NUMERICAL SIMULATION OF THE INVISCID FLOW PAST AIRFOILS

Khalid M. Sultan and Basim A. Belgasim

Mechanical Engineering Department, Faculty of Engineering,
Garyounis University, Benghazi, Libya
Dr-khalid-sultan@hotmail.de

الملخص

تعتبر محاكاة التدفق على الأجسام الانسيابيه من العمليات المهمة للعديد من المهندسين والباحثين في مجال ديناميكا الموائع. من بين الأمثلة العملية على ذلك حساب التدفق على جناح الطائرة أو حساب التدفق على شفرات التوربينات وضواغط الهواء حيث عادة ماتتم هذه الحسابات باستخدام برامج تجارية جاهزة، إلا أن هذه البرامج تباع دائما كنسخة قابلة للتشغيل ولا توجد إمكانية الإطلاع على الطرق الرياضية التي يرتكز عليها البرنامج أو تعديلها. تمت عملية بناء مولد شبكات حسابية كارتيزيه في هذه الدراسة، كما تمت عملية بناء محلل تدفق عديم الاحتكاك على الأجسام الانسيابية.مولد الشبكات الذي تم بناءه في هذه الدراسة يستند على طريقة الشعاع المرسل وذلك لتصنيف الخلايا إلى خلايا تدفق، خلايا تقع داخل الجسم، وخلايا قص. أما نقاط القص في خلايا القص فيتم تحديدها بعد أن وظفت منحنيات Bezier لتمثل شكل الجسم. معادلات الحركة تم تحويلها إلى معادلات جبرية قابلة للحل باستخدام الحاسوب عن طريق استخدام طريقة AUSM لحدود الحمل واستخدام طريقة Runge-Kutta للحدود غير المنتظمة. بعد اختبار مولد الشبكات أتضح أنه قادر على تمثيل الأشكال الإنسانيه التي اختبرت على الرغمٍ من بسٍاطة الطٍريقة. محلل التدفق الذي أنشئ وطوِّر في هذه الدراسة أظهر أداءً جيدا مقارنة مع النتائج المنشورة.العمل يجري الآن على إدخال حساسات لكي يتمكن محلل التدفق من تكييف الشبكة أثناء إجراء الحسابات.

ABSTRACT

For many engineers and researchers the simulation of the flow field around aerodynamic shapes such as airfoils or turbine and compressor vanes is of particular interest. In these situations the computations are almost carried out using commercial softwares that are available in the market. Nevertheless, these softwares are always provided without the source code, and hence they leave no chance for the user to modify and develop their existing algorithm. In this study, an unstructured Cartesian grid generator and flow solver was built from scratch in order to simulate inviscid flows past airfoils. In the grid generator, the ray-casting method is employed to classify the cells in the computational field to flow, body, or cut cells. In order to find the coordinates of the cut cells the surface of the given shape is represented by a set of consecutive Bezier curves. The governing equations are discretized using the AUSM scheme for the convective terms and a five-stage Runge-Kutta method for the unsteady terms. The grid generator based on the ray-casting method and Bezier curves showed good performance in representing the given geometries despite the simplicity of the method. The flow solver that is based on the AUSM scheme and Runge-Kutta method showed very good agreement when compared with the available literature. Work is now in progress to incorporate some adaptation sensors to enable the solver to adapt the grid during the solution process.

KEYWORDS: Unstructured Cartesian grid; Euler equations; AUSM scheme

INTODUCTION

The growing application field of fluid mechanics in practical engineering projects and products has demanded the modelling and solution of several flow problems. As a result, many engineers and researchers are using commercial softwares (FLUENT, CFX, and NUMECA are examples of flow solving softwares.) to solve their particular problems. Nevertheless, these commercial, professional, and user–friendly softwares are always provided without the source code and hence leaving no chance for the user, either an engineer or a researcher, to modify and / or develop their built in methods and techniques. By reviewing the different approaches that are commonly used to simulate both internal and external flows, one finds that there are two main approaches in this field namely; the flow simulations that are based on body-fitted grids and those that are based on Cartesian grids. Body-fitted grids, either structured or unstructured, have gained there popularity through the ease of implementing the boundary conditions while this is the main challenge when using the Cartesian grids. On the other side, in the case of body-fitted grids the grid generation process is time and computational resources consuming process. Moreover, the grid generation process with this approach does not lend itself to automation and has both computing and memory overheads due to the metric terms. The Cartesian grid method does not include any transformation of the physical domain to a computational domain and hence the metric terms are eliminated. As a result more efficient use of the available memory and shorter computing time are expected. Having mentioned this and keeping other guidelines in mind, such as the simplicity of the methods used and programming, an unstructured Cartesian grid generator and flow solver has been developed and tested on a number of test cases. In this work, an unstructured Cartesian grid generator and flow solver was built and developed to solve the flow past aerodynamic geometries. Although the grid generator is capable of generating Cartesian grids for even more complicated geometries, such as a complete airplane configuration, the flow solver was only tested on the NACA0012 airfoil for a number of published test cases, see [9]. The unstructured Cartesian grid generator developed in this work is based on the ray –casting method which is a very simple method and does not include any floating point operation. The flow solver employs the AUSM scheme for the discretization of the convective terms and a five stage Runge-Kutta method for advancing the solution in time.

CARTESIAN GRID GENERATION

The grid generation process can be subdivided into three main tasks: Generation of a base grid, cells classification process, and determination of the coordinates of the intersection points where the surface of the body cuts the cells. The generation of a base grid is started when the user identifies the required lengths of the computational domain in both the x and y-directions (L_x and L_y) as well as the required number of cells also in both the x and y-directions (n_x and n_y). By dividing the length of the domain by the number of cells in the x-direction one gets the cell width Δx , the height of the cell Δy is obtained similarly. An array of cells is then created and the cells are stored in this array with each cell denoted by its unique identity number that is represented by the array index. As soon as the cell is inserted in the array, its neighbor's identity numbers in the north, south, east, and west directions are stored. The cells that were already created to form the base grid were all similar, that is; they are not classified according to their location with respect to the body. This process is very important since the cells that lie inside the body (body cells) will not be important. On the other hand the cells that lie outside the body (flow cell) and those lie partially inside and partially outside the body (cut cells) are very important. The flow cells will be integrated in time to obtain a solution of the conservation equations, and the cut cells will impose the spatial boundary conditions required by those equations. In order to classify the cells the ray - casting method is employed, see [1]. This method is explained as follows: A ray is started at each corner point of the four corners of the cell to be examined and allowed to spread out (horizontally or vertically) until it reaches the edge of the computational domain. This ray is traced and the number of times it intersects the body boundary is counted. The corner point is considered inside the body if the number of intersection times is odd. On the other hand, the corner point is considered outside the body if the number of intersection times is even. Once the four corners of a cell are examined the cell can be classified and an index is assigned a particular value, for instance zero to denote flow cells, 1 to denote cut cells, and 2 to denote body cells. To specify wether the ray intersects the body boundary or not, the body boundary must be represented by an equation rather than separate points. For this task the Bezier curves were employed.

MATHEMATICAL MODEL

For an inviscid, unsteady compressible, and two dimensional flows with both mass diffusion and thermal conductivity are neglected, the conservation laws are:

Continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho V)}{\partial y} = 0 \tag{1}$$

Momentum equation:

x-direction

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} + \frac{\partial(\rho u V)}{\partial y} = 0 \tag{2}$$

y-direction

$$\frac{\partial(\rho V)}{\partial t} + \frac{\partial(\rho u V)}{\partial x} + \frac{\partial(\rho V^2 + p)}{\partial y} = 0 \tag{3}$$

Energy equation:

$$\frac{\partial(\rho e)}{\partial t}+\frac{\partial\left[u(\rho e+p)\right]}{\partial x}+\frac{\partial\left[V(\rho e+p)\right]}{\partial y}=0 \qquad (4)$$

Where ρ is the fluid density, u is the fluid velocity in x-direction, V is the fluid velocity in y-direction, p is the pressure and e is the total energy. One more equation is needed in order to have a number of equations that is equal to the number of unknowns ($\rho, u, V, p, T, and\ e$). Using the thermodynamic relation $c_v = R/(\gamma - 1)$ and the definition of the specific internal energy $\hat{u} = C_v T$, the equation of state can be reformulated to relate the pressure p to the conservative variable ρe as:

$$p = (\gamma - 1)(\rho e - \frac{1}{2}\rho\vec{V}^2) \qquad (5)$$

Equation (1) to (5) can be rewritten in a dimensionless compact form as:

$$\frac{\partial\vec{Q}}{\partial t^*}+\frac{\partial\vec{E}}{\partial x^*}+\frac{\partial\vec{F}}{\partial y^*}=0 \qquad (6)$$

where:

$$\vec{Q}=\begin{bmatrix}\rho^*\\ \rho^*u^*\\ \rho^*V^*\\ \rho^*e^*\end{bmatrix},\quad \vec{E}=\begin{bmatrix}\rho^*u^*\\ \rho^*u^{*2}+p^*\\ \rho^*u^*V^*\\ u^*\left(\rho^*e^*+p^*\right)\end{bmatrix}\text{,and}\quad \vec{F}=\begin{bmatrix}\rho^*V^*\\ \rho^*V^*u^*\\ \rho^*V^{*2}+p^*\\ V^*\left(\rho^*e^*+p^*\right)\end{bmatrix}$$

The dimensionless variables are defined as shown in the nomenclature. It is worth noting here that the governing equations in the dimensionless form are similar to those that are in dimensional form, so for reason of simplicity the asterisk will be dropped from now and as this text goes on.

Initial and Boundary Conditions:

The governing equation, equation (6), is a partial differential equation with both time and space derivatives and thus requires both initial and boundary conditions. Specifying the initial conditions is a trivial task since the flow can be assumed to be parallel flow at the start of calculation. This is a conventional approach which is used for the solution of inviscid flows, for instance see [2-3]. The boundary conditions, on the other side, are classified into two types: Far-field boundary conditions and fluid-body boundary conditions. The far-field boundary conditions are based on the method of characteristics, see [4], and summarized as follows:

$$\left.\begin{cases} \vec{Q}_{inflow} = \left(\rho_\infty ,\ \rho_\infty \vec{V}_\infty ,\ \dfrac{p_{cd}}{\gamma - 1} + \dfrac{1}{2} \rho_\infty \vec{V}_\infty^2 \right)^T \\ \vec{Q}_{outflow} = \left(\rho_{cd} ,\ \rho_{cd} \vec{V}_{cd} ,\ \dfrac{p_\infty}{\gamma - 1} + \dfrac{1}{2} \rho_{cd} \vec{V}_{cd}^2 \right)^T \end{cases}\right\} \text{subsonic flow} \tag{7}$$

$$\left.\begin{cases} \vec{Q}_{inflow} = \left(\rho_\infty ,\ \rho_\infty \vec{V}_\infty ,\ \dfrac{p_\infty}{\gamma - 1} + \dfrac{1}{2} \rho_\infty \vec{V}_\infty^2 \right)^T \\ \vec{Q}_{outflow} = \left(\rho_{cd} ,\ \rho_{cd} \vec{V}_{cd} ,\ \dfrac{p_{cd}}{\gamma - 1} + \dfrac{1}{2} \rho_{cd} \vec{V}_{cd}^2 \right)^T \end{cases}\right\} \text{supersonic flow} \tag{8}$$

The fluid-body boundary conditions for an inviscid flow at the wall are extrapolated from the computational domain. In other words, the flow properties except the velocity are all extrapolated from the nearest flow cell. With respect to the velocity, for a nonporous wall, there can be no mass flow into or out of the wall this means that the flow velocity vector immediately adjacent to the wall must be tangent to the wall. If $\vec{n}$ is a unit vector at a point on the surface of the wall, the velocity at the wall can be given as:

$$\vec{V}_{sc} = \vec{V}_{fc} - \left(\vec{V}_{fc} \cdot \vec{n} \right) \cdot \vec{n} \tag{9}$$

Where $\vec{V}_{sc}$ is the velocity at the cut surface and $\vec{V}_{fc}$ is the velocity of the nearest flow cell.

METHOD OF SOLUTION

The system of the conservation equations, in vector compact form equations (6), comprise a set of nonlinear partial differential equations containing both space and time derivatives. The spatial derivatives will be discretized using an upwind-high resolution scheme, namely the Advection Upstream Splitting Method AUSM scheme, see [5]. The time derivatives, on the other hand, will be discretized using a five stage Runge -Kutta method.

Discretization of the Convective Terms, the AUSM Scheme:

The AUSM scheme is one of several available upwind schemes that have the advantage of simplicity in programming without loosing the high accuracy. In order to explain this scheme, the flux $\vec{E}$ in equation (6) will be taken as an example. The flux in the y-direction will be treated similarly. The first step in this scheme is to split the flux $\vec{E}$ into convective and pressure terms, and a cell interface velocity $u_{1/2}$ is introduced which is used later to identify the upstream (upwind) direction, that is;

$$\vec{E} = u_{1/2} \begin{bmatrix} \rho \\ \rho u \\ \rho V \\ \rho H \end{bmatrix}_{L/R} + \begin{bmatrix} 0 \\ P_L^+ + P_R^- \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

The interface velocity is given by:

$$u_{1/2} = a_{L/R}\ M_{1/2} \tag{11}$$

Where $a_{L/R}$ is the sound speed at the left or right side with respect to the cell face. The decision to take either the left or the right value is taken according to the upstream direction. Mathematically, this can be written as:

$$(\bullet)_{L/R} = \begin{cases} (\bullet)_L & \text{if } u_{1/2} \le 0 \\ (\bullet)_R & \text{if } u_{1/2} > 0 \end{cases} \tag{12}$$

Substituting equation (11) into equation (10) leads to:

$$\vec{E} = M_{1/2} \begin{bmatrix} \rho a \\ \rho au \\ \rho aV \\ \rho aH \end{bmatrix}_{L/R} + \begin{bmatrix} 0 \\ P_L^+ + P_R^- \\ 0 \\ 0 \end{bmatrix} \tag{13}$$

The convective interface Mach number $M_{1/2}$ is defined by combining the wave speed of left and right running waves, that is;

$$M_{1/2} = M_L^+ + M_R^- \tag{14}$$

In defining the split Mach number, M_L^+ and M_R^-, Liou and Steffen [5] have provided a second degree polynomial to be used if the flow is subsonic and a first degree polynomial if the flow is supersonic, that is;

$$M_{L/R}^{\pm} = \begin{cases} \pm\frac{1}{4}\left(M_{L/R} \pm 1\right)^2 & \text{if } \left|M_{L/R}\right| \le 1 \\ \frac{1}{2}\left(M_{L/R} \pm \left|M_{L/R}\right|\right) & \text{if } \left|M_{L/R}\right| > 1 \end{cases} \tag{15}$$

The pressure part of the flux $\vec{E}$ is calculated similarly by two polynomials provided by [5] as:

$$P_{L/R}^{\pm} = \begin{cases} \pm\frac{P}{4}\left(M_{L/R} \pm 1\right)^2 \left(2 \pm M_{L/R}\right) & \text{if } \left|M_{L/R}\right| \le 1 \\ \frac{P}{2}\left(M_{L/R} \pm \left|M_{L/R}\right|\right) / M_{L/R} & \text{if } \left|M_{L/R}\right| > 1 \end{cases} \tag{16}$$

It should be noted here that the pressure term of the flux does not undergo any upwinding as it is clear from equation (13). This is expected because pressure waves can travel in all directions.

Discretization of the Unsteady Terms (Runge–Kutta method) :

The discretization of the unsteady terms can be performed using either an explicit or an implicit time integration scheme. In order to avoid the greater memory requirement of the implicit scheme an explicit five stage Runge-Kutta integration scheme of the following form is selected:

$$\vec{Q}^{(0)} = \vec{Q}^{(n)}$$

$$\vec{Q}^{(1)} = \vec{Q}^{(0)} + \alpha_1 \Delta t \cdot \mathrm{Re}\, s\left(\vec{Q}^{(0)}\right)$$

$$\vec{Q}^{(2)} = \vec{Q}^{(1)} + \alpha_2 \Delta t \cdot \mathrm{Re}\, s\left(\vec{Q}^{(1)}\right)$$

$$\vec{Q}^{(3)} = \vec{Q}^{(2)} + \alpha_3 \Delta t \cdot \mathrm{Re}\, s\left(\vec{Q}^{(2)}\right)$$

$$\vec{Q}^{(4)} = \vec{Q}^{(3)} + \alpha_4 \Delta t \cdot \mathrm{Re}\, s\left(\vec{Q}^{(3)}\right)$$

$$\vec{Q}^{(5)} = \vec{Q}^{(4)} + \alpha_5 \Delta t \cdot \mathrm{Re}\, s\left(\vec{Q}^{(4)}\right)$$

$$\vec{Q}^{(n+1)} = \vec{Q}^{(5)}$$

The values of the coefficients $\alpha_{1\,to\,5}$ are given, as in [2], by:

$\alpha_1 = 0.059$, $\alpha_2 = 0.14$, $\alpha_3 = 0.273$, $\alpha_4 = 0.5$ and $\alpha_5 = 1$.

The time step can be computed by the procedure given by [6] as:

$$\frac{1}{\Delta t} = \frac{1}{\Delta t_x} + \frac{1}{\Delta t_y} \quad \text{where:} \quad \Delta t_x = \frac{CFL * \Delta x}{|u| + a} \text{ , and } \quad \Delta t_y = \frac{CFL * \Delta y}{|V| + a}$$

Here CFL is the Courant-Fredrich-Levy number that has a value of 1.0 which is normally recommended for explicit schemes.

RESULTS AND DISCUSSION

Results of Grid Generation:

A 2D unstructured Cartesian grid for the NACA0012 airfoil, generated by the algorithm explained in this paper, is shown in Figure (1). As can be clearly seen, the grid generated represents precisely this well known symmetrical airfoil. A longitudinal cross-section of the transportation jet B -747 is taken from [7] and shown in Figure (2). This figure was digitized using the commercial software "get Data", see [8], in order to extract a set of data points that represent the configuration of the airplane. This data was fed to the grid generation algorithm described as an input data file that contains a set of points with each point represented by its x and y coordinates and the resulting grid is shown in Figure (3). Comparing Figures (2) and (3), it is noticed that the resulting Cartesian grid is a good representation of the given plane fuselage.

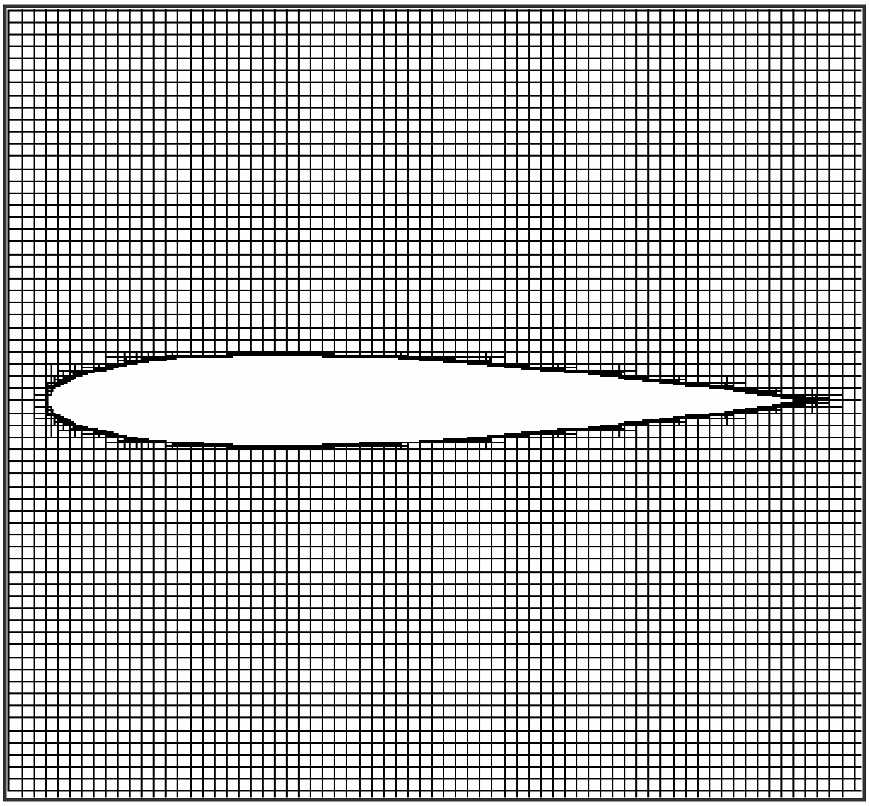

Figure 1: A 2D unstructured Cartesian grid of the NACA0012 airfoil generated by the ray –casting method.

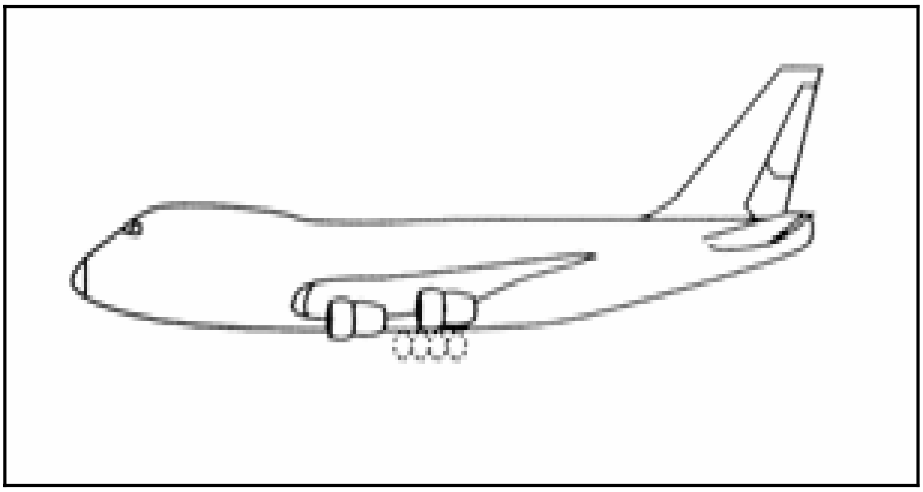

Figure 2: The geometrical configuration of the B-747 jet, as given by [7].

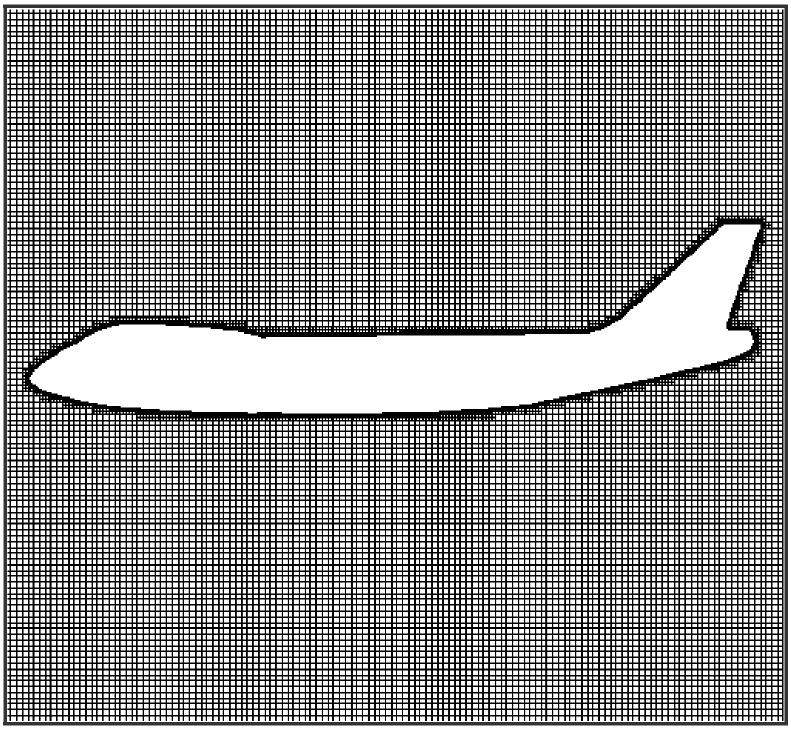

Figure 3: The grid of the B-747 jet generated by the ray –casting method .

Solution of the Inviscid Flow past NACA 0012 Airfoil:

The standard test cases of the AGARD working group 07, see [9], were chosen as validation test cases for the methods explained for flow solving of the inviscid flow. These standard test cases are summarized in Table (1).

Table 1: Values of the Mach number and the angle of attack of the AGARD test cases 01 to 04

Test case	M_{∞}	Angle of attack θ^{o}
AGARD01	0.8	1.25
AGARD02	0.85	1.00
AGARD03	0.95	0.00
AGARD04	1.2	0.00

The AGARD 01 test case is characterized by two shock waves on the upper and lower surfaces of the airfoil. The lower shock wave is weaker than the upper one, and catching this weak shock wave is a measure of the accuracy of the solution. The Mach number contours and the pressure contours are shown in Figures (4) and (5).

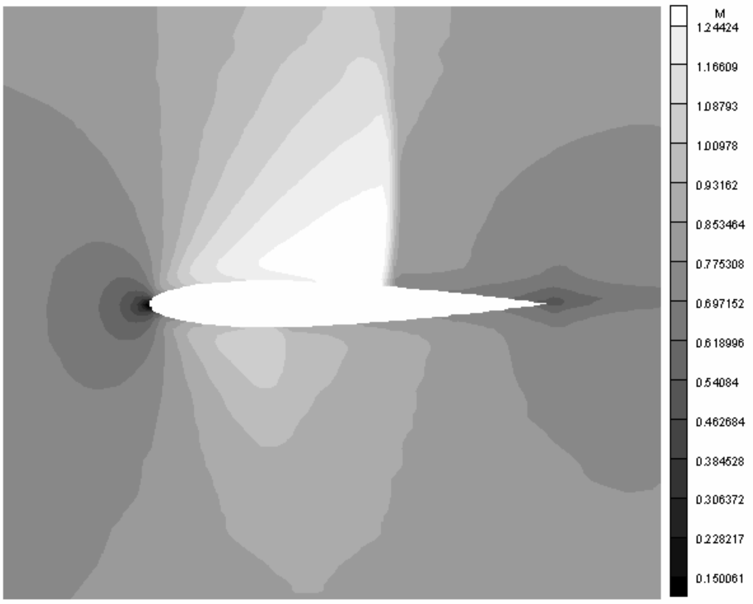

Figure 4: Mach number contours for AGARD01 test case.

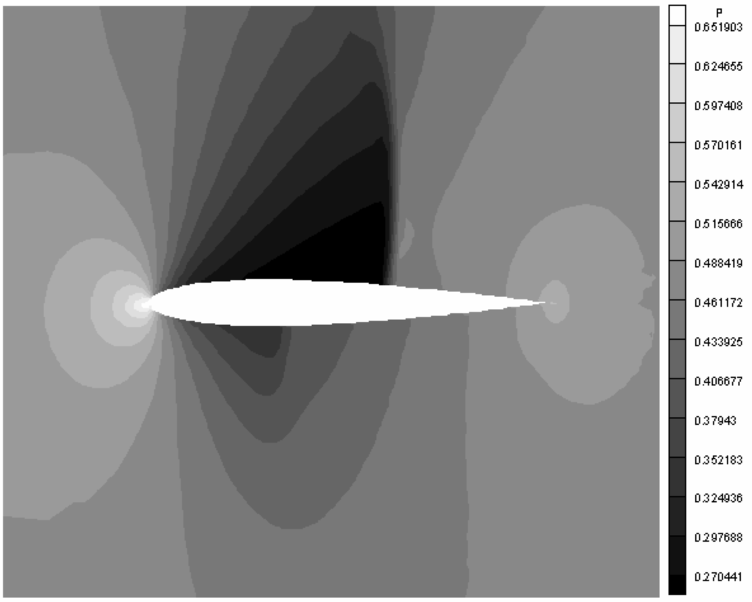

Figure 5: pressure contours for AGARD01 test case.

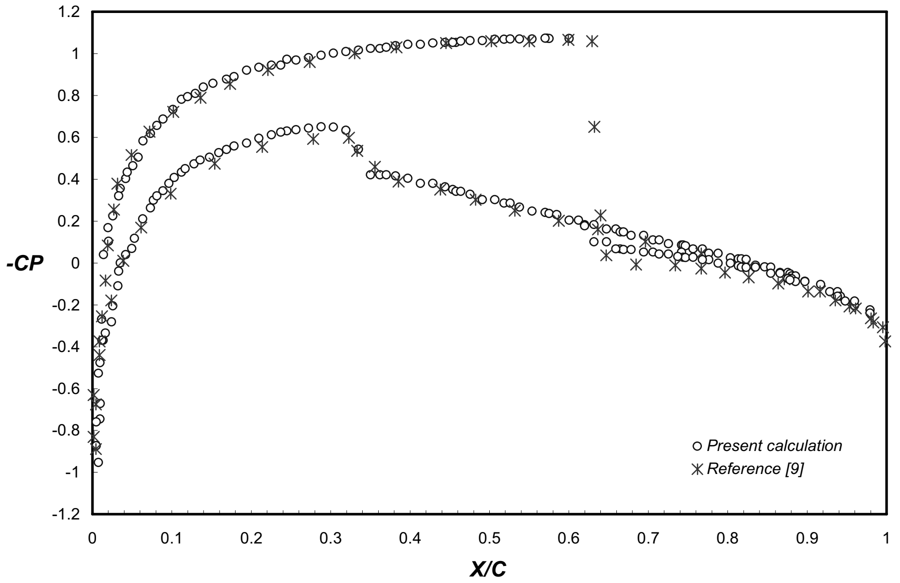

Figure 6: CP distribution on the upper and lower surfaces of the airfoil for AGARD01 test case.

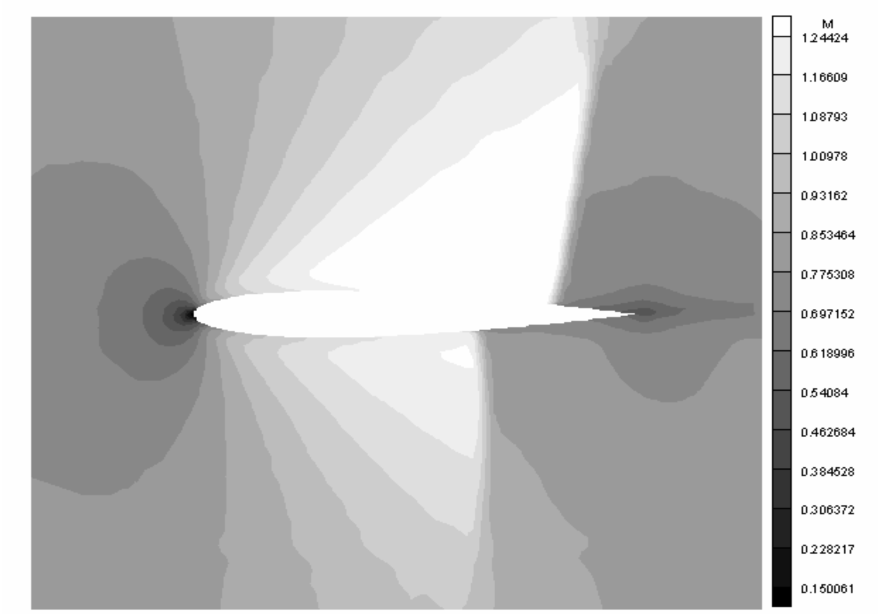

Figure 7: Mach number contours for AGARD02 test case

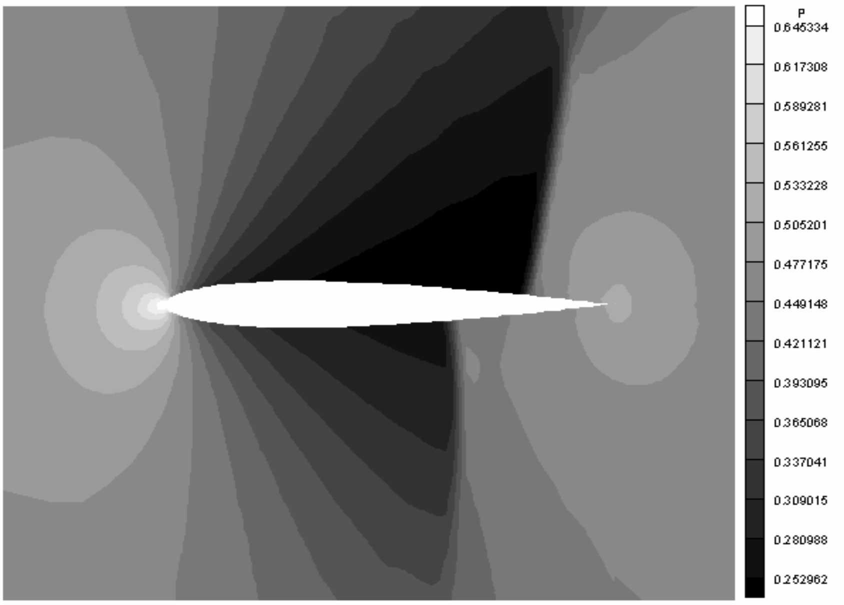

Figure 5: pressure contours for AGARD02 test case.

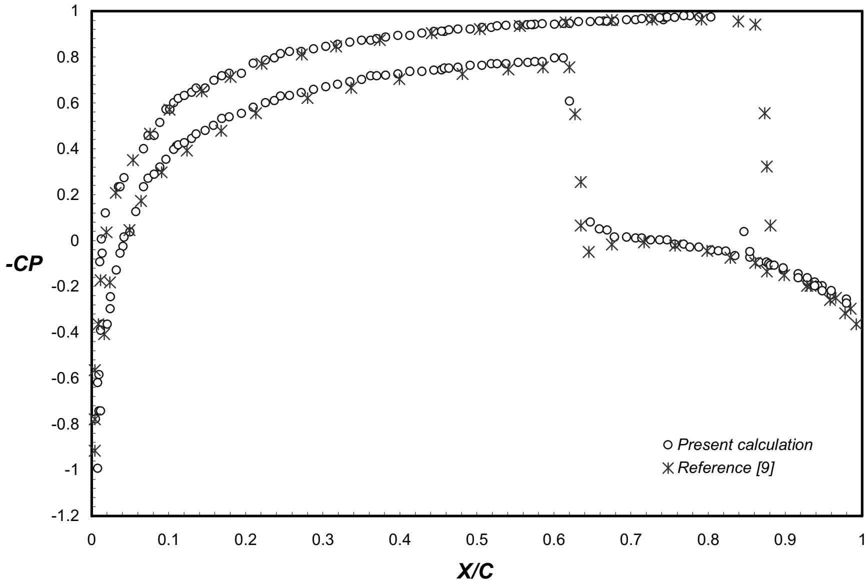

Figure 9: CP distribution on the upper and lower surfaces of the airfoil for AGARD02 test case.

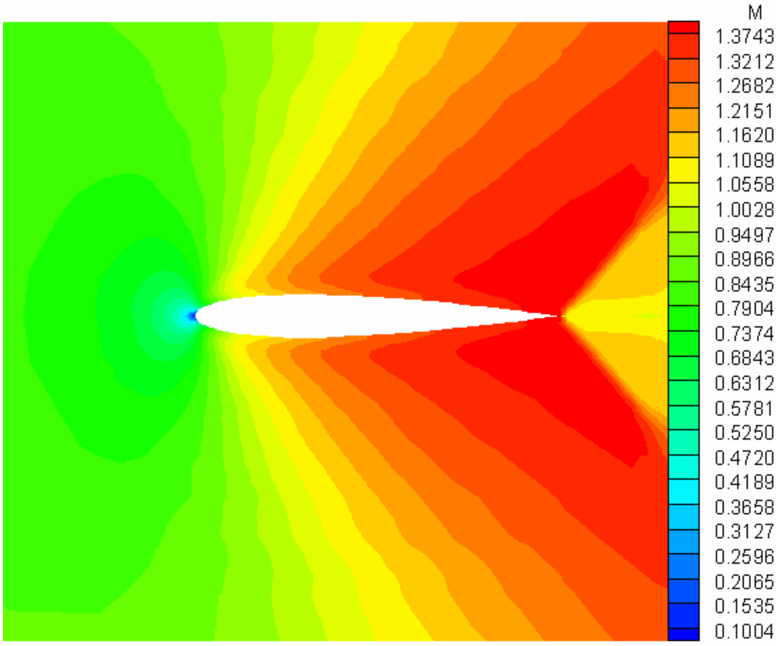

Figure 10: Mach number contours for AGARD03 test case.

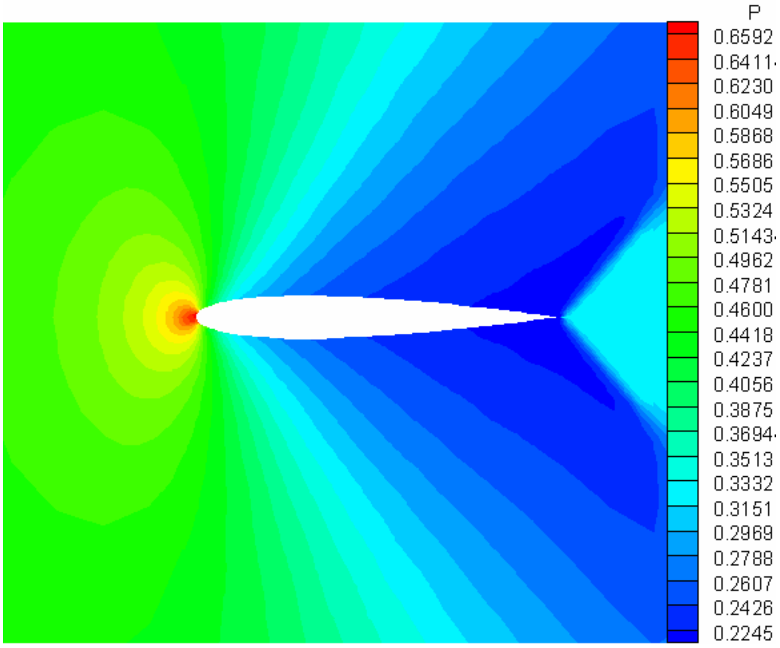

Figure 11: pressure contours for AGARD03 test case.

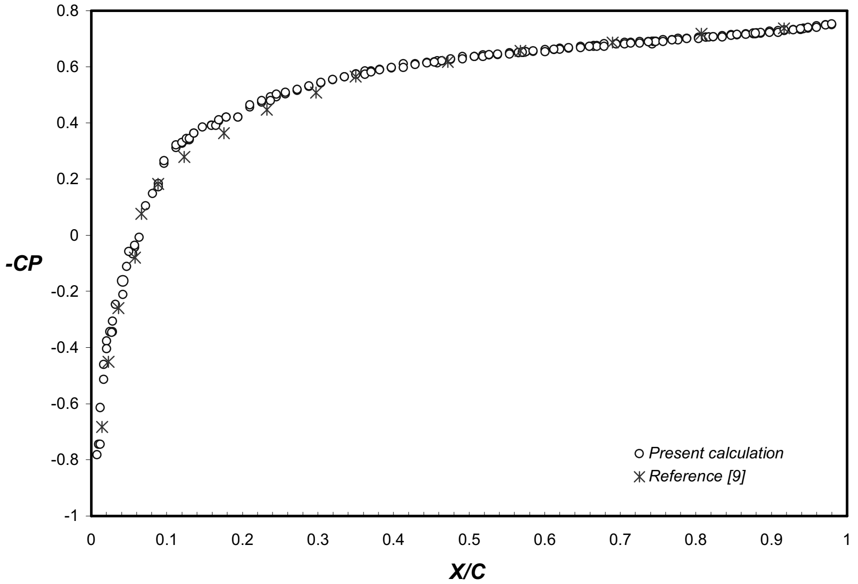

Figure 12: CP distribution on the upper and lower surfaces of the airfoil for AGARD03 test case.

The CP distribution on the upper and lower surfaces of the airfoil is shown in Figure (6) compared to the solution given by [9] who have used body-fitted structured grids to obtain their solution. The agreement of the present calculation with that of Pulliam and Barton [9] despite the different grids used in the two solutions is obvious.

Similar observations are made for the AGARD02 test case with the only exception that the two shock waves on the upper and lower surfaces of the airfoil are of comparable strengths; see Figures (7) to (9).

The transonic flow of the AGARD03 test case is denoted as the fish tail shock wave because two oblique shock waves are emanating from the trailing edge with an angle of about 55° as measured from the chord, as shown in Figures (10) and (11). A comparison of the CP distribution is shown in Figure (12) for the upper surface and the lower surface. Again it is observed that there is a good agreement between the two solutions.

The AGARD04 is similar to the AGARD03 except that the detached shock in front of airfoil nose is more obvious as seen in Figures (13) and (14).The comparison of the CP distribution as computed in this study with that of reference [9] shows again a good agreement as it is noticed in Figure (15).

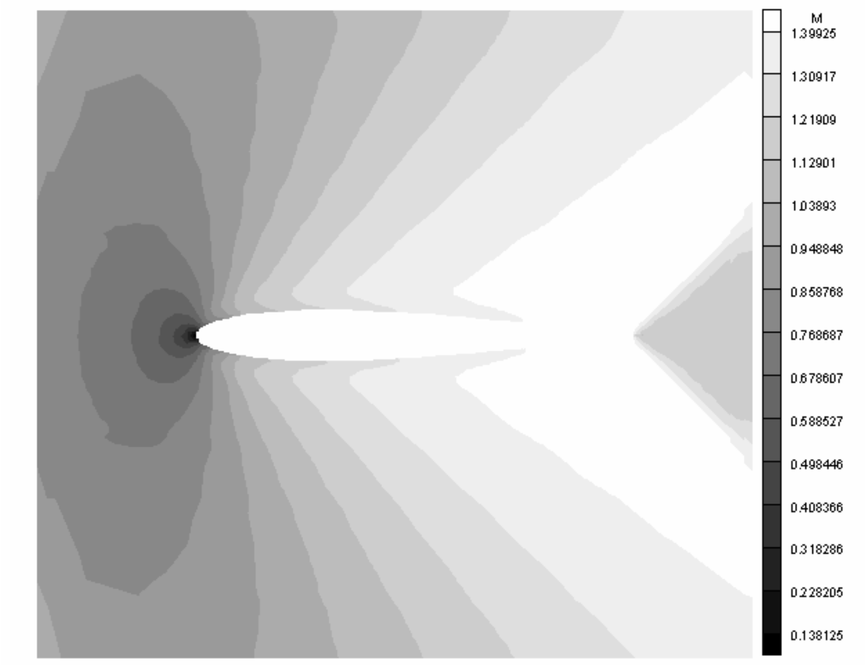

Figure 13: Mach number contours for AGARD02 test case

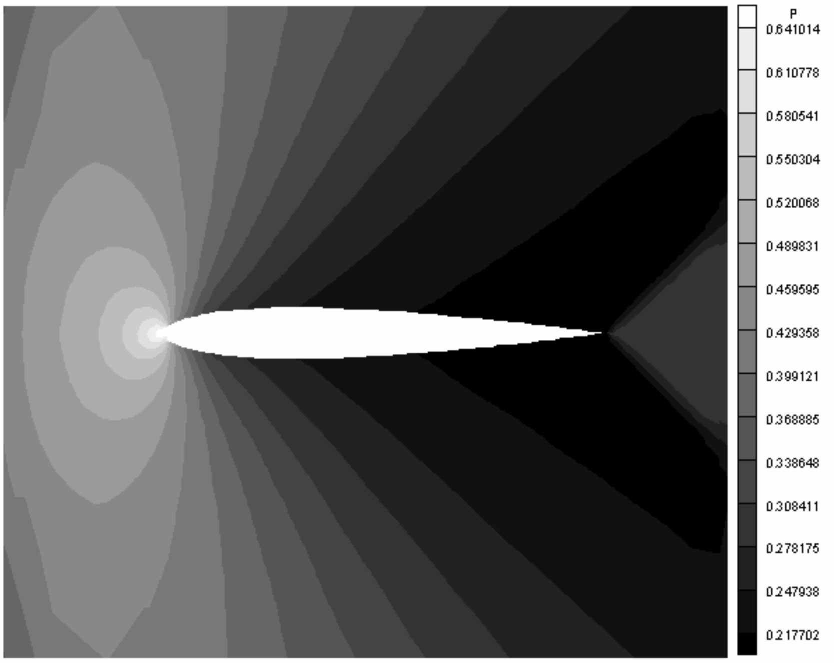

Figure 14: pressure contours for AGARD03 test case.

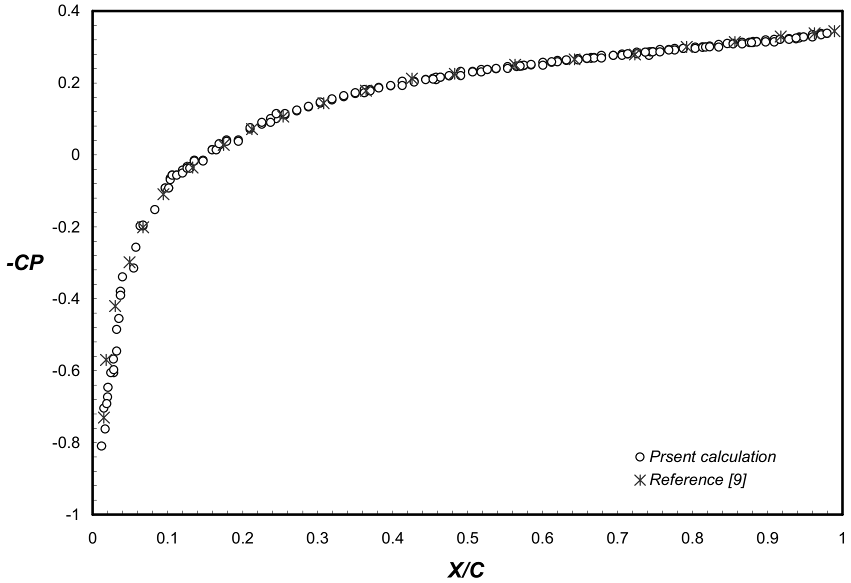

Figure 15: CP distribution on the upper and lower surfaces of the airfoil for AGARD04 test case

CONCLUSIONS

The method of grid generation (the ray-casting method) has shown a very good performance despite its simplicity. It is also worth noting that the Bezier curves are very powerful tool that can be simply implemented in a computer program to represent curves of unknown shapes.

Discretization of the convective terms of the conservation laws using the AUSM method together with the use of the five-stage Runge-Kutta method for the discretization of the unsteady terms have led to solutions that are in good agreements with other lengthy and time consuming schemes. Having stated that, one can conclude that writing his own code and / or using an open source – ready to use – softwares provide(s) a very good chance to enrich one's knowledge and experience.

Another conclusion which can be extracted from this work is that the Cartesian approach for flow simulations is easier to both understand and program than the body fitted grid approach since the former does not include any transformation of the physical domain to a computational domain.

ACKNOLEDGEMENT

The first author would like to thank Dr. Ehab Fares for the fruitful discussions and the scientific cooperation during the time of work at the Aerodynamic Institute of the RWTH-AACHEN University in Germany.

REFERENCES

[1] De Zeeuw, D. and Powell, K. G., "An Adaptively Refined Cartesian Mesh Solver for the Euler Equations", J. Comp. Phys., 104, 56-68 (1993)

[2] Fares, E., Meinke, M., and Schroeder, W., ”Numerical Simulation of the Interaction of Wingtip Vortices and Engine Jets in the Near Field”, AIAA paper 2000-2222 , (2000).

[3] Opela, Mario, ”Grobstruktur-Simulation der Interaktion des Nachlaufs eines bewegten Kreiszylinders mit einer Turbinenschaufel”, Ph.D. Dissertation (in German), Aerodynamisches Institut Aachen –RWTH AACHEN, (2003).

[4] Hirch, C., ”Numerical Computation of Internal and External Flow”, Vol. 2, John Wiley and Sons, (1992).

[5] Liou, M, S. and Steffen, C. J., ”A New Flux Splitting Scheme”, J. Comp. Phys., 107, 23 -39 (1993).

[6] Gaffney, R., Hassan, H., and Salas, M., ”Euler Calculations for Wings Using Cartesian Grids”, AIAA paper 87 -0356, (1987).

[7] Nelson, R. C., ”Flight Stability and Automatic Control”, second Edition, McGraw Hill company, (1998).

[8] GetData, a commercial digitizing tool available at http://www.libexpat.org/.

[9] Pulliam, T. and Barton, J., ”Euler Computation of AGRAD Working Group 07 Test Cases”, AIAA paper 85 -0018, (1985).

NOMENCLATURES:

Latin

a	speed of sound, m/s
C_v	specific heat at constant volume, kJ/(kg K)
c	Chord length, m
CP	Coefficient of pressure
e	Specific total energy, kJ/kg
e^*	Dimensionless specific total energy e / a_o^2
L_x, L_y	Lengths of the computational domain in x and y directions, m
M	Mach number
n_x, n_y	Number of cells in x and y directions
P	Pressure, kPa
P^*	Dimensionless pressure $P/(\rho_o a_o^2)$
R	Gas constant, kJ/(kg K)
T	Temperature, K
T^*	Dimensionless temperature T / T_o
t	Time, s
t^*	Dimensionless time $\frac{t \cdot a_o}{c}$
u	Component of velocity vector in x direction, m/s
$\hat{u}$	Specific internal energy, kJ/kg
u^*	Dimensionless component of velocity vector in x direction u / a_o
$\vec{V}$	Velocity vector
V	Component of velocity vector in y direction, m/s
V^*	Dimensionless component of velocity vector in y direction V / a_o
x	Horizontal coordinate, m

y	Vertical coordinate, m

Greek

ρ	Density, kg/m^3
ρ^*	Dimensionless density ρ / ρ_o
θ	Angle of attack, radian
γ	specific heat ratio

Subscripts

o	Stagnation condition
∞	Free stream value
fc	The nearest flow cell to the cut surface
cd	Computational domain
sc	Cut surface

Superscripts

*	Dimensionless variables
n	Solution at time level n
n+1	Solution at time level n+1