons could be applied to the solver for refinement in three dimensional space.

NLR -7301 airfoil points

Airfoil Data:					
1.000000	1.494277E-02	8.289919E-02	-5.883842E-02	5.759999E-03	2.657714E-02
9.890835E-01	1.657285E-02	7.389852E-02	-5.681385E-02	8.033172E-03	3.119666E-02
9.772437E-01	1.821649E-02	6.567809E-02	-5.478715E-02	1.098309E-02	3.607763E-02
9.643493E-01	1.917988E-02	5.819098E-02	-5.275245E-02	1.477851E-02	4.111988E-02
9.526795E-01	1.953210E-02	5.140123E-02	-5.067647E-02	1.960687E-02	4.615280E-02
9.428632E-01	1.944147E-02	4.525360E-02	-4.859960E-02	2.570975E-02	5.083963E-02
9.342002E-01	1.905715E-02	3.969373E-02	-4.655543E-02	3.314366E-02	5.493407E-02
9.262452E-01	1.844563E-02	3.469608E-02	-4.450462E-02	4.173582E-02	5.860228E-02
9.186485E-01	1.763614E-02	3.023043E-02	-4.243881E-02	5.141019E-02	6.202954E-02
9.111276E-01	1.663522E-02	2.624240E-02	-4.039916E-02	6.222215E-02	6.514498E-02
9.034036E-01	1.541891E-02	2.270687E-02	-3.837268E-02	7.419923E-02	6.799029E-02
8.951613E-01	1.392956E-02	1.959781E-02	-3.636072E-02	8.736816E-02	7.065430E-02
8.860273E-01	1.199762E-02	1.686958E-02	-3.439751E-02	1.017657E-01	7.322783E-02
8.754532E-01	9.332777E-03	1.448536E-02	-3.248730E-02	1.174505E-01	7.568856E-02
8.625527E-01	5.631745E-03	1.243492E-02	-3.062834E-02	1.344646E-01	7.808837E-02
8.457202E-01	2.918663E-04	1.066920E-02	-2.883932E-02	1.528565E-01	8.038185E-02
8.250030E-01	-6.630416E-03	9.164336E-03	-2.713409E-02	1.726567E-01	8.257949E-02
8.032055E-01	-1.386199E-02	7.872004E-03	-2.552333E-02	1.938884E-01	8.464053E-02
7.803196E-01	-2.124571E-02	6.776264E-03	-2.401453E-02	2.165561E-01	8.655240E-02
7.562906E-01	-2.840810E-02	5.841217E-03	-2.260142E-02	2.406463E-01	8.829530E-02
7.311667E-01	-3.520022E-02	5.053942E-03	-2.129057E-02	2.661264E-01	8.984053E-02
7.050425E-01	-4.155884E-02	4.389001E-03	-2.007844E-02	2.929411E-01	9.115937E-02
6.780237E-01	-4.740670E-02	3.821460E-03	-1.895828E-02	3.210114E-01	9.223486E-02
6.502920E-01	-5.296887E-02	3.339584E-03	-1.793149E-02	3.502357E-01	9.302679E-02
6.219929E-01	-5.827638E-02	2.933059E-03	-1.699346E-02	3.804876E-01	9.350574E-02
5.932566E-01	-6.322600E-02	2.589659E-03	-1.613523E-02	4.116189E-01	9.366567E-02
5.642325E-01	-6.777043E-02	2.299176E-03	-1.534942E-02	4.434609E-01	9.346852E-02
5.350670E-01	-7.179967E-02	2.053096E-03	-1.463093E-02	4.758282E-01	9.290183E-02
5.059035E-01	-7.513048E-02	1.844111E-03	-1.397914E-02	5.085221E-01	9.195519E-02
4.769035E-01	-7.767550E-02	1.665941E-03	-1.338829E-02	5.413325E-01	9.056596E-02
4.482392E-01	-7.948611E-02	1.513502E-03	-1.285277E-02	5.740483E-01	8.873536E-02
4.200733E-01	-8.068644E-02	1.361465E-03	-1.228370E-02	6.064566E-01	8.644214E-02
3.925499E-01	-8.138816E-02	1.200926E-03	-1.164557E-02	6.383487E-01	8.364213E-02
3.657953E-01	-8.167103E-02	1.033757E-03	-1.092960E-02	6.695254E-01	8.030787E-02
3.399167E-01	-8.160332E-02	8.620141E-04	-1.012643E-02	6.998045E-01	7.644492E-02
3.150049E-01	-8.122198E-02	6.870911E-04	-9.225646E-03	7.290323E-01	7.213821E-02
2.911358E-01	-8.052865E-02	5.117796E-04	-8.215961E-03	7.570868E-01	6.751481E-02
2.683639E-01	-7.961497E-02	3.450535E-04	-7.083839E-03	7.838694E-01	6.268139E-02
2.467317E-01	-7.847898E-02	1.982805E-04	-5.815168E-03	8.093137E-01	5.776541E-02
2.262651E-01	-7.714084E-02	7.705308E-05	-4.394514E-03	8.333760E-01	5.287363E-02
2.069737E-01	-7.565740E-02	-8.950266E-06	-2.806079E-03	8.560349E-01	4.810045E-02
1.888557E-01	-7.404825E-02	-4.294989E-05	-1.032463E-03	8.772870E-01	4.351589E-02

1.718975E-01	-7.232996E-02	-1.101520E-05	9.456107E-04	8.971339E-01	3.912094E-02
1.560785E-01	-7.051211E-02	9.615486E-05	3.148192E-03	9.156092E-01	3.498605E-02
1.413642E-01	-6.864120E-02	3.129058E-04	5.595359E-03	9.327445E-01	3.110611E-02
1.277138E-01	-6.674758E-02	6.520439E-04	8.310100E-03	9.485875E-01	2.749631E-02
1.150907E-01	-6.480268E-02	1.148581E-03	1.131453E-02	9.631886E-01	2.414229E-02
1.034441E-01	-6.284557E-02	1.840017E-03	1.463071E-02	9.766068E-01	2.103514E-02
9.273168E-02	-6.084649E-02	2.771303E-03	1.827920E-02	9.888897E-01	1.810574E-02
8.289919E-02	-5.883842E-02	4.037316E-03	2.226594E-02	1.000000	1.494277E-02
Flap Data:					
1.273393	-1.104745E-01	9.479136E-01	-3.380343E-02	1.043060	-1.616922E-02
1.270088	-1.090811E-01	9.471037E-01	-3.355524E-02	1.049877	-1.725057E-02
1.266425	-1.075376E-01	9.464064E-01	-3.329406E-02	1.056911	-1.852098E-02
1.262380	-1.058324E-01	9.458246E-01	-3.297840E-02	1.064148	-1.997013E-02
1.257924	-1.039542E-01	9.453375E-01	-3.263118E-02	1.071568	-2.159573E-02
1.253033	-1.018876E-01	9.449276E-01	-3.227050E-02	1.079153	-2.340092E-02
1.247724	-9.953219E-02	9.446073E-01	-3.192012E-02	1.086885	-2.536271E-02
1.241912	-9.702086E-02	9.443691E-01	-3.159151E-02	1.094741	-2.747372E-02
1.235538	-9.445485E-02	9.441875E-01	-3.128440E-02	1.102697	-2.972987E-02
1.228616	-9.177712E-02	9.440242E-01	-3.095187E-02	1.110725	-3.212643E-02
1.221150	-8.896457E-02	9.438668E-01	-3.056931E-02	1.118795	-3.466315E-02
1.213135	-8.604755E-02	9.437186E-01	-3.014057E-02	1.126877	-3.733918E-02
1.204571	-8.306723E-02	9.435969E-01	-2.968356E-02	1.134942	-4.014018E-02
1.195472	-8.005244E-02	9.435130E-01	-2.918571E-02	1.142960	-4.305886E-02
1.185871	-7.704662E-02	9.434801E-01	-2.862709E-02	1.150901	-4.609006E-02
1.175804	-7.407776E-02	9.435087E-01	-2.800800E-02	1.158739	-4.921097E-02
1.165340	-7.114610E-02	9.436079E-01	-2.736076E-02	1.166456	-5.239050E-02
1.154533	-6.830575E-02	9.437980E-01	-2.668866E-02	1.174025	-5.562151E-02
1.143460	-6.557547E-02	9.440843E-01	-2.598812E-02	1.181425	-5.888415E-02
1.132209	-6.296426E-02	9.444554E-01	-2.525102E-02	1.188640	-6.215837E-02
1.120868	-6.048834E-02	9.449388E-01	-2.448526E-02	1.195645	-6.544030E-02
1.109525	-5.816614E-02	9.455310E-01	-2.370667E-02	1.202428	-6.871004E-02
1.098277	-5.599114E-02	9.462142E-01	-2.292551E-02	1.208985	-7.194100E-02
1.087210	-5.396334E-02	9.470331E-01	-2.214002E-02	1.215295	-7.513099E-02
1.076403	-5.209191E-02	9.479682E-01	-2.135598E-02	1.221350	-7.827815E-02
1.065933	-5.036100E-02	9.490342E-01	-2.058910E-02	1.227134	-8.138465E-02
1.055861	-4.876733E-02	9.502523E-01	-1.984693E-02	1.232650	-8.443324E-02
1.046240	-4.731019E-02	9.516004E-01	-1.912832E-02	1.237896	-8.741099E-02
1.037113	-4.596946E-02	9.531094E-01	-1.845255E-02	1.242872	-9.031407E-02
1.028507	-4.473864E-02	9.547587E-01	-1.779737E-02	1.247592	-9.311900E-02
1.020442	-4.360468E-02	9.565703E-01	-1.718203E-02	1.252053	-9.583124E-02
1.012923	-4.258190E-02	9.585384E-01	-1.659133E-02	1.256180	-9.856313E-02
1.005952	-4.163947E-02	9.606791E-01	-1.604644E-02	1.260035	-1.012304E-01
9.995188E-01	-4.077372E-02	9.629951E-01	-1.554100E-02	1.263677	-1.037512E-01
9.936075E-01	-3.998842E-02	9.654965E-01	-1.508836E-02	1.267114	-1.061291E-01
9.881976E-01	-3.927529E-02	9.681882E-01	-1.467614E-02	1.270351	-1.083687E-01
9.832653E-01	-3.862612E-02	9.710785E-01	-1.430412E-02	1.273393	-1.104745E-01

9.787833E-01	-3.803542E-02	9.741742E-01	-1.396085E-02
9.747232E-01	-3.749824E-02	9.774851E-01	-1.365307E-02
9.710554E-01	-3.700950E-02	9.810190E-01	-1.337730E-02
9.677498E-01	-3.656863E-02	9.847855E-01	-1.313738E-02
9.647765E-01	-3.617889E-02	9.887918E-01	-1.293583E-02
9.621068E-01	-3.583968E-02	9.930463E-01	-1.278068E-02
9.597144E-01	-3.554074E-02	9.975560E-01	-1.269434E-02
9.575750E-01	-3.527346E-02	1.002326	-1.270015E-02
9.556643E-01	-3.503386E-02	1.007360	-1.281134E-02
9.539602E-01	-3.481982E-02	1.012659	-1.303282E-02
9.524435E-01	-3.462267E-02	1.018222	-1.338758E-02
9.510958E-01	-3.443102E-02	1.024049	-1.387176E-02
9.498966E-01	-3.425515E-02	1.030136	-1.448431E-02
9.488385E-01	-3.405068E-02	1.036476	-1.524949E-02

6. References

[1] Computartional Fluid Mechanics ,G. Bergeles, Athens 2006.
[2] Design and testing of a multiblock Grid-Generation procedure for aircraft design and research, NLR, Advisory group for aerospace research and development, 1990.
[3] Unstructured Mesh Related Issues In Computational Fluid Dynamics (CFD) – Based Analysis And Design ,11th International Meshing Roundtable ,September 15-18, 2002.
[4] R-refinement grid adaptation algorithms and issues ,D.Scott McRae .Comput. Methods Appl. Mech. Engrg. Volume 189 issue 4 (2000)
[5] Popov E.V. Geometrical Modeling of Tent Fabric Structures with the Stretched Grid Method. Proceedings of the 11 th International Conference on Computer Graphics&Vision GRAPHICON'2001, UNN, Nizhny Novgorod, 2001. P.138-143.
[6] Rainald Lohner ,Joseph d. Baum Adaptive h-refinement on 3D unstructured grids for transient problems ,INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 14,1407-1419 (1992).
[7] Yi Li, *Sa*chin Premasutha ,Antony Jameson ,Comparison of *h*- and *p*- Adaptations for Spectral Difference Methods 40th Fluid Dynamics Conference and Exhibit 28 June - 1 July 2010, Chicago, Illinois.
[8] J. Peraire, M. Vahdati, K. Morgan, O.C. Zienkiewic .Adaptive remeshing for compressible flow computations , J. Comp. Phys., 72 (1987), pp. 449–466.
[9] S. Rippa. Long and thin triangles can be good for linear interpolation. SIAM J. Numer. Analysis, 29:257{270, 1992.
[10] D.J. Mavripils . Unstructured and adaptive mesh generation for high Reynolds number viscous flows .ICASE Report 91-25 ,NASA Langley Research Center ,1991.
[11] A comparison of Sequential Delaunay Triangulation Algorithms, Peter Su,Robert L.Scot Drysdale.
[12] Fluent6 documentation.
[13] : NASA Tech Briefs, September 2008 , Spline-Based Smoothing of Airfoil Curvatures
[14] Numerical solving of Navier-Stokes equations using unstructured and structured meshes – int parallel environment. D. Koumpogiannis NTUA 1998.
[15] Jan Homann, Adaptive Mesh Refinement, 2005 Why Adaptive Mesh.Multi dimensional solution of hyperbolic system of conservation laws often too time consuming.