ARISTOTELIO UNIVERSITY OF THESSALONIKI Polytechnic School
Department of Mechanical Engineering
Laboratory of FLUID & TURBINE ENGINEERING

COMPUTATIONAL FLUID MECHANICS

Author : Diogenis Tsichlakis

Thessaloniki, May 2022

1. THE PROBLEM

This paper is the additional part (bonus) of the 1st computational topic in the course 'Computational Fluid Mechanics'. In the first part, we calculated the velocity field through the boundary layer, above a horizontal plate, with certain dimensions and using two different methods (Explicit and Entwined). In the 2nd part now we are asked to densify the mesh differently in the y-direction (vertical direction) in such a way that it thickens the mesh as we approach the wall.

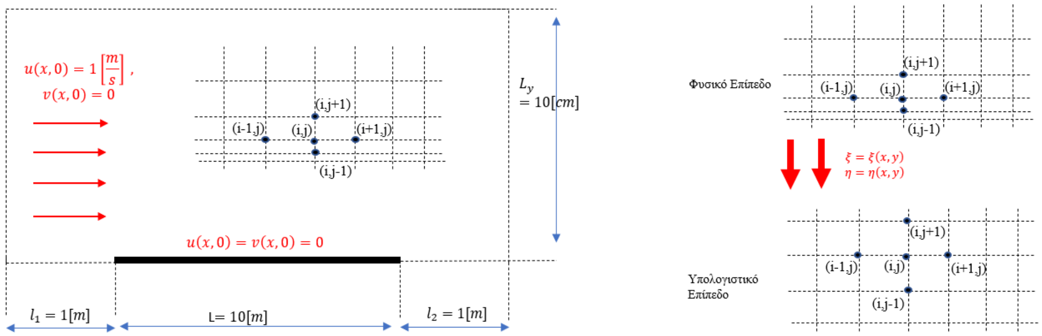
After this different geometric discretization is performed, we are asked to solve the boundary layer again, using one of the two methods we exploited to solve the first part. For this particular work we choose to solve the boundary layer with the explicit method and to compare at the end the results of the method with the initial uniform grid, with those of the differentiated grid.

2. PROBLEM DERIVIATION

First we rewrite our system of equations (I), where for the variables of this system we know that the velocity components are functions of the 2 spatial coordinates, and also the kinematic viscosity of the air.$u = u(x, y), v = v(x, y) \nu = 1.46\ 10^{-5}\ \left[\frac{m^2}{s}\right]$

$$\begin{cases} \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} = 0 \\ u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} = \nu\dfrac{\partial^2 u}{\partial y^2} \end{cases} \quad (\boldsymbol{I})$$

In Figure 1, the form of our model is presented. In this figure we see the boundary conditions at the borders and the shape of the mesh (dotted) on the left and the meshes under study on the right.



Shape1. Problem model and different levels of work

As we can see on the right of the figure above, the 2 different meshes that we will use are presented. The upper grid is the one we choose to express in the nodes the variables of the problem and the lower grid is the one we choose to numerically solve our problem. The densification that we can see in the upper grid, is carried out so that we get better information near the plate where the gradients of the velocities are greater. To be able to calculate the variables under study, in the boundary layer, we need to transform our condensed mesh into a uniformly ordered one. These transformations have the form

$$\xi = \xi(x,y) \;\; \kappa\alpha\iota \; \eta = \eta(x,y) \qquad (\boldsymbol{II})$$

Which transform the expressions of our equations, from the (x,y) plane to the (ξ,η). First we will present the way in which the equations of system (I) are transformed from one level to another and then we will define the analytical relations of the equations of system (II).

Based on the expressions of the equations in system (II) and using chain derivation, the general form of the partial derivatives appearing in system (I) is obtained, in terms of the coordinates of the (ξ, η) plane as follows:

$$\frac{\partial}{\partial x} = \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial \xi}{\partial x}\right) + \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial \eta}{\partial x}\right)$$

$$\frac{\partial}{\partial y} = \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial \xi}{\partial y}\right) + \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial \eta}{\partial y}\right)$$

$$\frac{\partial^2}{\partial y^2} = \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial^2 \xi}{\partial y^2}\right) + \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial^2 \eta}{\partial y^2}\right) + \left(\frac{\partial^2}{\partial \xi^2}\right)\left(\frac{\partial \xi}{\partial y}\right)^2 + \left(\frac{\partial^2}{\partial \eta^2}\right)\left(\frac{\partial \eta}{\partial y}\right)^2 + 2\left(\frac{\partial^2}{\partial \eta \partial \xi}\right)\left(\frac{\partial \eta}{\partial y}\right)\left(\frac{\partial \xi}{\partial y}\right)$$

In the above three equations, the coefficients of the derivatives (terms in red) constitute the metric terms which can be calculated directly if we know the expressions of the equations of system (II).

Since we are densifying our mesh only for the y direction and keeping the step in the horizontal direction constant, the transformation in this direction will be expressed by the relation . The analytical transformation for the vertical direction, we choose to have the form , so we get the system of transformations:$\xi = x\eta = \sqrt{\frac{y}{4}}$

$$\begin{cases} \xi = x \\ \eta = \frac{1}{2}\sqrt{y} \end{cases} \quad (\mathbf{3\alpha})$$

And the inverse transformation:

$$\begin{cases} x = \xi \\ y = 4\eta^2 \end{cases} \quad (\mathbf{3\beta})$$

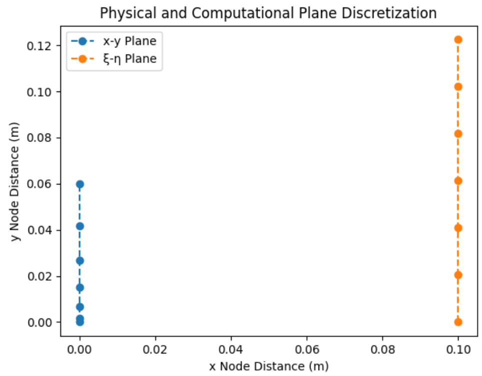
Deriving the 2nd equation of system (3b) in terms of η, we have:

$$\frac{dy}{d\eta} = 8\eta \ \Rightarrow dy = 8\eta \, d\eta$$

Considering the differentials as finite quantities, we finally get the relationship between the step in the vertical direction in the 2 different planes, from this equation we see that as we move away from the wall and n increases, y increases as well:

$$\Delta y = 8\eta\Delta\eta \ \ (\mathbf{4})$$

With this choice for the transformation of the nodes, the form of the 2 levels of discretization (Physical and Computational) will be the one presented in Figure 2. In this figure we observe the dimensions of our physical mesh (xy) and its densification as we approach the plate. On the contrary, the mesh resulting from the above transformation has larger dimensions, but keeps the distances between the nodes constant.



Shape2. Discretization of Physical and Computational Grid

Let us now calculate the metric terms in the partial differential equations presented above:

$$\frac{\partial \xi}{\partial x} = 1 \,, \qquad \frac{\partial \xi}{\partial y} = 0 \,, \ \ \frac{\partial \eta}{\partial x} = 0 \,,$$

$$\frac{\partial \eta}{\partial y} \ = \frac{1}{4\sqrt{y}} \ \alpha\pi ó \ (3\beta) \ \ y = 4\eta^2 \Rightarrow \rightarrow \ Ἀρα \ \frac{\partial \eta}{\partial y} \ = \frac{1}{8\eta}$$

$$\frac{\partial^2 \xi}{\partial y^2} = 0 \,,$$

$$\frac{\partial^2\eta}{\partial y^2} = -\frac{1}{8}y^{-\frac{3}{2}} \overset{(3\beta)}{\Longrightarrow} \frac{\partial^2\eta}{\partial y^2} = -\frac{1}{64\eta^3}$$

Based on these relations the partial derivatives become:

$$\begin{cases} \dfrac{\partial}{\partial x} = \dfrac{\partial}{\partial \xi} \\ \dfrac{\partial}{\partial y} = \left(\dfrac{\partial}{\partial \eta}\right)\dfrac{1}{8\eta} \\ \dfrac{\partial^2}{\partial y^2} = -\left(\dfrac{\partial}{\partial \eta}\right)\dfrac{1}{64\eta^3} + \left(\dfrac{\partial^2}{\partial \eta^2}\right)\dfrac{1}{64\eta^2} \end{cases} \qquad (\boldsymbol{IV})$$

Based on the relations of system (IV) we transform the equations of system (I) and they come in the form:

$$\begin{cases} \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} = 0 \\ u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} = \nu\dfrac{\partial^2 u}{\partial y^2} \end{cases}$$

$$\overset{(IV)}{\Longrightarrow} \begin{cases} \dfrac{\partial u}{\partial \xi} + \dfrac{1}{8\eta}\dfrac{\partial v}{\partial \eta} = 0 \\ u(\xi,\eta)\dfrac{\partial u}{\partial \xi} + v(\xi,\eta)\dfrac{1}{8\eta}\dfrac{\partial u}{\partial \eta} = \nu\left(-\left(\dfrac{\partial u}{\partial \eta}\right)\left(\dfrac{1}{64\eta^3}\right) + \left(\dfrac{\partial^2 u}{\partial \eta^2}\right)\left(\dfrac{1}{64\eta^2}\right)\right) \end{cases}$$

$$\Rightarrow \begin{cases} \dfrac{\partial u}{\partial \xi} + \left(\dfrac{1}{8\eta}\right)\dfrac{\partial v}{\partial \eta} = 0 \\ u(\xi,\eta)\dfrac{\partial u}{\partial \xi} + v(\xi,\eta)\left(\dfrac{1}{8\eta}\right)\dfrac{\partial u}{\partial \eta} = -\nu\left(\dfrac{1}{64\eta^3}\right)\left(\dfrac{\partial u}{\partial \eta}\right) + \nu\left(\dfrac{1}{64\eta^2}\right)\left(\dfrac{\partial^2 u}{\partial \eta^2}\right) \end{cases} \qquad (\boldsymbol{V})$$

The system (V) constitutes the system of equations that we will discretize and solve to calculate the boundary layer velocities at the computational level. Finally, based on the transformations, we will transfer these results to the physical plane (x,y).

We differentiate the partial differential derivatives with the same finite differences as in the 1st part of the paper (Central Differences):

- $\left(\frac{\partial u}{\partial \xi}\right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta\xi} + O(\Delta\xi)^2$(CENTRAL DIFFERENCE) for the 1st equation
- $\left(\frac{\partial u}{\partial \xi}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta\xi} + O(\Delta\xi)^2$(FORWARD) for the 2nd equation
- $\left(\frac{\partial v}{\partial \eta}\right)_{i,j} = \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta\eta} + O(\Delta\eta)^2$ (CENTRAL DIFFERENCE)

- $\left(\frac{\partial u}{\partial \eta}\right)_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta\eta} + O(\Delta\eta)^2$(CENTRAL DIFFERENCE)
- $\left(\frac{\partial^2 u}{\partial \eta^2}\right)_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta\eta)^2} + O(\Delta\eta)^2$(FINITE DIFFERENCE OF THE 2ND ORDER)

We substitute these finite differences into the system of equations (V) after first expressing it at the node (i,j). We see that the 1st derivative of u with respect to x is expressed by central difference for the 1st equation of the system (greater accuracy) and by forward difference for the 2nd equation of the system (reduction of computational load).

$$\left(\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta\xi}\right) + \left(\frac{1}{8\eta}\right)\left(\frac{v_{i,j+1} - v_{i,j-1}}{2\Delta\eta}\right) = 0$$

$$\begin{aligned} u(\xi,\eta)\left(\frac{u_{i+1,j} - u_{i,j}}{\Delta\xi}\right) + v(\xi,\eta)\left(\frac{1}{8\eta}\right)\left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta\eta}\right) \\ = -\nu\left(\frac{1}{64\eta^3}\right)\left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta\eta}\right) + \nu\left(\frac{1}{64\eta^2}\right)\left(\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta\eta)^2}\right) \end{aligned}$$

$$v_{i,j+1} = v_{i,j-1} - 8\eta\left(\frac{\Delta\eta}{\Delta\xi}\right)\left(u_{i+1,j} - u_{i-1,j}\right) \quad \textbf{(4)}$$

and

$$\begin{aligned} u_{i+1,j} = u_{i,j} &- \frac{v(\xi,\eta)}{u(\xi,\eta)}\left(\frac{1}{8\eta}\right)\left(\frac{\Delta\xi}{2\Delta\eta}\right)\left(u_{i,j+1} - u_{i,j-1}\right) \\ &- \frac{\nu}{u(\xi,\eta)}\left(\frac{1}{64\eta^3}\right)\left(\frac{\Delta\xi}{2\Delta\eta}\right)\left(u_{i,j+1} - u_{i,j-1}\right) \\ &+ \frac{\nu}{u(\xi,\eta)}\left(\frac{1}{64\eta^2}\right)\left(\frac{\Delta\xi}{(\Delta\eta)^2}\right)\left(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}\right) \quad \textbf{(5)} \end{aligned}$$

The velocity term in the (n) direction by which we will divide is non-zero since the above equations will be applied from the 2nd row of nodes onwards (above the plate). In the numerical procedure that we will follow, we will need to use a finite difference for the lower boundary, to calculate the velocity v in the second row of nodes. Approaching this partial derivative with a 2nd degree polynomial, after operations we have:$u(\xi,\eta)$

$$\left(\frac{\partial v}{\partial \eta}\right)_{i,0} = \frac{-3v_{i,0} + 4v_{i,1} - v_{i,2}}{2\Delta\eta} + O(\Delta\eta)^2$$

To calculate the term we replace the differences only for the 1st equation so we have:$v_{i,1}$

$$\left(\frac{\partial u}{\partial x}\right)_{i,0} + \left(\frac{\partial v}{\partial y}\right)_{i,0} = 0 \Rightarrow \frac{u_{i+1,0} - u_{i-1,0}}{2\Delta\xi} + \left(\frac{1}{8\eta}\right)\left(\frac{-3v_{i,0} + 4v_{i,1} - v_{i,2}}{2\Delta\eta}\right) = 0$$

Because since we will start the calculations from the 2nd column and above but also because of the SS we have: $\left(\frac{1}{15\eta^2}\right) \neq 0 u_{i+1,0} = u_{i-1,0} = 0 \; \kappa\alpha\iota \; v_{i,0} = 0$

We get :

$$\Rightarrow v_{i,1} = \frac{1}{4} v_{i,2} \quad \mathbf{(6)}$$

By rearranging the terms of equation (4) and (5) we have:

$$v_{i,j+1} = v_{i,j-1} - 8\eta \left(\frac{\Delta\eta}{\Delta\xi}\right)\left(u_{i+1,j} - u_{i-1,j}\right) \; \mathbf{(7)}$$

$$\begin{aligned} u_{i+1,j} = u_{i,j} &- \frac{v(\xi,\eta)}{u(\xi,\eta)} \left(\frac{1}{8\eta}\right)\left(\frac{\Delta\xi}{2\Delta\eta}\right)\left(u_{i,j+1} - u_{i,j-1}\right) \\ &- \frac{\nu}{u(\xi,\eta)} \left(\frac{1}{64\eta^3}\right)\left(\frac{\Delta\xi}{2\Delta\eta}\right)\left(u_{i,j+1} - u_{i,j-1}\right) \\ &+ \frac{\nu}{u(\xi,\eta)} \left(\frac{1}{64\eta^2}\right)\left(\frac{\Delta\xi}{(\Delta\eta)^2}\right)\left(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}\right) \end{aligned}$$

In a more accessible form this equation is written:

$$\begin{aligned} u_{i+1,j} = u_{i,j} - A_{i,j}\left(u_{i,j+1} - u_{i,j-1}\right) - B_{i,j}\left(u_{i,j+1} - u_{i,j-1}\right) \\ + C_{i,j}\left(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}\right) \; \mathbf{(8)} \end{aligned}$$

Where:

$$A_{i,j} = \frac{v(\xi,\eta)}{u(\xi,\eta)} \left(\frac{1}{8\eta}\right)\left(\frac{\Delta\xi}{2\Delta\eta}\right)$$

$$B_{i,j} = \frac{\nu}{u(\xi,\eta)} \left(\frac{1}{64\eta^3}\right)\left(\frac{\Delta\xi}{2\Delta\eta}\right)$$

$$C_{i,j} = \frac{\nu}{u(\xi,\eta)} \left(\frac{1}{64\eta^2}\right)\left(\frac{\Delta\xi}{(\Delta\eta)^2}\right)$$

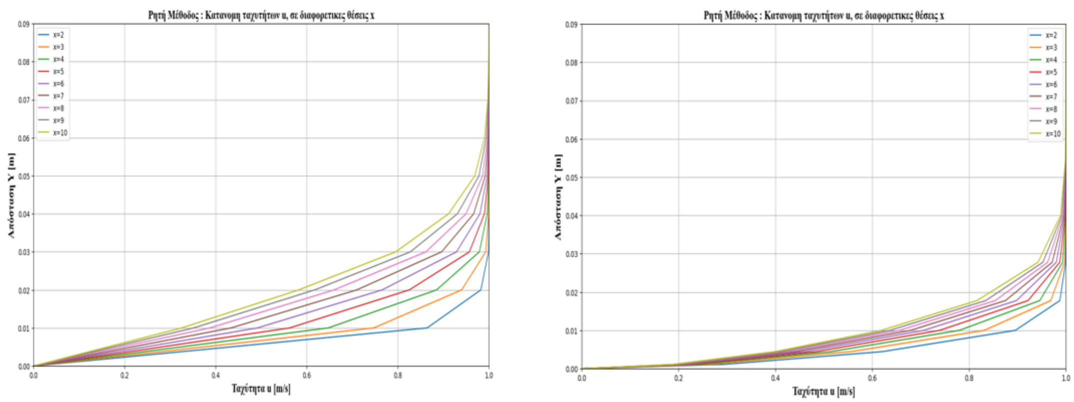
Using equations (6), (7) and (8) we calculate the boundary layer in the (ξ,η) plane. Then, using the transformations (3b), we calculate the coordinates of the nodes (x,y) to which correspond the velocities we calculated in the (ξ,η) plane.

Based on the above equations and applying the algorithm we described in the first part of the paper, we construct the diagrams presented below.

3. SOLUTION - RESULTS

In order to extract the results obtained from the differentiated mesh, using the explicit method, the file 'CFD_main3.py' was written in Python language and is delivered together with the work.

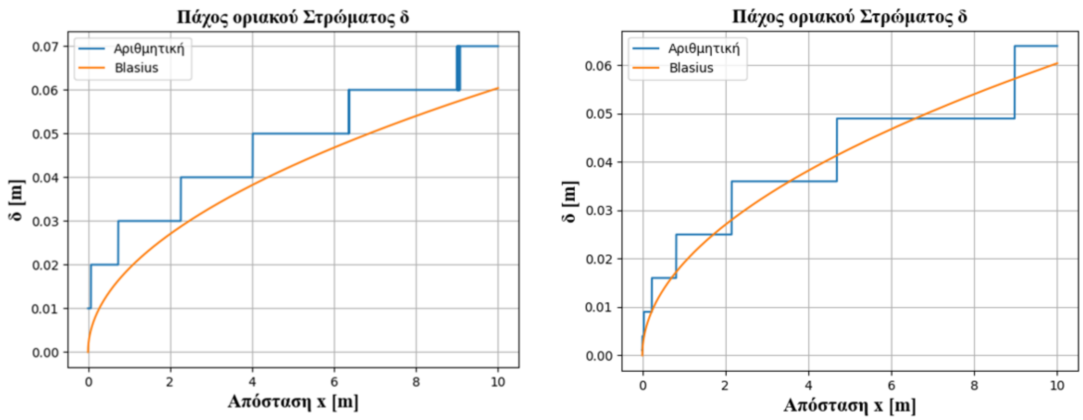
The following figures show the extracted diagrams. On the left of each figure is the solution from the first part of the work with the uniform grid, while on the right are the solutions with the differentiated grid. Note that the 2 meshes consist of the same number of nodes, the same pitch Δx and a different pitch Δy according to the analysis made above. The number of nodes in the y direction is 11 while correspondingly in the x direction it is 12000.



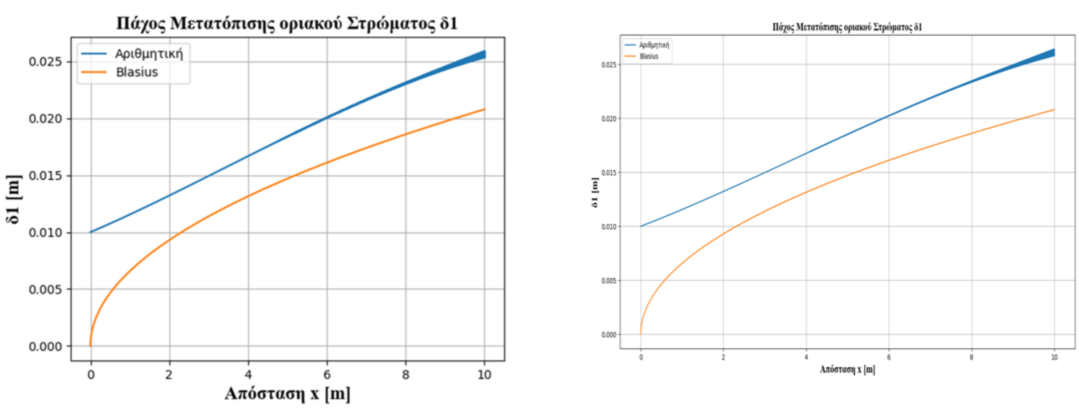
Shape3. Comparison of velocity profiles for the 2 different grids

From Figure 3, we see that the velocity profiles for the same x positions show an equally parabolic shape. The form of the curves is not identical due to the different values of the vertical axis, nevertheless we can say that both grids give satisfactory results.

In Figure 4, the boundary layer thickness plots of the 2 grids are illustrated. We see that in both figures, we are very close to the Blasius solution, although it is obvious that with the non-uniform mesh we approximate it better, which is due to the concentrated nodes at the wall and the better information we get about the steep velocity gradients near the wall this. Apart from differentiating the equations to calculate the velocities, the coordinates for the nodes, and how to define Δy so that we can calculate the thickness, no other changes were made to the code delivered in the first part of the paper for the explicit method(' CFD_main1.py').δ_1

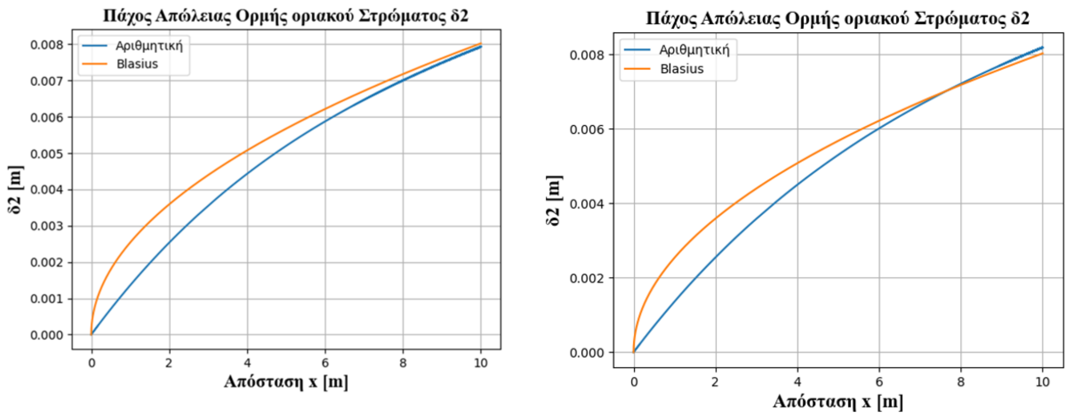


Shape4.Boundary layer thickness comparison for the 2 different grids

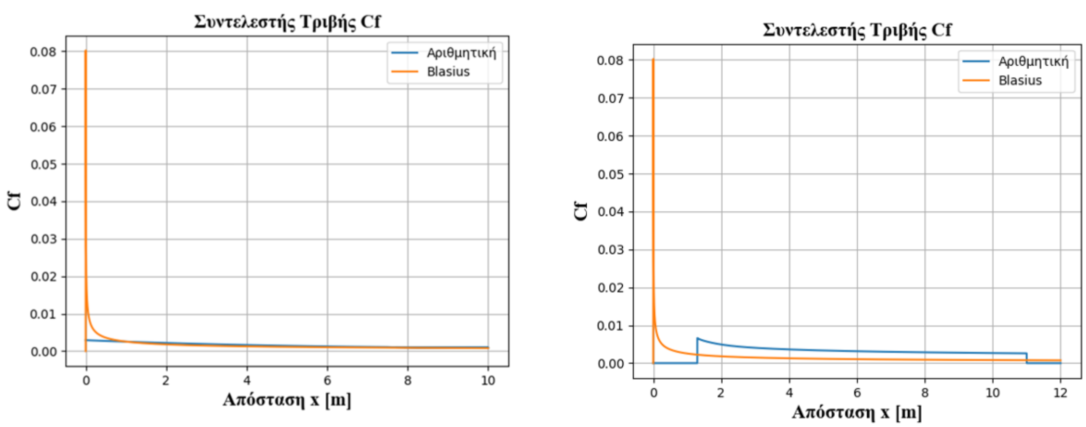


Shape5.Boundary layer displacement thickness comparison for the 2 different grids

From Figure 5, we can say that the boundary layer displacement thicknesses for the 2 grids are marginally identical. Likewise for the momentum loss thicknesses, with small differences near the end of the plate. Finally observing Figure 6. We see that both grids give similar accuracy for the friction coefficient.



Shape6.Boundary layer momentum loss thickness comparison for the 2 different grids



Shape7.Comparison of coefficient of friction for the 2 different grids

In general we conclude, mainly from the diagrams for the boundary layer thickness (Figure 4) and the thicknesses and (Figures 5 and 6), that by densifying the mesh we achieve better accuracy. Of course, as the distance x increases, the computational errors increase and the accuracy decreases compared to the uniformly distributed grid, which exhibits greater stability in terms of computational error (less complicated expressions for calculating velocities).$\delta_1 \delta_2$

The conclusions drawn above are also confirmed by Table 1, in which the values for the execution time of each code and the error of the calculated thickness for the boundary layer compared to the Blasius solution are found. We notice that for a non-uniformly distributed grid, we have an increase in the total execution time of the calculations, of course with this grid we get a seven times smaller error, compared to our original grid. So the behavior we pointed out above is also verified by the error value.

Panel1. Comparison of Time and Accuracy for the 2 different grids

COMPARISON OF GRIDS		
	ORIGINAL	**DIFFERENTIATED**
TIME (sec)	11,523	14,732
DEVIATION FROM BLASIUS	0.828%	0.13%

Finally, let us emphasize, as we also pointed out in the 1st part of the work, that, due to the finite differences chosen for the discretization of the equations, the computing power we have and the way the code is written in Python, the steps we used are particularly large. Also a test was made to increase the nodes or more complex densification of the mesh near the wall, it failed due to the appearance of large and unstable errors, which large and unstable errors also appear in the case of trying to decrease the pitch in the x direction.