Adaptive Mesh Refinement Finite Volume solution for the Boundary Layer problem around an airfoil

Sanderson L. Gonzaga de Oliveira, Mauricio Kischinhevsky

Universidade Federal Fluminense, Passo da Pátria St.,156, Bl. E, 24.210-240, Niterói, RJ, Brazil

Phone/Fax: 55-21-2629-5625/5627 {sgonzaga,kisch}@ic.uff.br

Abstract: In physics and fluid mechanics, the boundary layer is the fluid layer in the immediate vicinity of a bounding surface. It is important in many aerodynamic problems. This work presents a numerical simulation of the two-dimensional laminar boundary-layer problem considering a steady incompressible flow with no-slip condition on the surface. The adaptive mesh refinement is performed by Autonomous Leaves Graph in the Finite Volume solution. A modified Hilbert curve algorithm is used to connect and provide the ordering of the graph nodes. Initially, the numerical solution for the flat plate problem is compared to its analytical solution, namely Blasius solution. Next, simulations of the flux around a NACA airfoil shape are presented. Computer experiments show that an adaptive mesh refinement using Autonomous Leaves Graph with the modified Hilbert curve ordering is appropriate for an aerodynamic problem. Finally, results illustrate that the method provides a good trade-off between speed and accuracy.

Keywords: Finite Volume method, adaptive mesh refinement, boundary layer, NACA airfoils, space-filling curves.

1 – Introduction

Boundary layers have been of great importance in the study of viscous fluid flow. In 1904, Ludwig Prandtl made the biggest breakthrough by demonstrating the existence of a thin boundary layer in fluid flow. Moreover, he found that there exists a thin layer near an object surface, where the viscous aerodynamic forces are as important as the inertial forces (Venkatachari, 2005). In other words, Prandtl was the first to realize that the relative magnitude of the inertial and viscous forces changed from a layer very near the surface to a region far from the surface. He first proposed the interactively coupled, namely, two layer solution, which properly models many flow problems. Furthermore, he allowed aerodynamicists to simplify the fluid flow equations by dividing the flow field into two areas. One inside the boundary layer, where viscosity is dominant and the majority of the drag experienced by a body immersed in a fluid is created, and one outside the boundary layer, where viscosity can be neglected without significant effects on the solution. As a result, this allows a closed-form solution to the flow in both areas, which is a significant simplification over the solution to the full Navier-Stokes equations.

By making the boundary layer approximation, the flow is divided into an inviscid portion, which is easily solved by a number of numerical methods. In addition to that, the boundary layer is governed by a specific partial differential equation (PDE). Thus, there are two regions to consider. Firstly, is must be taken into account the boundary layer in which the fluid viscosity, though small, exerts an influence on the fluid motion. Subsequently, the velocity gradient is large in this region. Secondly, in the remaining region no large velocity gradients occur and the fluid viscosity influence is negligible.

Such as boundary-layer problem, many initial value and boundary problems for unsteady PDEs use small scale structures which develop, propagate, decline or disappear when the solution evolutes. Examples include boundary layers in viscous fluids and reaction zones in combustion processes. Consequently, the numerical solution to those problems can be very difficult due to the location, time and also nature of such structures are not ordinarily known in the beginning of the process.

In relation to structures of meshes, a mesh is called structured when all internal points, volumes or elements have the same number of neighbors. Thus, using a structured mesh in order to cover the differential problem is not appropriate in the situations earlier mentioned because those meshes do not evaluate the differential scales of the phenomena being studied. Usually, those meshes are computationally expensive because they may have a large number of points to furnish an accurate solution. On the other hand, techniques that use an adaptive mesh refinement are less computationally expensive in those problems earlier mentioned. Those techniques are robust, reliable, and efficient as well. Furthermore, numerically solving PDEs in an efficient time requires a mesh which its points are more refined in regions where the solution or its derivatives quickly change during evolution of time. In fact, some adaptive mesh refinement strategy is required especially in unsteady problems. Following these aspects, the purpose of this work is to automatically build a coarse mesh where the numerical solution furnishes an appropriate approximation among piecewise control volumes. In addition, its purpose is also to construct a fine mesh where the numerical solution does not supply an appropriate approximation among piecewise control volumes, such as singularities and boundary layers, among others. Moreover, the developed mesh of this work has a smooth transition among neighbor piecewise control volumes which have different refinement levels. The reason for developing such technique is because an adaptive mesh refinement enormously reduces the required number of points to obtain an accurate numerical solution to problems that are almost smooth. And also its purpose is to reach a reconstruction with the desired quality in non-smooth problems.

After this brief introduction to boundary layer and adaptive mesh refinement, section 2 presents more details about boundary layer. Along the same lines, section 3 deals with flat plate boundary layer. Next, section 4 presents the used adaptive mesh refinement scheme in detail. In the following, section 5 treats the numerical method for reconstruction. Afterwards, section 6 describes the Blasius solution. Section 7 describes the Finite Volume reconstruction of the momentum equation of the boundary layer problem. Later, section 8 shows some experimental results. Finally, section 9 draws some conclusions.

2 – Boundary Layer

The boundary layer is a very thin layer of fluid flowing over an object surface, for instance, air over an airfoil. As defined, the molecules that directly touch the object surface are virtually motionless. In addition, each layer of molecules within the boundary layer moves faster than the layer that is closer to the object surface. At the top of the boundary layer, the molecules move at the same speed as the molecules outside the boundary layer. As established, this speed is called the *free-stream velocity* (u_∞). Furthermore, the actual speed at which the molecules move depends upon the object shape, the mass of the fluid going by the object, and on two other important fluid properties, namely, the viscosity, or stickiness, of the fluid, and its compressibility, or springiness, of the fluid, i.e., it is how much it can be compacted. For this reason, the flow outside of the boundary layer reacts to the shape of the boundary layer edge just as the physical object surface. Thus, the boundary layer gives to any object an effective shape that is usually slightly different from the physical shape.

Aerodynamic forces depend on a complex way on the fluid viscosity. As the fluid moves past the object, the molecules right next to the surface stick to the surface. The molecules just above the surface are slowed down in their collisions with the molecules sticking to the surface. These molecules in turn slow down the flow just above them. The farther one moves away from the surface, the fewer the collisions are affected by the object surface. This creates a thin fluid layer near the surface in which the velocity changes from zero at the surface to the free stream value away from the surface. Therefore, this layer is called the boundary layer because it occurs on the fluid boundary.

The details of the flow within the boundary layer are very important for many problems in aerodynamics and aeronautical engineering, including the skin friction drag on an object, the heat transfer that occurs in high speed flight and also wing stall. The boundary layer may also lift off or separate from the body, creating an effective shape much different from the physical object shape and causing a dramatic decrease in lift and increase in drag as well. When this happens, the airfoil stalls.

The definition of the boundary layer thickness is to a certain extend arbitrary because transition from the velocity in the boundary to that outside it takes place asymptotically. This is, however, of no practical importance because the velocity in the boundary layer attains a value which is very close to the external velocity already at a small distance from the wall. Therefore, it is possible to define the boundary layer thickness as that distance from the wall where the velocity differs by one per cent from the external velocity.

The deduction of the boundary layer equations was possibly one of the most important advances in aerodynamics. Using a magnitude analysis order, the well-known governing Navier-Stokes equations of viscous fluid flow can be greatly simplified within the boundary layer. Notably, the PDE characteristic becomes parabolic, rather than the elliptical form of the full Navier-Stokes equations.

This greatly simplifies the solution to the equations. Thus, the Navier-Stokes equations for a two-dimensional steady incompressible flow in Cartesian coordinates are given by Eq.(1-2).

$$\text{Momentum: } u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{dp}{dx}+\nu\frac{\partial^2 u}{\partial y^2} \qquad (1)$$

$$\text{Continuity:} \frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \qquad (2)$$

where p is pressure, u is the PDE dependent variable, ρ is the fluid density and v is the kinematic viscosity. Regarding left-hand side of Eq.(2), it is the convective term. Considering first part of right-hand side of Eq.(2), it is one example of source term. When it comes to second part of right-hand side of Eq.(2), it is the diffusion term.

3 - Flat Plate Boundary Layer

Consider now a thin flat plate, immersed in a uniform steady stream of viscous fluid, whose undisturbed velocity u_∞ is perpendicular to the sharp leading edge and parallel to the plate surface. Above the surface, the flow velocity increases in the y direction until, for all practical purposes, it equals the free stream velocity. More precisely, Δ is defined as that distance above the wall where $u_e=0.99u_\infty$; here, u_∞ is the velocity at the outer boundary layer edge. In fact, the quantity Δ is called the velocity boundary-layer thickness (Anderson, 1991). Afterwards, Fig. 1 sketches the process of a flat plate boundary layer. We consider a two-dimensional no-slip steady flow over a flat plate at 0^o angle of attack considering a laminar incompressible flow without heat transfer. Moreover, the energy equation is not needed to calculate the velocity field for an incompressible flow. Thus, the nonlinear governing PDEs in terms of dimensional variables are given by Eq.(1-2).

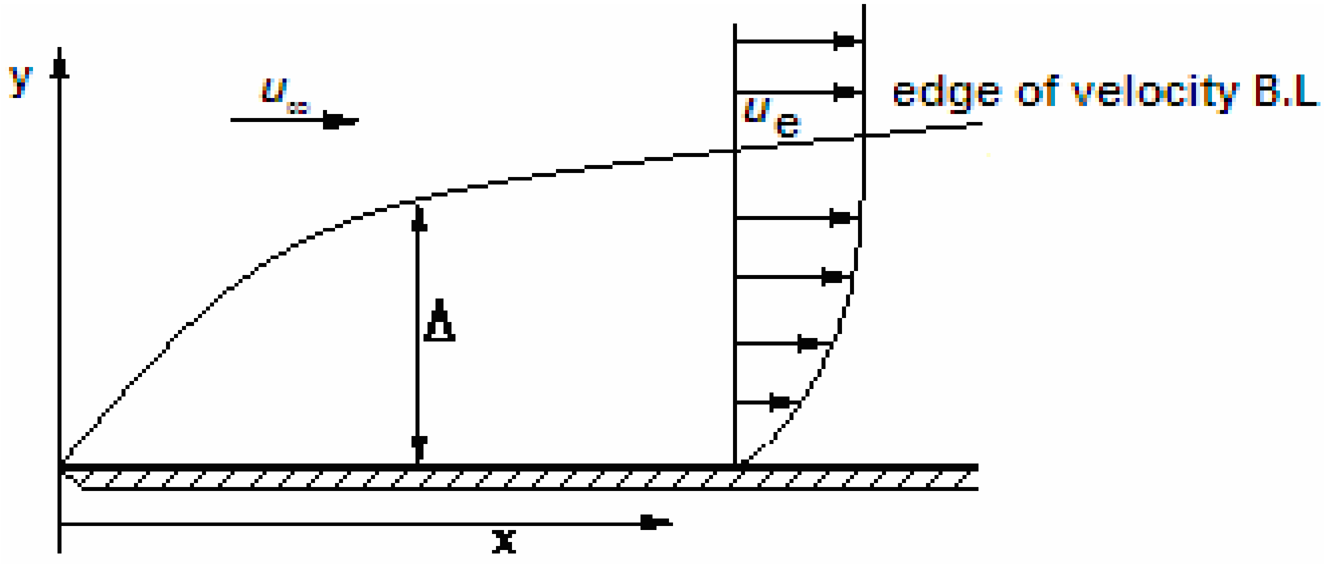


Figure 1: Flat plate boundary layer process - adapted from (Anderson, Tannehill and Pletcher, 1984).

4 – Reconstruction by Finite Volume method with a graph-based adaptive mesh refinement

Burgarelli, Kischinhevski and Biezuner (2006) replaced the tree-based adaptive mesh refinement implementation by a one-level-at-a-time approach, which yields a graph-like implementation in which the children nodes (leaves) become autonomous as their parent node is deleted. Furthermore, a graph-based implementation for dealing with adaptive mesh refinement in the numerical solution to evolutionary PDEs is employed. That is to say that the technique presents a plug-in feature that allows replacement of a group of nodes, which represent control volumes of the mesh, in any region of interest for another one with any refinement level. To be precise, it manages only local changes that occur in the data structure without needing rebuild the whole mesh. In addition, it was also especially designed to minimize the number of operations needed in the adaptive mesh refinement. Therefore, this scheme implementation allows flexibility in the adaptive mesh refinement. Besides that, storage requirements and computational cost compare competitively with hierarchical tree-based adaptive mesh refinement schemes. Moreover, low storage is achieved because only the children nodes are stored when a refinement takes place. Thus, a graph data structure connects nearest-neighboring children nodes through direct links. Furthermore, neighboring volume nodes which were generated from different parent volume nodes can be directly linked whether they have the same refinement level. On the other hand, they are indirectly linked through a transition node in case they have different refinement levels. Thus, Autonomous Leaves Graph contains two linked node types: *volume nodes* which correspond to mesh control volumes, and *transition nodes*, which are used to connect volume nodes in different refinement levels. In a two-dimensional implementation, each volume node has four oriented pointers along corresponding directions. In fact, those pointers can point either to neighboring volume nodes or transition nodes.

By now, consider the unit square as an initial mesh consisting of four control volumes inside the square. Afterwards, each control volume is identified by its center as shown in Fig. 2a. As nodes represent control volumes, each node has oriented links along the four directions. As a result, it has the scheme presented on Fig. 2b. Similarly, the remaining links that do not point to one of the four volume nodes in the square are then directed to the four transition nodes displayed in Fig. 2c as white circles. As shown in Fig. 2d, those non-used links of the four transition nodes are null pointers. For convenience, the two arrows connecting two nodes are replaced by one single line in order to simplify the graphical presentation as depicted in Fig. 2e. Since four lines depart and also arrive from each node, such graph may be used to represent the basic links for any square control volume. As an example, a parent volume node is represented in Fig. 2f. In a certain refinement when the parent volume node is deleted, each child node owns an identifier for a possible unrefinement process. The reason for such identifier is therefore to determine that the four volume nodes belong to the same group. In other words, this group, represented in Fig. 2g, is formed by the four volume nodes with the same refinement level, i.e., they were originated from the same parent volume node.

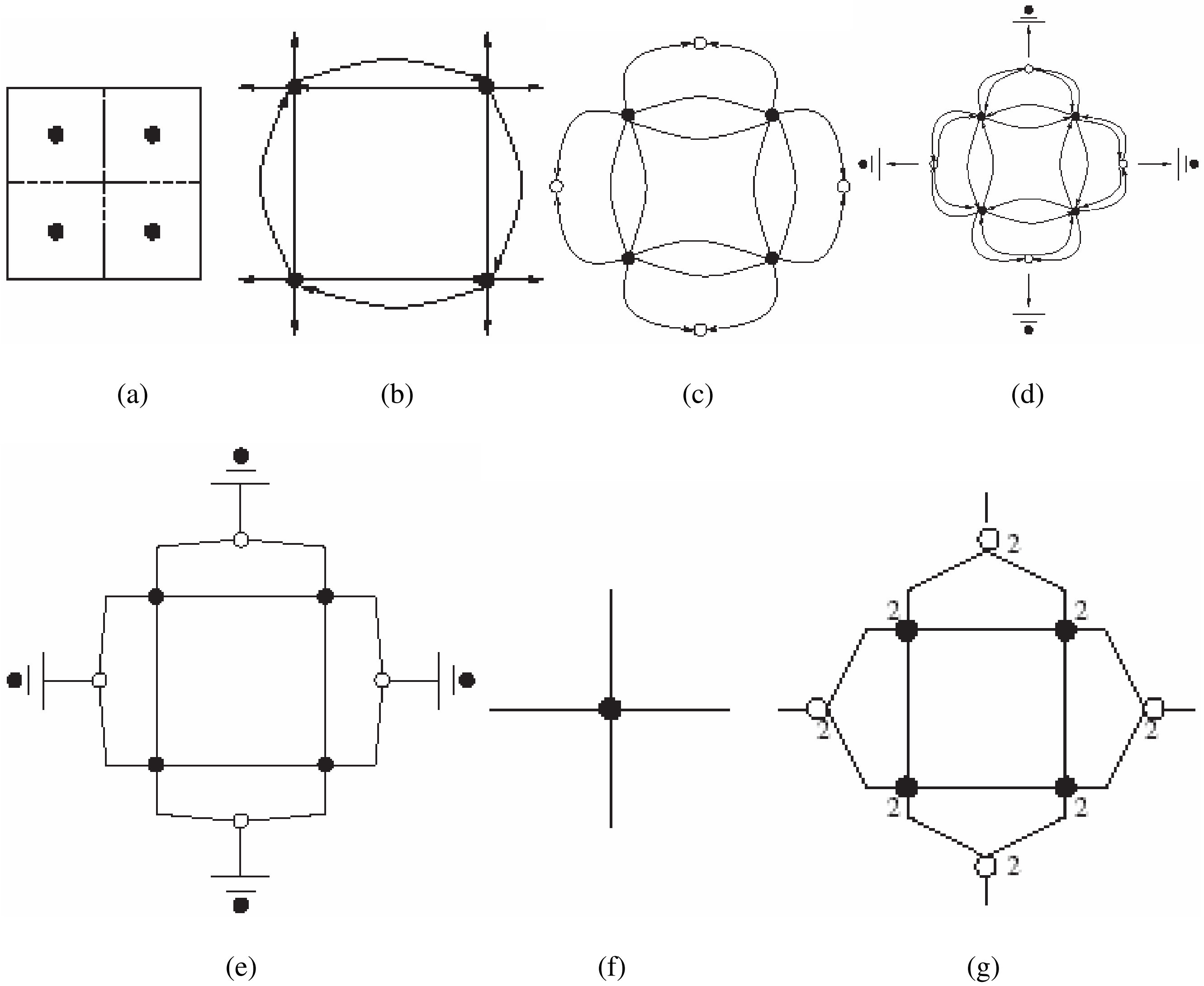


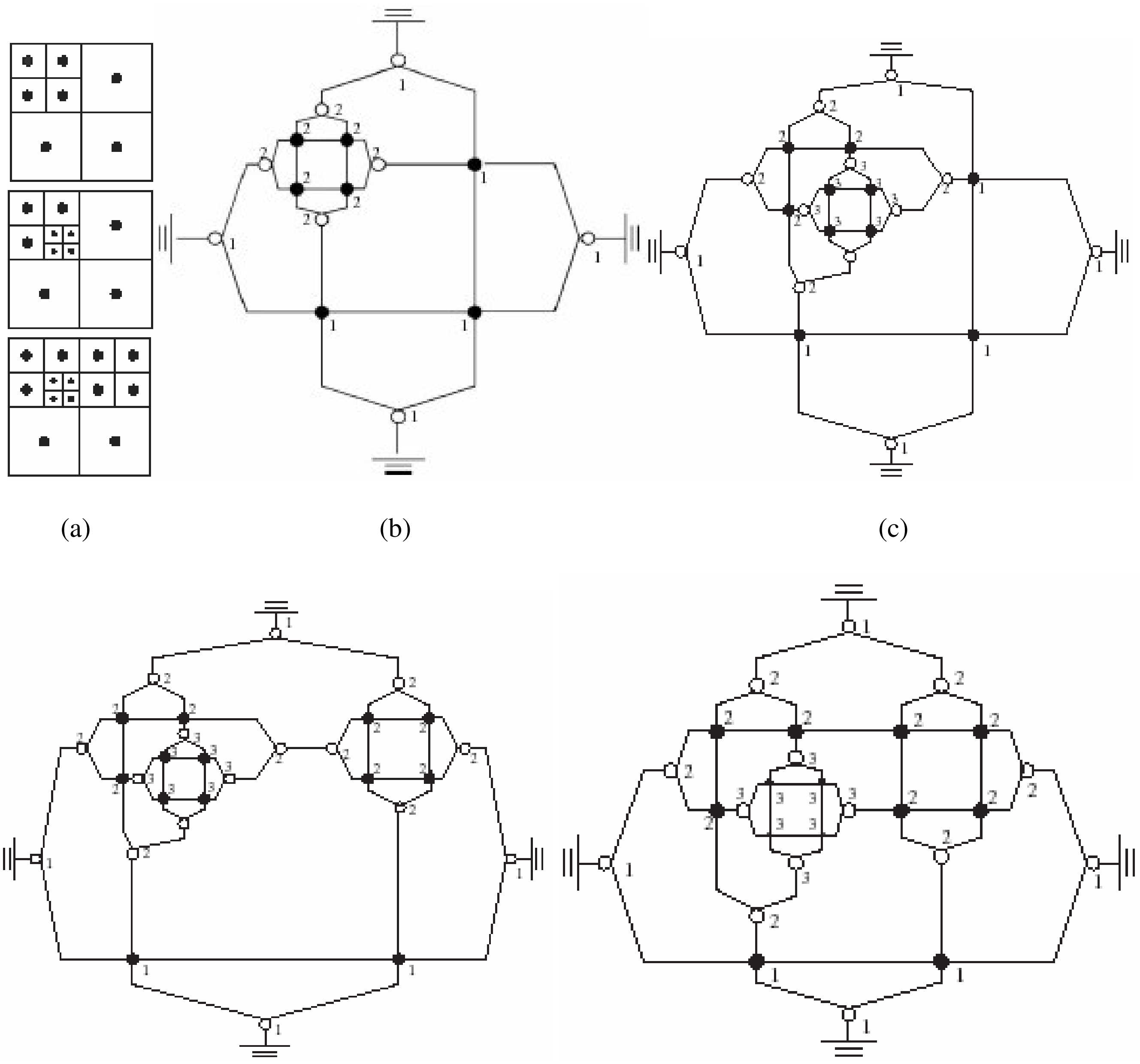
Figure 2: (a) Unit square; (b) Graph structure links; (c) Defining transition nodes; (d) Initial full scheme; (e) Non-directional scheme; (f) A single volume node of the graph and; (g) Elementary graph for refinement (Burgarelli, Kischinhevski and Biezuner, 2006).

Control volumes in Fig. 3a are assigned for refinement by a certain criterion determined by problem in order to verify how a refinement is performed. In addition to that, a control volume configuration as shown in Fig. 3a is created. So, refinement is implemented through replacing the basic structure that represents the links of the volume node to be refined (Fig. 2f) by the bunch structure created (Fig. 2g). To be precise, the outward links departing from the transition nodes are connected to the volume nodes at which the four links of the volume node being replaced were connected. Subsequently, the resulting graph for the unit square becomes that of Fig. 3b. Next, consider a second additional refinement as the one indicated in Fig. 3a. Likewise, application of the same principle of replacing the local basic unit by the basic refinement unit leads to the structure depicted in Fig. 3c. Going one refinement step further, the last configuration in Fig. 3a corresponds to the graph of Fig. 3d. Then, two adjacent transition nodes appear with the same refinement level in this case, namely 2. Afterwards, this latest graph is simplified by eliminating these two redundant transition nodes. Thus, it leads to the graph depicted in Fig. 3e.

On the whole, the refinement sequence described above illustrates all processes during the grid refinement stage of the algorithm. That is to say that whenever two neighboring transition nodes at the same level are encountered, they are both deleted, simplifying the graph and ensuring that the search algorithm for neighbors of a volume node will work efficiently. Moreover, the search algorithm finds either the immediate neighbor or the transition nodes that connects it with neighbors having different refinement levels.

In relation to memory allocation, its requirements are very low since it is needed only for children nodes created during refinement. Moreover, the updates that occur during refinement, including graph simplification, are very efficient, namely, $O(1)$.

There is an extra pair of pointers in each volume node in order to create a total ordering of all volume nodes. That is to say that starting from the first volume node, it is defined a double linked list connecting all volume nodes. In other words, the four volume nodes generated are properly inserted in the linked list at every time a refinement is made. Thus, since each modification in the list is merely local, an algorithm based on the Hilbert's curve construction, which Figs. 7 and 8 are examples, is used to implement the total ordering of the volume nodes.



(a) (b) (c)

(d) (e)

Figure 3: (a) A sequence of refinements; (b) Refining the northwest control volume; (c) one more level of refinement; (d) Refining the northeast control volume and; (e) Graph simplification (Burgarelli, Kischinhevski, Biezuner, 2006).

5 – Finite Volume method reconstruction of the flat plate boundary layer problem

When it comes to the flat plate boundary layer problem, this studied problem has no pressure gradient flow for such a flow presented in Eq.(1-2) (Neel, 1997). Since pressure is invariant, i.e. *dp/dx=0* because the inviscid flow over a flat plate yields a constant pressure over the surface, Eq.(1) can be written as Eq.(3).

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\nu\frac{\partial^2 u}{\partial y^2} \tag{3}$$

Consider a two-dimensional control volume depicted in Fig. 4. Thus, by integrating Eq.(3) in the control volume, it yields Eq.(4).

$$\int_w^e\int_s^n (u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y})dydx=\int_w^e\int_s^n (\nu\frac{\partial^2 u}{\partial y^2})dydx \tag{4}$$

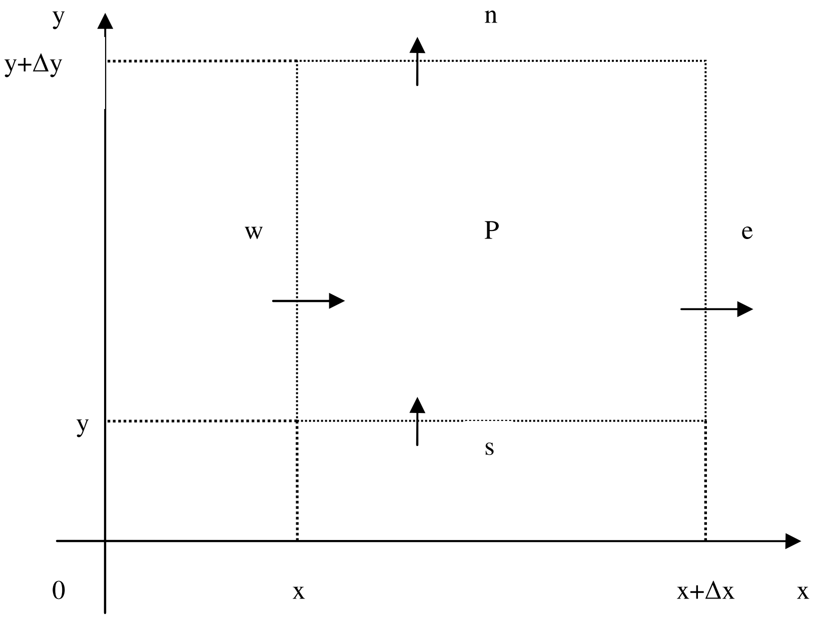


Figure 4: Two-dimensional control volume (adapted from Sperandio, Mendes and Silva, 2004).

Applying the Divergence theorem, Eq.(4) yields Eq.(5).

$$\int_s^n u(u)\cdot \hat{n}_x dy + \int_w^e v(u)\cdot \hat{n}_y dx = \int_w^e \nu' \frac{\partial u}{\partial y}\cdot \hat{n}_y dx \qquad (5)$$

or

$$\int_s^n (uu\,|_e - uu\,|_w)dy + \int_w^e (vu\,|_n - vu\,|_s)dx = \int_w^e (\nu_n \frac{\partial u}{\partial y}|_n - \nu_s \frac{\partial u}{\partial y}|_s)dx \qquad (6)$$

Assuming that the flow in the middle of the control volume edge represents the middle of its variation in the edge (Sperandio, Mendes and Silva, 2004), Eq.(6) can be written as Eq.(7).

$$\Delta y(uu\,|_e - uu\,|_w) + \Delta x(vu\,|_n - vu\,|_s) = \Delta x(\nu_n \frac{\partial u}{\partial y}|_n - \nu_s \frac{\partial u}{\partial y}|_s) \qquad (7)$$

Interpolation functions aim to evaluate the dependent variable value in the control volume edge as well as its derivative. Therefore, the here used interpolation function evaluates the value of a generic property *u* in the control volume interface. Besides that, differencing schemes apply to the convective term linearization, i.e. the convected quantity discretization. As far as differencing schemes are concerned, early attempts to solve advection-diffusion problems applied the Central Differencing scheme (CDS); however it is predominantly diffusive. Hence, solutions exhibited non-physical behavior to problems with predominant advection. Since CDS simply follows the linear interpolation idea, it is the most straightforward discretization of the convected variable. In terms of a Taylor-series expansion, CDS is second-order accurate; nevertheless, it is rarely used nowadays owing to its conditional stability (Madsen, 1998). On the other hand, Upwind Differencing scheme (UDS) is a well-known remedy for the difficulties encountered in CDS. It was first put forward by Courant, Isaacson and Rees in 1952 and subsequently reinvented by Gentry, Martin and Daly in 1966, Brakat and Clark in 1966, and Runchal and Wolfshtein in 1969 (Patankar, 1980). As defined, it consists of setting the volume-face value equal to the nearest volume-center value in the upstream direction. In fact, UDS is only first-order accurate but still an improvement over CDS, as it gets rid upstream propagation of disturbances. Since the here flat plate problem studied flow is from west to east and from south to north in order to evaluate the boundary layer, an UDS is used as presented in Eq.(8).

$$u_P u_P - u_P u_W + v_p u_P - v_p u_S = \frac{\nu}{h}(u_N - 2u_P + u_S) \qquad (8)$$

where $h = \Delta x = \Delta y$ and $\nu = \nu_n = \nu_s$. In the following, Eq.(8) is divided by u_p in order to an easier computational implementation. In addition to that, u_w is considered in the previous iteration. Thus, algebraic manipulations yield Eq.(9).

$$\text{Momentum:}\, u_P^{k+1} - u_N^{k+1}\frac{\nu}{hu_P^k} - u_S^{k+1}\frac{v_P^k + \frac{\nu}{h}}{u_P^k} = u_W^k - 2\frac{\nu}{h} - v_P^k \qquad (9)$$

where ν is the kinematic viscosity, h represents the vertical and horizontal edge size of a control volume. Including, (u_W, v_W), (u_N, v_N), and (u_S, v_S) are the west, north and south neighbors of a control volume (u_P, v_P), respectively. Following the same technique, Eq.(2) is discretized as shown in Eq.(10).

$$\text{Continuity:}\, v_P^{k+1} - v_N^{k+1} = u_W^{k+1} - u_P^{k+1} \qquad (10)$$

To sum up, the here proposed Finite Volume method equations are written in Eq.(9-10), which are semi-implicit numerical approximations of the mathematical modeling given by Eq.(3-2), respectively. Afterwards, Fig. 5 depicts the discretization scheme adopted, which the gray circles represent the control volumes of the Finite Volume mesh.

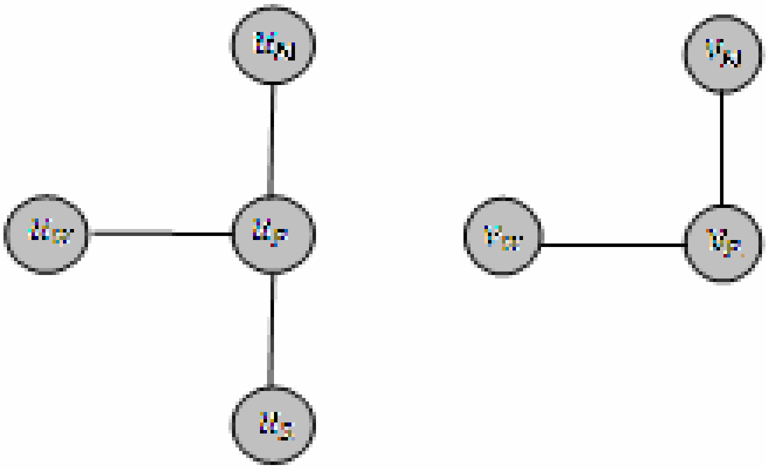


Figure 5: Adopted discretization scheme: momentum and continuity discretization, respectively.

6 - Blasius incompressible laminar flow over a flat plate

This flow problem brings out some fundamental aspects of viscous flow and boundary layer theory. In fact, no pressure gradient exists and a constant boundary layer edge velocity occurs for this flow as well. Moreover, it also gives rise in similar solutions. Furthermore, all the profiles along the plate can be represented by a single curve with the proper dimensionless. For the solution to this constant property, flat plate flow is known as the Blasius solution, which shows that for a flow with Reynolds number (*Re*) much larger than unity, i.e. *Re* >> 1, the velocity profiles have the same dimensionless shape in the boundary layer region. Precisely, it is defined a dimensionless similarity variable in the normal direction given by Eq.(11) to demonstrate this. Subsequently, the velocity parallel to the plate is dimensionless by the edge velocity u_∞, which is also the free-stream velocity for this particular case (Schlichting, 1979). In brief, this work determines the boundary layer height Δ over a flat plate by the analytical Eq.(12).

$$\eta = y\sqrt{\frac{u_\infty}{2\nu x}} \tag{11}$$

$$\Delta = \frac{5x}{\sqrt{\frac{u_\infty x}{\nu}}} \tag{12}$$

where x is the horizontal distance from the flat plate leading edge, u_∞ is the air velocity in free-stream region, and v is the air kinematic viscosity.

7 - Finite Volume reconstruction of the momentum equation of boundary layer problem

According to Anderson (1984), the usual boundary conditions can be applied as $\lim_{y\to\infty} u(x, y) = u_e(x)$ where the subscript *e* refers to conditions at the boundary layer edge. The pressure gradient term in Eq.(1) is to be evaluated from the given boundary information. With $u_e(x)$, specified, *dp/dx* can be evaluated from an application of the equations which govern the inviscid outer flow

(Euler's equation) giving $dp/dx=-\rho u_e\, du_e/dx$. Taking this into account, the here proposed Finite Volume method of the momentum equation is written in Eq.(13) which is a fully implicit numerical approximation of the mathematical modeling given by Eq.(1).

$$u_P^{k+1} - u_N^{k+1}\frac{\nu}{hu_P^k} - u_S^{k+1}\frac{v_P^k + \frac{\nu}{h}}{u_P^k} = u_W^k - 2\frac{\nu}{h} - v_P^k + u_{ep}\frac{u_{eP}^k - u_{eW}^k}{hu_P^k} \qquad (13)$$

where ν is the kinematic viscosity, *h* represents the vertical and horizontal edge size of a control volume. Including, (u_W,v_W), (u_N,v_N), and (u_S,v_S) are the west, north and south neighbors of a control volume (u_P,v_P), respectively.

8 - Experimental investigation

Tests were accomplished with air kinematic viscosity $v=1.5^{\cdot}10^{-5}\ m^2/s$. Regarding refining decision, it is done in both directions, namely, horizontal and vertical directions. In other words, a volume node is refined whether the sum of its flux and the flux of one of its neighbors, divided by the edge length of the control volume, is larger than a criteria determined by the user.

8.1 – Flat Plate boundary layer reconstruction

Tests of flat plate boundary layer reconstruction were realized with $u_\infty=200$, $u_\infty=10$ and also a limit of eight refinement levels for each control volume. Initially, Blasius solutions are depicted in Fig. 6. That is to say that lines in Fig. 6 show the boundary layers for both tests.

One example of tests performed for the here approach adaptive mesh refinement reconstruction is depicted in Fig. 7 comprising of $x=[0;1]$, $y=[0;0.1]$ and $u_\infty=200$. Moreover, black line shows the modified Hilbert curve for ordering mesh and red line shows the boundary layer.

A test comparing the adaptive mesh refinement and a non-adaptive mesh refinement was accomplished in order to show the adaptive mesh refinement improvement. The non-adaptive mesh refinement needed 340 control volumes whereas test with the adaptive mesh refinement, presented in Fig. 7, needed 164 control volumes, which makes up 42.2% of the non-adaptive mesh refinement scheme. Along the same lines, the adaptive mesh refinement scheme lasted 234 milliseconds whereas the non-adaptive mesh refinement scheme lasted 469 milliseconds. Considering the numerical average error between the reconstruction and Blasius solution, it is 0.000562 in this test.

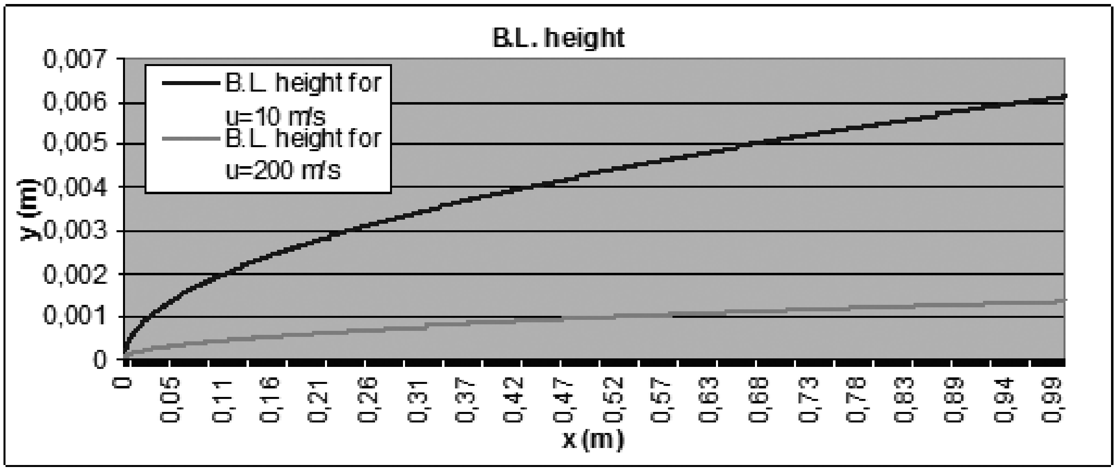


Figure 6: Blasius solution to kinematic viscosity ν=1.5·10^{-5}, u_∞ = *10 m/s* and u_∞ = *200 m/s*.

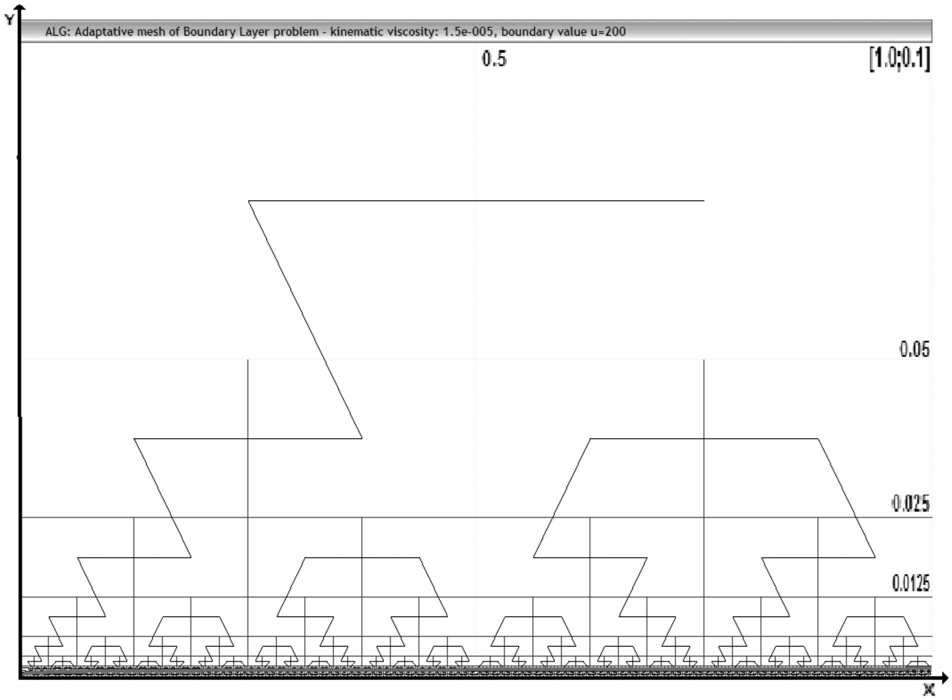

Y
ALG: Adaptative mesh of Boundary Layer problem - kinematic viscosity: 1.5e-005, boundary value u=200
0.5
[1.0;0.1]
0.05
0.025
0.0125
X

Figure 7: Flat plate boundary layer problem with kinematic viscosity *ν=1.5·10^{-5}*, *u_∞=200* and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;0.1]*).

Figure 8 depicts a test accomplished with *x=[0;1]*, *y=[0; 0.01]* and $u_\infty=10$ in order to show that results are similar to Blasius solution. In this test, it needed 340 control volumes for the non-adaptive mesh refinement scheme whereas test accomplished to adaptive mesh refinement scheme needed 169 control volumes, which is 49.7% of the total amount of the non-adaptive mesh refinement scheme. In addition, the adaptive mesh refinement scheme lasted 813 milliseconds whereas the non-adaptive mesh refinement scheme lasted 1282 milliseconds. In this test, numerical average error between reconstruction and Blasius solution is 0.008305. Shortly, table I summarizes those results.

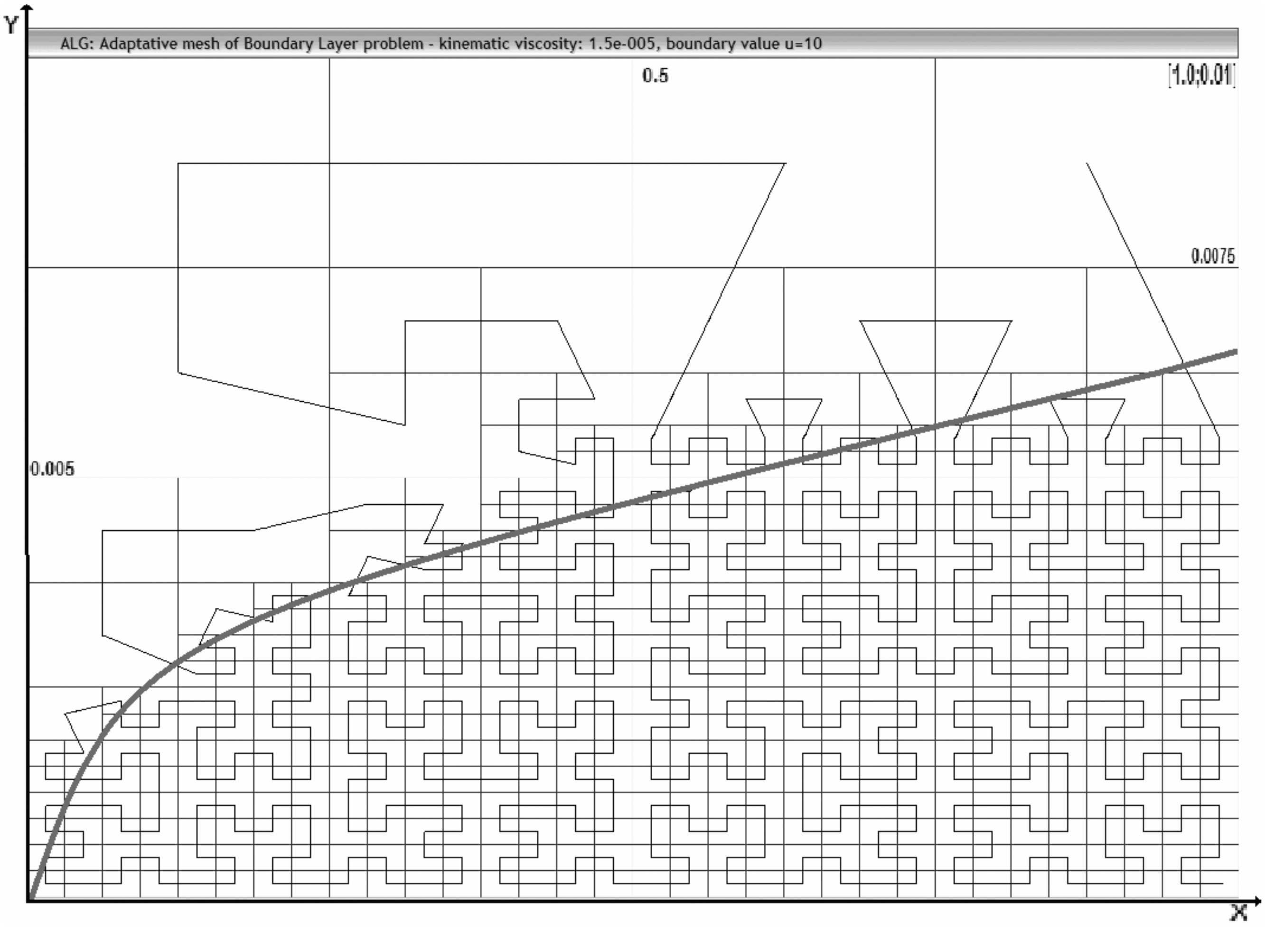


Figure 8: Flat plate boundary layer problem with kinematic viscosity $v=1.5^{\cdot}10^{-5}$, $u_\infty=10$ and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;0.01]*) showing horizontal component of the vector field.

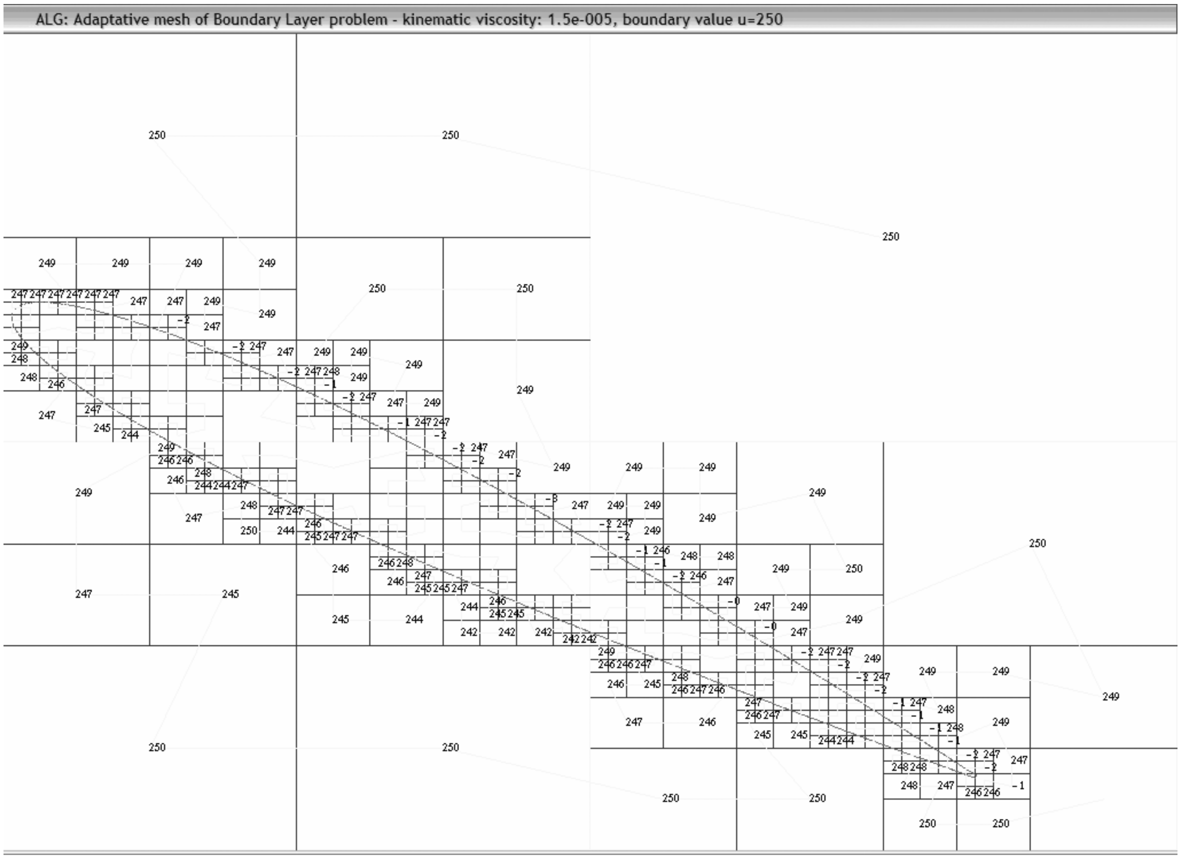
Table I: Comparison between adaptive and non-adaptive schemes.

Tests	$\boldsymbol{u_\infty}$**=200**		$\boldsymbol{u_\infty}$**=10**	
	Number of refinements	*Processing time in milliseconds*	*Number of refinements*	*Processing time in milliseconds*
Non-adaptive mesh Refinement	340	469	340	1282
Adaptive mesh Refinement	164	234	169	813
Numerical average error comparing reconstruction and Blasius solution	0.000562		0.001694	

8.2 – NACA0012 boundary layer reconstruction

In the figures of this section, a NACA0012 airfoil is presented in red with 35° of angle of attack. In relation to the modified Hilbert curve, it is represented by a yellow line. Besides that, such tests were accomplished with $u_\infty=250$. Thus, test of NACA0012 boundary layer numerical simulation shown in Fig. 9 with a limit of 6 refinement levels for every control volume around the airfoil. More precisely, Fig. 9 shows horizontal component of the vector field as integers. Figure 10 shows a part of Fig. 9, where boundary layer and turbulence can be observed.

Another test of NACA0012 boundary layer numerical simulation is shown in Fig. 11. It presents a limit of 5 refinement levels for every control volume around the airfoil. To be precise, Fig. 11 shows both horizontal and vertical components (u,v) of the vector field as integers. Fig. 12 shows a part of Fig. 11, where boundary layer and turbulence can be observed. Figure 13 sketches the flux shown in Fig. 9 representing its vector fields. Figure 14 shows details of Fig. 13 representing the turbulence inner the boundary layer. Figure 15 shows the horizontal component of the vector field of the numerical simulation. Similarly, the NACA0012 airfoil presents angle of attack of 35°. In addition to that, Autonomous Leaves Graph presents 10 refinement levels, kinematic viscosity of $v=1.5{\cdot}10^{-5}$ and $u_\infty=350$. This test resulted in 7474 graph nodes. Likewise, Fig. 16 shows the refinement levels of the numerical simulation of the NACA0012 airfoil with 35° of angle of attack with Autonomous Leaves Graph with 10 refinement levels.



ALG: Adaptative mesh of Boundary Layer problem - kinematic viscosity: 1.5e-005, boundary value u=250

Figure 9: Numerical simulation of the NACA0012 airfoil boundary layer with kinematic viscosity $v=1.5^{\cdot}10^{-5}$, $u_{\infty}=250$, 35° of angle of attack and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;1]*) showing the horizontal component of the vector field.

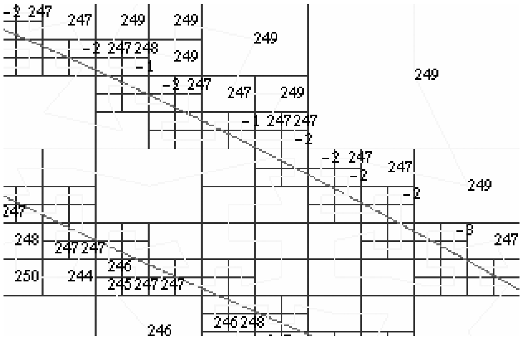
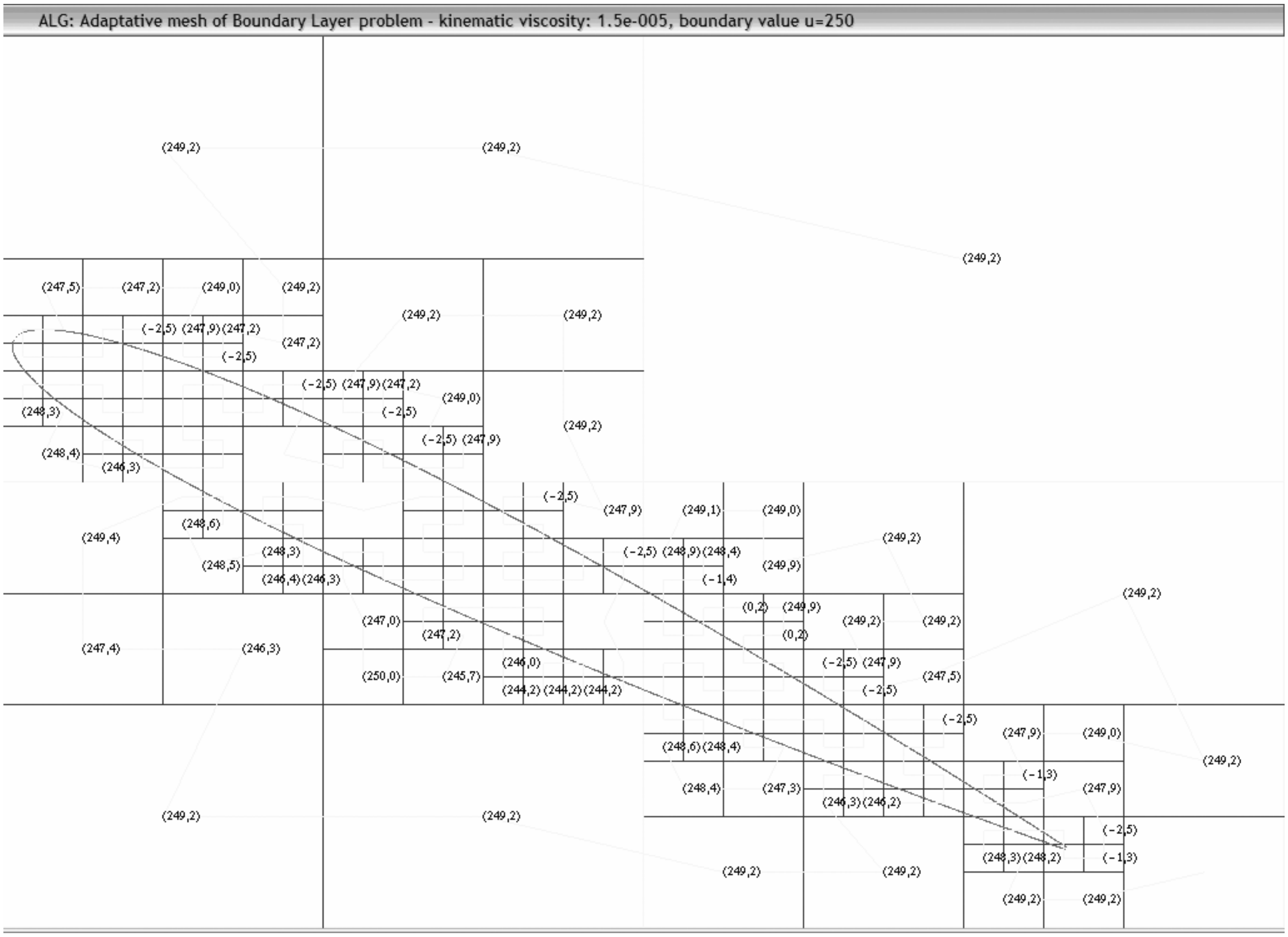


Figure 10: Details of the numerical simulation of the NACA0012 airfoil boundary layer with kinematic viscosity $v=1.5^{\cdot}10^{-5}$, $u_{\infty}=250$, 35° of angle of attack and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;1]*) showing the horizontal component of the vector field.



ALG: Adaptative mesh of Boundary Layer problem - kinematic viscosity: 1.5e-005, boundary value u=250

Figure 11: Numerical simulation of the NACA0012 airfoil boundary layer with kinematic viscosity *v=1.5·10^{-5}*, *u_∞=250*, 35^{o} of angle of attack and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;1]*) showing both horizontal and vertical components (*u,v*) of the vector field as integers.

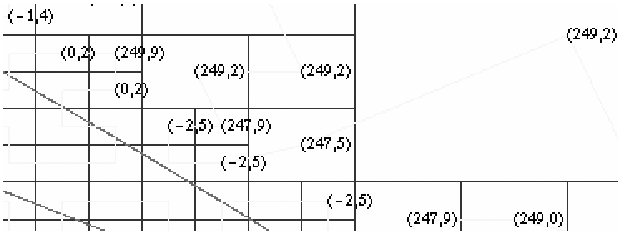


Figure 12: Details of the numerical simulation of the NACA0012 airfoil boundary layer with kinematic viscosity *v=1.5·10^{-5}*, *u_∞=250*, 35^{o} of angle of attack and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;1]*) showing both horizontal and vertical components (*u,v*) of the vector field.

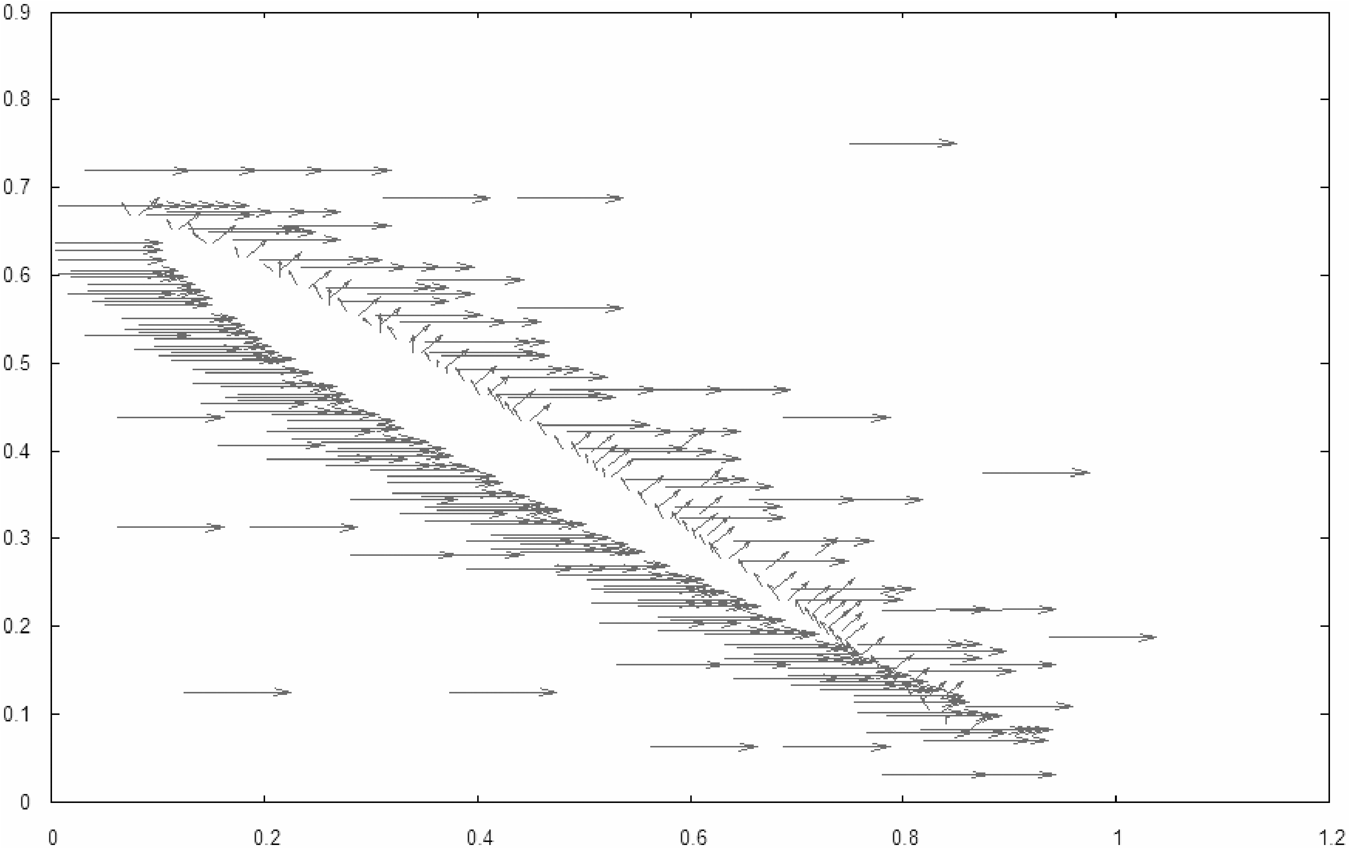


Figure 13: NACA0012 airfoil boundary layer flux representation with kinematic viscosity *ν=1.5·10*$^{-5}$, *u*$_{\infty}$*=250*, 35^{o} of angle of attack, 7 refinement levels and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;1]*) showing the directions of the vector field (*u,v*).

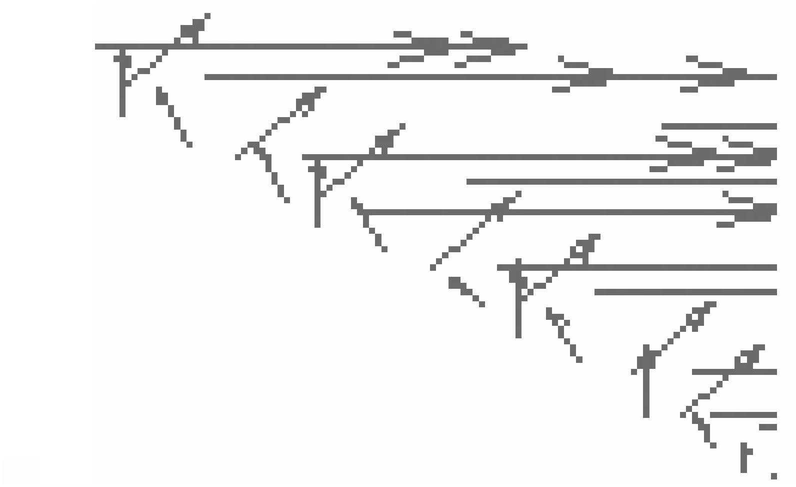


Figure 14: Details of NACA0012 airfoil boundary layer flux representation with kinematic viscosity *ν=1.5·10*$^{-5}$, *u*$_{\infty}$*=250*, 35^{o} of angle of attack, 7 refinement levels and Autonomous Leaves Graph adaptive mesh refinement (*x=[0;1]*, *y=[0;1]*) showing the directions of the vector field (*u,v*).

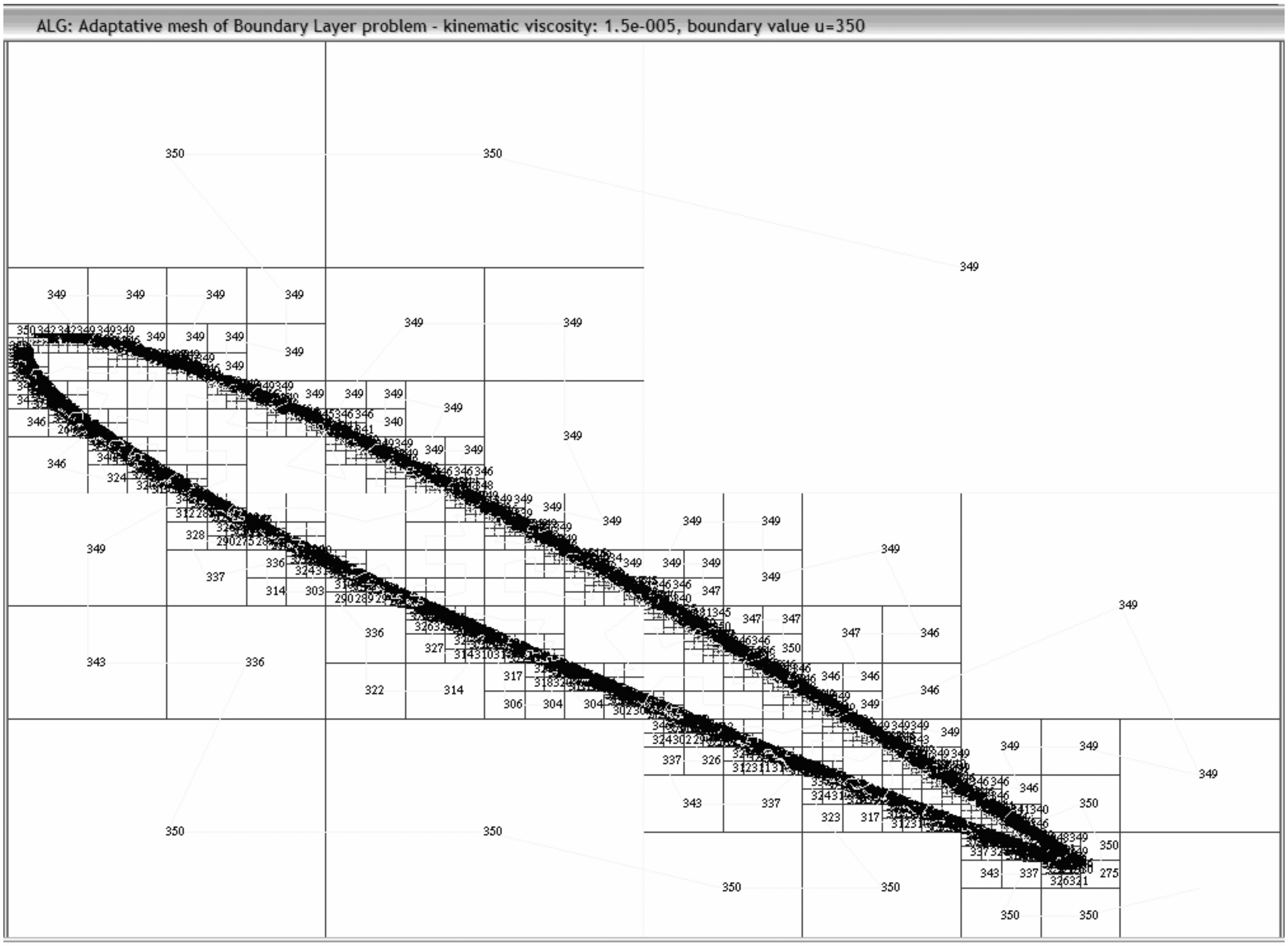

ALG: Adaptative mesh of Boundary Layer problem - kinematic viscosity: 1.5e-005, boundary value u=350

Figure 15: boundary layer numerical simulation of the NACA0012 airfoil with kinematic viscosity $v=1.5^{\cdot}10^{-5}$, $u_{\infty}=350$, 35° of angle of attack and Autonomous Leaves Graph adaptive mesh refinement ($x=[0;1]$, $y=[0;1]$) showing the horizontal component of the vector field in 10 refinement levels.

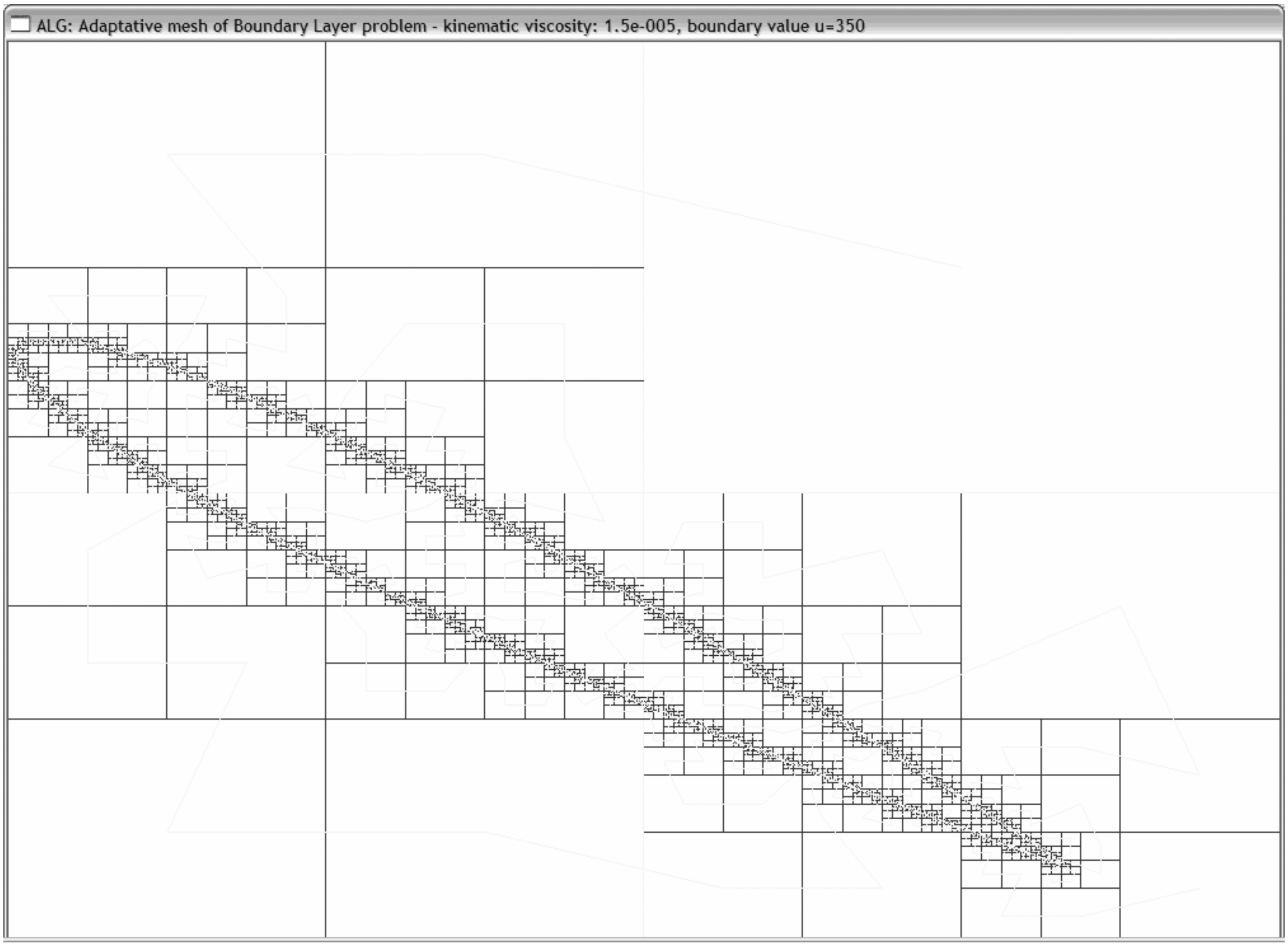

ALG: Adaptative mesh of Boundary Layer problem - kinematic viscosity: 1.5e-005, boundary value u=350

Figure 16: Representation of the modified Hilbert curve and adaptive refinement in the numerical simulation of the NACA0012 airfoil boundary layer with kinematic viscosity $v=1.5\cdot 10^{-5}$, $u_{\infty}=350$, 35^{o} of angle of attack and Autonomous Leaves Graph adaptive mesh refinement ($x=[0;1]$, $y=[0;1]$) in 10 refinement levels.

9 – Considerations and future work

This work presents boundary layer problem reconstruction by Autonomous Leaves Graph using Finite Volume method. Flat plate boundary layer reconstruction is compared to Blasius solution and NACA0012 airfoil is numerically reconstructed showing that Autonomous Leaves Graph for adaptive mesh refinement with the modified Hilbert curve for ordering volume nodes of the graph is adequate for a complex problem such as an aerodynamic problem. In the future, a FVM simplex unstructured mesh should be investigated.

Acknowledgments

Authors are grateful to CAPES and FAPERJ for financial support.

References

Anderson, D.A., Tannehill, J.C. and Pletcher, R.H., 1984, “Computational Fluid Mechanics and Heat Transfer”, Hemisphere, New York, 599 p.

Anderson, J.D., 1991, “Fundamentals of Aerodynamics”, McGraw-Hill Inc., New York, 921 p.

Burgarelli, D.D., Kischinhevski, M. and Biezuner, R.J., 2006, “A new adaptive mesh refinement strategy for numerically solving evolutionary PDE’s”, J. of Computational and Applied Math., Vol.196, pp. 115–131.

Madsen, J.I., 1998, “Design optimization of internal flow devices”, Ph.D. Thesis, Institute of Energy Technology and Institute of Mechanical Engineering of Alborg University, Denmark, 182 p.

Neel, R.E., 1997, “Advances in computational fluid dynamics: turbulent separated flows and transonic potential flows”m Ph.D. Thesis, Faculty of Virginia Polytechnic Institute and State University, 252 p.

Patankar, S.V., 1980, “Numerical Heat Transfer e Fluid Flow”, Taylor & Francis, New York, 197 p.

Schlichting, H., 1979, “Boundary-Layer Theory”, McGraw-Hill Inc., New York, 817 p.

Sperandio, D., Mendes, J.T., Silva, L.H.M., 2006, “Cálculo Numérico: Características Matemáticas e Computacionais dos Métodos Numéricos”, Pearson Prentice Hall, São Paulo, Brazil, 354 p.

Venkatachari, B.S., 2005, “Development and validation of a transient viscous flow solver based on a space-time CE/SE framework”, M.Sc. Dissertation, The University of Alabama at Birmingham, 120 p.