COMPUTATIONAL AERODYNAMICS

LABORATORY MANUAL

B.TECH
(IV YEAR – I SEM)
(2018-19)

Department of Aeronautical Engineering

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956

Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – A Grade - ISO 9001:2015 Certified)

Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, India.

DEPARTMENT OF AERONAUTICAL ENGINEERING

VISION

Department of Aeronautical Engineering aims to be indispensable source in Aeronautical Engineering which has a zeal to provide the value driven platform for the students to acquire knowledge and empower themselves to shoulder higher responsibility in building a strong nation.

MISSION

a) The primary mission of the department is to promote engineering education and research.

(b) To strive consistently to provide quality education, keeping in pace with time and technology.

(c) Department passions to integrate the intellectual, spiritual, ethical and social development of the students for shaping them into dynamic engineers.

PROGRAMME EDUCATIONAL OBJECTIVES (PEO'S)

PEO1: PROFESSIONALISM & CITIZENSHIP To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical, ecological and economic issues.
PEO2: TECHNICAL ACCOMPLISHMENTS To provide knowledge based services to satisfy the needs of society and the industry by providing hands on experience in various technologies in core field.
PEO3: INVENTION, INNOVATION AND CREATIVITY To make the students to design, experiment, analyze, interpret in the core field with the help of other multi disciplinary concepts wherever applicable.
PEO4: PROFESSIONAL DEVELOPMENT To educate the students to disseminate research findings with good soft skills and become a successful entrepreneur.
PEO5: HUMAN RESOURCE DEVELOPMENT To graduate the students in building national capabilities in technology, education and research.

PROGRAM SPECIFIC OBJECTIVES (PSO's)

1. To mould students to become a professional with all necessary skills, personality and sound knowledge in basic and advance technological areas.
2. To promote understanding of concepts and develop ability in design manufacture and maintenance of aircraft, aerospace vehicles and associated equipment and develop application capability of the concepts sciences to engineering design and processes.
3. Understanding the current scenario in the field of aeronautics and acquire ability to apply knowledge of engineering, science and mathematics to design and conduct experiments in the field of Aeronautical Engineering.
4. To develop leadership skills in our students necessary to shape the social, intellectual, business and technical worlds.

PROGRAM OBJECTIVES (PO'S)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
12. **Life- long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

IV Year B. Tech, ANE-I Sem

L	T/P/D	C
0	-/3/-	2

(R15A2186)COMPUTATIONAL AERODYNAMICS LAB

Objectives:

- To develop an understanding for the major theories, approaches and methodologies used in CFD.
- To build up the skills in the actual implementation of CFD methods (e.g. boundary conditions, turbulence modeling etc.) in using commercial CFD codes.
- To gain experience in the application of CFD analysis to real engineering designs.

LIST OF EXPERIMENTS:

1. Solution for the one dimensional wave equations using explicit method of lax using finite Difference method (code development)
2. Solution for the one dimensional heat conduction equation using explicit method using finite difference method (code development)
3. Generation of the Algebraic Grid (code development)
4. Generation of the Elliptic Grids (code development)
5. Numerical simulation of Flow over an airfoil using commercial software
6. Numerical simulation of Supersonic flow over a wedge using commercial Software
7. Numerical simulation of Flat plate boundary layer using commercial software
8. Numerical simulation of Laminar flow through pipe using commercial software
9. Numerical simulation of Flow past cylinder using commercial software
10. Numerical simulation of flow through nozzle using commercial software
11. Numerical simulation of flow over wing using commercial software
12. Numerical simulation of combustion using commercial software

Note: Any 10 Experiments can be conducted.

Equipment Needed:

1. **Computers**: Core 2 duo processor with 1 GB RAM
2. **Softwares:** Matlab or scilab and Ansys or equivalent softwares

Reference Books:

1. MATLAB an Introduction with Applications Fifth Edition AMOS GILAT by WILEY Publications
2. Programming in SCI lab by VINU V DAS New Age International Publications
3. ANSYS FLUENT and CFX Tutorials

Outcomes:

- Students will develop a better intuition of Aerodynamics more quickly than is possible with traditional analytical approaches.
- Ability to undertake problem identification, formulation and solution and apply knowledge of basic science and engineering fundamentals.
- Developing a geometrical model of the flow, applying appropriate boundary conditions, specifying solution parameters, and visualizing and analyzing the results.

CONTENTS

CODE OF CONDUCT FOR THE LABORATORIES

- All students must observe the Dress Code while in the laboratory.
- Sandals or open-toed shoes are NOT allowed.
- Foods, drinks and smoking are NOT allowed.
- All bags must be left at the indicated place.
- The lab timetable must be strictly followed.
- Be PUNCTUAL for your laboratory session.
- Program must be executed within the given time.
- Noise must be kept to a minimum.
- Workspace must be kept clean and tidy at all time.
- Handle the systems and interfacing kits with care.
- All students are liable for any damage to the accessories due to their own negligence.
- All interfacing kits connecting cables must be RETURNED if you taken from the lab supervisor.
- Students are strictly PROHIBITED from taking out any items from the laboratory.
- Students are NOT allowed to work alone in the laboratory without the Lab Supervisor
- USB Ports have been disabled if you want to use USB drive consult lab supervisor.
- Report immediately to the Lab Supervisor if any malfunction of the accessories, is there.

Before leaving the lab

- Place the chairs properly.
- Turn off the system properly
- Turn off the monitor.
- Please check the laboratory notice board regularly for updates.

INTRODUCTION TO MODELING AND SIMULATION SOFTWARE TO AERODYNAMIC PROBLEMS

A **model** is a mathematical object that has the ability to predict the behavior of a real system under a set of defined operating conditions and simplifying assumptions.The term *modeling refers to the development of a mathematical representation of a physical situation*

WHAT IS MODELING?

- Modeling is the process of producing a model.
- A model is a representation of the construction and working of some system of interest.
- A model is similar to but simpler than the system it represents.
- One purpose of a model is to enable the analyst to predict the effect of changes to the system. Generally, a model intended for a simulation study is a mathematical model developed with the help of simulation software.
- Mathematical model classifications include
- Deterministic (input and output variables are fixed values) or Stochastic (at least one of the input or output variables is probabilistic);
- Static (time is not taken into account) or
- Dynamic (time-varying interactions among variables are taken into account).
- Typically, simulation models are stochastic and dynamic.

WHAT IS SIMULATION?

Simulation is the process of exercising a model for a particular instantiation of the system and specific set of inputs in order to predict the system response. simulation refers to the procedure of solving the equations that resulted from model development

- A simulation of a system is the operation of a model of the system.
- The operation of the model can be studied, and hence, properties concerning the behavior of the actual system or its subsystem can be inferred.
- In its broadest sense, simulation is a tool to evaluate the performance of a system, existing or proposed, under different configurations of interest and over long periods of real time.
- Simulation is used
- before an existing system is altered or a new system built,
- to reduce the chances of failure to meet specifications,
- to eliminate unforeseen bottlenecks,
- to prevent under or over-utilization of resources,
- to optimize system performance.

INTRODUCTION TO AERODYNAMICS

The study of the dynamics of air is known as aerodynamics. It includes the intermolecular forces, the motion of molecules due to variation of flow field variables like pressure, velocity, temperature etc. Aerodynamics is an applied science with many practical applications in engineering. No matter how elegant an aerodynamic theory may be, or how mathematically complex a numerical solution may be, or how sophisticated an aerodynamic experiment may be, all such efforts are usually aimed at one or more of the following practical objectives:

1. The prediction of forces and moments on, and heat transfer to, bodies moving through air. For example:
   - Estimation of lift, drag and moments of airfoils, wings, fuselages, engine nacelles and whole airplane.
   - Aerodynamic heating of flight vehicles ranging from the supersonic transport to planetary probe entering the atmosphere of Jupiter.
2. Determination of flows moving internally through ducts. For example:
   - To calculate and measure the flow properties in compressors, combustion chamber, nozzle of rockets and air breathing jet engines and to calculate engine thrust.
   - To know the flow conditions in the test section of a wind tunnel.
   - To know how much fluid can flow through pipes under different conditions.
   - A very interesting application of aerodynamics is high-energy chemical and gas dynamic lasers.

These types of problems are solved in Computational Aerodynamics lab in two ways as follows:

(i) Code development like MATLAB

(ii) Commercial CFD Software like ANYS

EXPERIMENT 1
ONE DIMENSIONAL WAVE EQUATION

Aim: to write a MATLAB code for the solution of one dimensional Wave using explicit of LAX method

The one dimensional scalar wave equation is given as

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

This equation represents a linear advection process with wave speed c = constant, which is the speed of the travelling wave or the speed of propagation. u(x,t) is the signal or wave information. The wave propagates at constant speed to the right if c > 0 and to the left if c <1. The spatial domain can vary from -∞ to ∞. Suppose the initial conditions are

$$u(x,0) = u_0(x)$$

where $u_0(x)$ is any function. The exact solution to the wave equation then is

$$u = u_o(x - ct)$$

$u_0(x)$ is called the wave shape of wave form. Travelling or propagation here means that the shape of the signal function with respect to x stays constant, however the function is translated left or right with time at the speed c.

Numerical Solution

Method of descretisation – finite difference form

Replace the spatial partial derivative with a central difference expression

$$\frac{\partial u}{\partial x} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}$$

Where n is the temporal index and j is the spatial

index. Replace the time derivative with a forward

difference formula

$$\frac{\partial u}{\partial t} = \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

We then have

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -c\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \qquad (1)$$

Now let us replace u_j^n by an average value between grid points j+1 and j-1 as

$$u_j^n = \frac{u_{j+1}^n + u_{j-1}^n}{2}$$

Substituting this in equation (1) we get the explicit method of Lax for the 1D scalar wave equations as,

$$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - c\frac{\Delta t}{\Delta x}\frac{u_{j+1}^n - u_{j-1}^n}{2}$$

Test Case for the numerical solution

Solve the one dimensional wave equation in the spatial domain of [0, 2*pi] with an initial

step function

condition given by

$U_0(x,0) = 1$ for x $\leq$

pi-1

= 0 otherwise

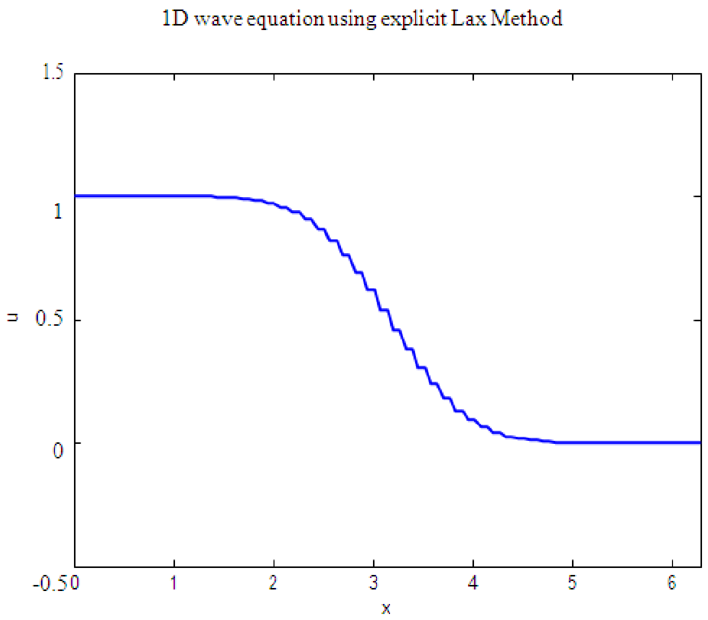
Choose 100 grid points and find the wave form at t = 0.2 s.

Matlab code for the one dimensional wave equation

```
% Solves the one dimensional scalar wave equation du/dt + du/dx = 0
[0,2*pi]
% Using LAX METHOD
clc;
clear all;
t0 = 0;
tf = 1;
M = 100; % number of points in x
direction
N = 100; % number of points in y
direction

% define the mesh in
space
dx =2*pi/M;
x = 0:dx:2*pi;

% define the mesh in
time
dt = (tf-t0)/N;
t = t0:dt:tf;

% calculate value for
lamda
c = 1;
lambda = c*dt/dx
display('lambda should be less than 1 for stability:')
% choose the wave number of the initial data and give its decay rate
 u0 = x<=(pi-1);
```

```
u = zeros(M+1,N+1);
u(:,1) = u0;

% Implement the time marching Lax scheme:
for n=1:N
 for i=2:M
u(i,n+1)=(u(i+1,n)+u(i-1,n))/2-(lambda/2)*(u(i+1,n)-u(i-1,n));
         end
% Introduce exact values at the endpoints.
          u(1,n+1)=1;
         u(M+1,n+1)=0;
end
% plot the result in 21 intervals
for  j=0:20
plot(x,u(:,1+5*j),'LineWidth',2);
axis([0,2*pi,-0.5,1.5]);
title('1D wave equation using explicit Lax Method','FontSize',12)
 xlabel('x');
ylabel('u');
pause(1)
end
%plot(x,u(:,101));
```

Results:



Exercise problems:

1.1 Write a Matlab Program to solve one dimensional wave equation using FTCS method.

1.2 Write a Matlab program to find wave propagation at t=0.5 secs

EXPERIMENT 2
ONE DIMENSIONAL TRANSIENT HEAT CONDUCTION

Aim: to write a MATLAB code for the solution of one dimensional transient heat conduction equation using explicit method

The one dimensional transient (unsteady) heat conduction equation is given as

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

Where α is the thermal diffusivity

This equation represents the conduction of heat energy in time and space. Transient nature of this equation is represented in the dependence of temperature with time as opposed to a steady state condition.

Numerical Solution

Method of discretization – finite difference form

Replace the time derivative with a forward difference expression

$$\frac{\partial T}{\partial t} = \frac{T_j^{n+1} - T_j^n}{\Delta t}$$

Where n is the temporal index and j is the spatial index.
Replace the second order spatial derivative on the RHS with a central difference formula

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{j+1}^n + T_{j-1}^n - 2T_j^n}{\Delta x^2}$$

We then have

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \alpha \frac{T_{j+1}^n + T_{j-1}^n - 2T_j^n}{\Delta x^2} \qquad (1)$$

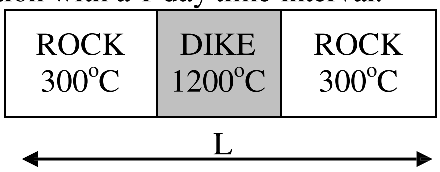
i.e. $$\boldsymbol{T_j^{n+1} = (1 - 2A)T_j^n + AT_{j+1}^n + AT_{j-1}^n} \qquad (2)$$

where $$A = \alpha \frac{\Delta t}{\Delta x^2}$$

Equation (2) is the final explicit update equation for the one dimensional transient heat conduction equation.

Test Case for the numerical solution

A country rock has a temperature of 300^{o}C and the dike a width of 5m, with a magma temperature of 1200^{o}C. Total length of the rock formation is 100m. Initial conditions are temperatures of 300^{o}C and 1200^{o}C for the rock and dike respectively. Boundary conditions at x = -L/2 and x = L/2 are at 300^{o}C (see figure). Find the temperature distribution after 100 days. Use 200 grid points in the x direction with a 1 day time interval.

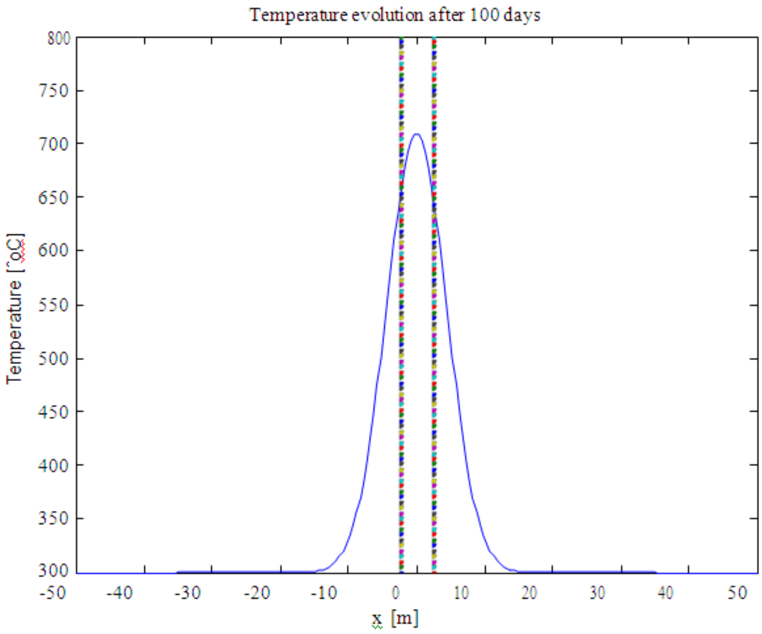


Matlab code for the one dimensional transient heat conduction equation

```
% Solves the 1D heat equation with an explicit finite difference scheme
clear all
clc
%Physical parameters
L = 100; % Length of modeled domain [m]
Td = 1200; % Temperature of magma [0C]
Tr = 300; % Temperature of country rock [0C]
kappa = 1e-6; % Thermal diffusivity of rock [m2/s]
W = 5; % Width of dike [m]
day = 3600*24; % # seconds per day
dt = 1*day; % Timestep [s]
% Numerical parameters
nx = 200; % Number of gridpoints in x-direction
nt = 100; % Number of timesteps to compute
dx = L/(nx-1); % Spacing of grid
 x = -L/2:dx:L/2;% Grid

% Setup initial temperature profile
T = ones(size(x))*Tr;
T(abs(x)<=W/2) = Td;
time = 0;
for n=1:nt % Timestep loop
% Compute new temperature
Tnew = zeros(1,nx);
for i=2:nx-1
Tnew(i) = T(i) + (kappa*dt/(dx)^2)*(T(i+1)-(2*T(i))+T(i-1));
end
% Set boundary conditions
Tnew(1) = T(1);
Tnew(nx) = T(nx);
% Update temperature and time
T = Tnew;
time = time+dt;
end
% Plot solution
plot(x,Tnew);
xlabel('x [m]')
ylabel('Temperature [^oC]')
title(['Temperature evolution after ',num2str(time/day),' days'])

% draw the dike boundaries
x1 = -2.5;
```

```
x2 = 2.5;
y = linspace(300,800);
% Plot the dike
boundaries
hold on
plot(x1,y, x2, y);
```

Results:



Exercise problems:

2.1 Write a Matlab Program to solve one dimensional heat conduction equation using Lax method.

2.2Write a Matlab program to one dimensional heat conduction equation after 200 days.

EXPERIMENT 3
GENERATION OF THE ALGEBRAIC GRIDS

Aim: To write a MATLab code for generating elliptic grid over airfoil.

Problem

Generate an algebraic grid about the upper surface of the airfoil. Points are clustered in j direction near the lower surface (using β=1.05 in algebraic grid). Make sure the number of points in i and j are flexible.

Introduction, Theory, & Formulations

A key component of grid generation is the conversion from the physical domain to the computational domain, in order to allow for equidistant grid lines in rectangular form. In considering a simple two dimensional case, physical coordinates x and y must be converted to computational coordinates ξ and η. These computational coordinates are furthermore known via the rectangular grid relations. As a result, they must be converted back into physical coordinates in order to be of use. For the particular case concerning an airfoil placed on the x axis, the following relationships exist:

$$x = \xi \tag{1}$$

$$y = H.\frac{(\beta+1)-(\beta-1).(\frac{\beta+1}{\beta-1})^{1-\eta}}{(\frac{\beta+1}{\beta-1})^{1-\eta}+1} \tag{2}$$

As can be seen, Eq. (1) simply states that the x coordinate is the ξ coordinate, as there exists no irregularities to alter that axis. The precise relationship in Eq. (2) is due to a required clustering near the bottom surface. Here, β represents the clustering parameter, which is given, and H represents the total height along the y axis. However, this does not account for the geometry of the airfoil, wherein its top surface coordinate is a function of the distance along the x axis. The exact equation is:

$$y = \frac{t}{0.2}.\left(0.2969x^{\frac{1}{2}} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4\right) \tag{3}$$

Here, y represents the max height of the airfoil, which would thus be the correspond to y=0 in Eq.(2). Height is determined by subtracting this value from maximum height. This allows a total expression for the grid y coordinative can be obtained. Note that the x used in Eq. (3) assumes 0 at the nose of the airfoil and 1 at the tail. The previous equations effectively define all that is needed to generate an algebraic grid. However, this grid will simply be used as a starting point for the generation of an elliptic grid. Thus, once x and y are obtained algebraically, they will be set as initial conditions for the x and y values used in order to perform iterations of the developed finite difference equations.

MATLAB code for Algebraic Grid Generation

```
%Algebraic Grid Generation
clear all;
clc;
%Assign values for t and beta
t=0.15;
beta=1.05;
%Prompt user for number of grid points
n=input('Enter the number of grid points in the i direction: ');
m=input('Enter the number of grid points in the j direction: ');
%Create zeroes matrix for surface
plots z=zeros(n,m);
%Assign lengths and values for eta and xi
L=3;
eta=linspace(0,1,m);
xi=linspace(0,L,n);
%x is equal to xi
X=xi;
%Find height
ytop=2;
for i=1:n
if X(i) < 1
ybottom(i)=0;
elseif X(i) > 2
ybottom(i)=0;
else
x2(i)=X(i)-1;
  ybottom(i)=(t/.2)*(0.2969*x2(i)^.5-0.126*x2(i)-0.3516*x2(i)^2+0.2843*x2(i)^3-0.1015*x2(i)^4);
end
  H(i)=ytop-ybottom(i);
  end
%Loop to calculate coordinates
zeta=beta+1;
gamma=beta-1;
alpha=zeta/gamma;
for i=1:n
for
j=1:m
chi=1-
eta(j);
y(i,j)=H(i)*(zeta-gamma*alpha^chi)/(alpha^chi+1)+ybottom(i);
x(i,j)=X(i);
end
end
surface(x,y,z);
xlabel ('x');
ylabel ('y');
title ('Algerbraic Grid');
```

Discussion of Results

Enter the number of grid points in i direction: 50
Enter the number of grid points in the j direction: 50

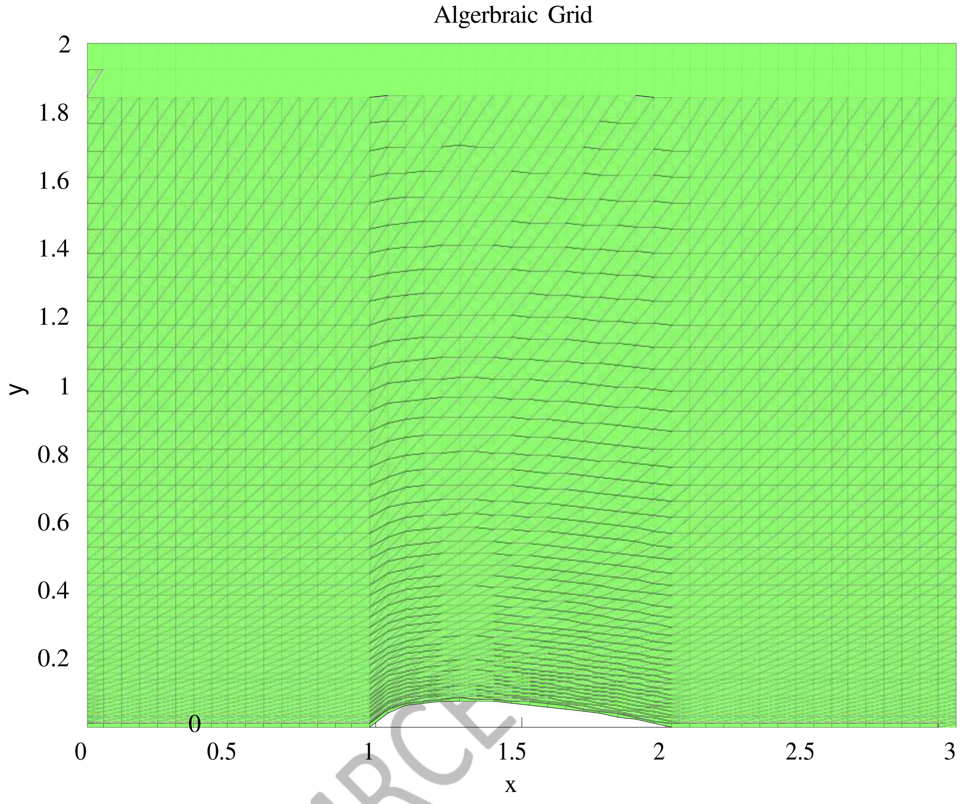


Figure shows the algebraic grid generation with the growth rate β=1.05 the grids are very fine at y=0 and it gets coarser as the y increases.
The value of growth rate β can be varied and you can see the difference in the growth rate of the grid.

Exercise problems:

3.1 Write a Matlab Program to generate algebraic grid over flat plate.

3.2 Write a Matlab program to generate algebraic grid over circle.

EXPERIMENT 4
GENERATION OF THE ELLIPTIC GRIDS

Aim: To write a MATLab code for generating elliptic grid over airfoil.

Problem

Starting with an algebraic grid, generate an elliptic grid about the upper surface of the airfoil. Points are clustered in j direction near the lower surface (using β=1.05 in algebraic grid). Make sure the number of points in i and j are flexible.

Using a predetermined algebraic grid, an elliptic grid can be generated in order to fine tune the results for airfoil flow. Coding an algebraic grid necessitates an accounting for the geometry of the airfoil, as well as clustering via appropriate equations. Once these issues are addressed, partial differential equations can be utilized in order to generate an elliptic grid.

Introduction, Theory, & Formulations

A key component of grid generation is the conversion from the physical domain to the computational domain, in order to allow for equidistant grid lines in rectangular form. In considering a simple two dimensional case, physical coordinates x and y must be converted to computational coordinates ξ and η. These computational coordinates are furthermore known via the rectangular grid relations. As a result, they must be converted back into physical coordinates in order to be of use. For the particular case concerning an airfoil placed on the x axis, the following relationships exist:

$$x = \xi \tag{1}$$

$$y = H.\frac{(\beta+1)-(\beta-1).\left(\frac{\beta+1}{\beta-1}\right)^{1-\eta}}{\left(\frac{\beta+1}{\beta-1}\right)^{1-\eta}+1} \tag{2}$$

As can be seen, Eq. (1) simply states that the x coordinate is the ξ coordinate, as there exists no irregularities to alter that axis. The precise relationship in Eq. (2) is due to a required clustering near the bottom surface. Here, β represents the clustering parameter, which is given, and H represents the total height along the y axis. However, this does not account for the geometry of the airfoil, wherein its top surface coordinate is a function of the distance along the x axis. The exact equation is:

$$y = \frac{t}{0.2}.\left(0.2969x^{\frac{1}{2}} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4\right) \tag{3}$$

Here, y represents the max height of the airfoil, which would thus be the correspond to y=0 in Eq.(2). Height is determined by subtracting this value from maximum height. This allows a total expression for the grid y coordinative can be obtained. Note that the x used in Eq. (3) assumes 0 at the nose of the airfoil and 1 at the tail. The previous equations effectively define all that is needed to generate an algebraic grid. However, this grid will simply be used as a starting point for the generation of an elliptic grid. Thus, once x and y are obtained algebraically, they will be set as initial conditions for the x and y values used in order to perform iterations of the developed finite difference equations.

Two elliptic partial differential equations must be solved in order to fully define the desired grid. In doing this, boundary conditions are required. For this case, x and y values along the edges of the defined physical domain will be left in place. These being predefined allows all interior coordinates to be developed. The following system of elliptic partial differential equations can be used to define the domain:

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0 \qquad (4)$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0 \qquad (5)$$

Here, the subscripts denote second order derivative of that variable. Notice that these equations do not express x and y as dependent variables. Rather, they are treated as the independent variables, requiring a transformation. When such a mathematical transformation is preformed Eqs. (4) And (5) become, respectively:

$$a.\frac{\partial^2 x}{\partial \xi^2} - 2.b.\frac{\partial^2 x}{\partial \xi \partial \eta} + c.\frac{\partial^2 x}{\partial \eta^2} = 0 \qquad (6)$$

$$a.\frac{\partial^2 y}{\partial \xi^2} - 2.b.\frac{\partial^2 y}{\partial \xi \partial \eta} + c.\frac{\partial^2 y}{\partial \eta^2} = 0 \qquad (7)$$

Where,

$$a = \left({\partial x}/{\partial \eta}\right)^2 + \left({\partial y}/{\partial \eta}\right)^2 \qquad (8)$$

$$b = \frac{\partial x}{\partial \xi}.\frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi}.\frac{\partial y}{\partial \eta} \qquad (9)$$

$$c = \left({\partial x}/{\partial \xi}\right)^2 + \left({\partial y}/{\partial \xi}\right)^2 \qquad (10)$$

The previously stated equations must all be expressed in terms of finite differences. Once this is done, x and y at each grid point can be found through iterations. Expanding Equation (8) through (10) explicitly in central space yields:

$$a = \left[\frac{x^n_{i,j+1} - x^{n+1}_{i,j-1}}{2.\Delta\eta}\right]^2 + \left[\frac{y^n_{i,j+1} - y^{n+1}_{i,j-1}}{2.\Delta\eta}\right]^2 \qquad (11)$$

$$b = \left[\frac{x^n_{i+1,j} - x^{n+1}_{i-1,j}}{2.\Delta\xi}\right].\left[\frac{x^n_{i,j+1} - x^{n+1}_{i,j-1}}{2.\Delta\eta}\right] + \left[\frac{y^n_{i+1,j} - y^{n+1}_{i-1,j}}{2.\Delta\xi}\right].\left[\frac{y^n_{i,j+1} - y^{n+1}_{i,j-1}}{2.\Delta\eta}\right] \qquad (12)$$

$$c = \left[\frac{x^n_{i+1,j} - x^{n+1}_{i-1,j}}{2.\Delta\xi}\right]^2 + \left[\frac{y^n_{i,j+1} - y^{n+1}_{i,j-1}}{2.\Delta\xi}\right]^2 \qquad (13)$$

Here, the superscript, n, indexes the iteration, where n is the current iteration and n+1 is the following iteration. These equations are written this way due to the fact that points above and to the right of the point being evaluated are unknown, and, thus, old values must be used. The same procedure of finite differencing can be applied to Eqs. (6) and (7). However, results from these will be of the same form; that is, only the terms x and y will be different. Considering the expansion of Eq. (6) yields:

$$a.\left[\frac{x_{i+1,j}^{n}-2.x_{i,j}^{n+1}+x_{i-1,j}^{n+1}}{(\Delta\xi)^2}\right]-2.b\left[\frac{x_{i+1,j+1}^{n}-x_{i+1,j-1}^{n}-x_{i-1,j+1}^{n}-x_{i-1,j-1}^{n+1}}{4.\Delta\xi.\Delta\eta}\right]+c.\left[\frac{x_{i,j+1}^{n}-2.x_{i,j}^{n+1}+x_{i,j-1}^{n+1}}{\Delta\eta^2}\right]=0 \qquad (14)$$

Considering,

$$\alpha=\frac{a}{(\Delta\xi)^2};\ \beta=\frac{b}{2.\Delta\xi.\Delta\eta};\ \gamma=\frac{c}{\Delta\eta^2}$$

This equation can then be explicitly solved for the value $x_{i,j}^{n+1}$ which is the coordinate of interest. Doing so yields:

$$x_{i,j}^{n+1}=\frac{\alpha.(x_{i+1,j}^{n}+x_{i-1,j}^{n+1})+\beta.(x_{i+1,j+1}^{n}-x_{i+1,j-1}^{n}-x_{i-1,j+1}^{n}-x_{i-1,j-1}^{n+1})+\gamma.(x_{i,j+1}^{n}+x_{i,j-1}^{n+1})}{2.(\alpha+\gamma)}$$

Similarly, Considering the expansion of Eq.(7) and solving it for value of $y_{i,j}^{n+1}$:

$$y_{i,j}^{n+1}=\frac{\alpha.(y_{i,j+1}^{n}+y_{i,j-1}^{n+1})+\beta.(y_{i+1,j+1}^{n}-y_{i+1,j-1}^{n}-y_{i-1,j+1}^{n}-y_{i-1,j-1}^{n+1})+\gamma.(y_{i+1,j}^{n}+y_{i-1,j}^{n+1})}{2.(\alpha+\gamma)}$$

This formula can then be implemented through coding in order to find all values of x. The formulation is exactly the same for the y value. Through code, multiple iterations will occur until convergence is reached; that is, the desired x values will be found once the difference between $x_{i,j}^{n+1}$ and $x_{i,j}^{n}$ is below tolerance and the desired y values will be found once difference between $y_{i,j}^{n}$ and $y_{i,j}^{n}$ falls below said tolerance. These values, when plotted, should produce an elliptic grid that can be utilized to determine flow within the domain containing the airfoil.

MATLAB code for Elliptic Grid Generation

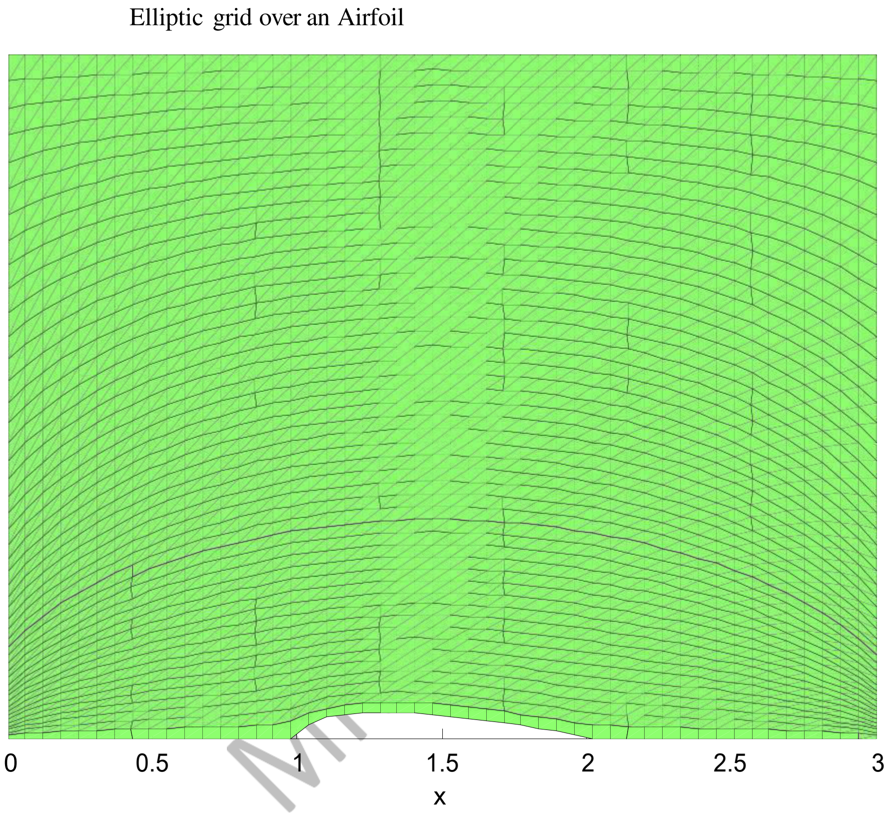
```
%Elliptic Grid Generation
Clear all;
clc;
%Assign values for t and beta
t=0.15;
beta=1.05;
%Prompt user for number of grid points
n=input('Enter the number of grid points in the i direction: ');
m=input('Enter the number of grid points in the j direction: ');
%Create zeroes matrix for surface
plots z=zeros(n,m);
%Assign lengths and values for eta and xi
L=3;
eta=linspace(0,1,m);
xi=linspace(0,L,n);
%x is equal to xi
X=xi;
%Find height
ytop=2;
for i=1:n
if X(i) < 1
ybottom(i)=0;
elseif X(i) > 2
ybottom(i)=0;
else
x2(i)=X(i)-1;
  ybottom(i)=(t/.2)*(0.2969*x2(i)^.5-0.126*x2(i)-0.3516*x2(i)^2+0.2843*x2(i)^3-0.1015*x2(i)^4);
end
  H(i)=ytop-ybottom(i);
  end
%Loop to calculate coordinates
zeta=beta+1;
gamma=beta-1;
alpha=zeta/gamma;
for i=1:n
for
j=1:m
chi=1-
eta(j);
y(i,j)=H(i)*(zeta-gamma*alpha^chi)/(alpha^chi+1)+ybottom(i);
x(i,j)=X(i);
end
end

%Elliptic initial conditions
xold=x;
yold=y;
```

```
%Calculate computational step sizes
delta_eta=1/(m-1);
delta_xi=L/(n-1);
dx=1; %Conditions to start loop
dy=1; %Conditions to start loop
%Assign tolerance value
tol=.0001;
%Nested loop to determine elliptic grid
xdiff=0;
ydiff=0;
count=0;
while dy > tol || dx > tol
for i=2:n-1
for j=2:m-1
a1=(xold(i,j+1)-x(i,j-1))/(2*delta_eta);
a2=(yold(i,j+1)-y(i,j-1))/(2*delta_eta); a=a1^2+a2^2;
c1=(xold(i+1,j)-x(i-1,j))/(2*delta_xi);
c2=(yold(i+1,j)-y(i-1,j))/(2*delta_xi); c=c1^2+c2^2;
b=a1*c1+a2*c2;
alpha=a/delta_xi^2;
beta=-2*b/(4*delta_xi*delta_eta); gamma=c/delta_eta^2; theta=1/(2*alpha+2*gamma);
phi_1=beta*(xold(i+1,j+1)-xold(i+1,j-1)-xold(i-1,j+1)+x(i-1,j-1));
x(i,j)=theta*(alpha*(xold(i+1,j)+x(i-1,j))+gamma*(xold(i,j+1)+x(i,j-1))+phi_1);
xdiff=x(i,j)-xold(i,j)+xdiff;
phi_2=beta*(yold(i+1,j+1)-yold(i+1,j-1)-yold(i-1,j+1)+y(i-1,j-1));
y(i,j)=theta*(alpha*(yold(i+1,j)+y(i-1,j))+gamma*(yold(i,j+1)+y(i,j-1))+phi_2);
  ydiff=y(i,j)-yold(i,j)+ydiff;
  end
end
dx=xdiff; dy=ydiff; xdiff=0;
ydiff=0;
xold=x;
yold=y;
count=c
ount+1;
end
fprintf('The solution took %i iterations to converge. \n \n', count);
surface(x,y,z);
xlabel ('x');
ylabel ('y');
title ('Elliptic grid over an Airfoil');
```

Result and Discussion

In the plot of an Elliptical, the Grid lines have been smoothed out due to the elliptic equations, eliminating extreme jaggedness resulting from the algebraic grid. This would ensure a more accurate flow model.

Enter the number of grid points in the i direction: 50
Enter the number of grid points in the j direction: 50
The solution took 2434 iterations to converge.



Overall, an elliptic grid was shown to provide desired results for discretization. It succeeded in smoothing out otherwise rough edges created through algebraic grid generation. At the same time, the algebraic grid provided a suitable starting point for the generation of the elliptic grid.

EXPERIMENT 5
FLOW OVER AN AEROFOIL

Aim: To simulate flow over NACA 0012 airfoil and find pressure velocity distribution over airfoil.

Problem description: Consider air flowing over NACA 0012 airfoil. The free stream velocity is 90 m/s

Assume standard sea-level values for the free stream

properties: Pressure = 101,325 Pa

Density = 1.2250 kg/m^3

Temperature = 288.16 K

Kinematic viscosity v = 1.4607e-5 m^2/s

Software: ICEM and CFX

Steps Involved In ICEM

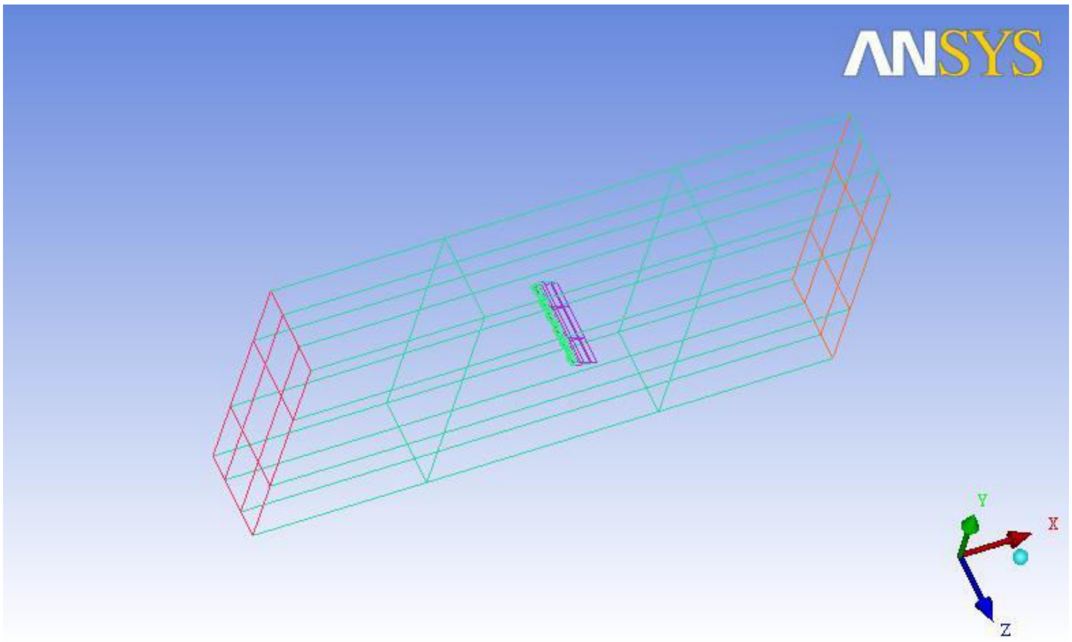
1) **Creation of Geometry in ICEM CFD:**

- Importing the Aerofoil coordinates
  File→Import Geometry→Formatted point data→Select the file of aerofoil coordinates which is in DAT format→ok. Now the coordinates will be displayed.
- Geometry→Create/modify curve→From points→Select above points and leave last 2 points→middle click
- Similarly on bottom side
- Join the end points of the curves
- Create a point in z direction using Geometry→create point→Explicit coordinate→x=0,y=0,z=1→Apply.
- Join points (0,0,0) and (0,0,1) using create/modify curves→from points.
- Create/Modify surfaces→curven driven→driving select curve along z axis →driven curves select airfoil→Apply

2) **Creation of Domain:**

- Geometry→Create/Modify surface →Satndard shapes→Box→X Y Z = 7 2 1→Origin=-3 -1 0→Apply
- Geometry→Create/Modify surface →Segment/Trim surface →Select surface →side walls of box →Select curves → airfoil curves →Apply
- Create Body→Part:body→Material point→Location→centroid of 2 points→2 screen location→select 2 points in box such that its center will be above or below

airfoil→Apply



3) Creation of parts:

- Parts in the tree→Right click→Create part
  Part Suction→ Select entity airfoil upper surface
  Part: Pressure→ Select entity airfoil lower surface
  Part: TE→ Select entity airfoil rear surface
  Part: Inlet→ Select entity box front face
  Part: Outlet→Select entity box back face
  Part: Walls→ Select entity remaining box walls

4) Generation of Mesh:

- Mesh→Global Mesh setup→Max Element size:1→Curvature?Proximity Based Refinement →Enabled→Min size: 0.01→Apply.
- Part Mesh Setup

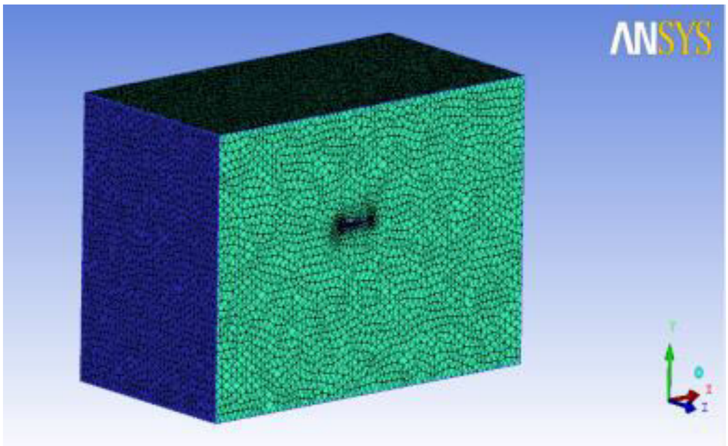
  Inlet→Maximum size : 0.2
  Outlet→Maximum size: 0.2
  Walls→Maximum size: 0.2

Suction and Pressure→Maximum size:0.05

TE→Maximum size:0.01

- Compute Mesh→Volume mesh→Mesh type→Tetra/Mixed→Mesh method→Robust octree→compute.

Now the required mesh has been generated as shown in below fig.



5) Writing output file:

- File→Save Project as→Give the name.
- Output→Select solver→Output solver as: CFX→save→output

type:ASCII→Sacle factor:1→Done.

6) Steps Involved in CFX:

- Select CFX Pre from ANSYS Menu
- File→New case→General→ok

Importing the mesh file:

- Outline→Right click Mesh→Import Mesh→ ICEM

CFD→Select file →open.

Now the mesh has imported into the CFX solver.

Problem setup:

- Analysis Type→Steady state
- Default Domain→Fluid Material→Air at 25^0c
- Fluid Models→isothermal ok
- Boundary → Name:inlet→ boundary type:Select inlet→Subsonic

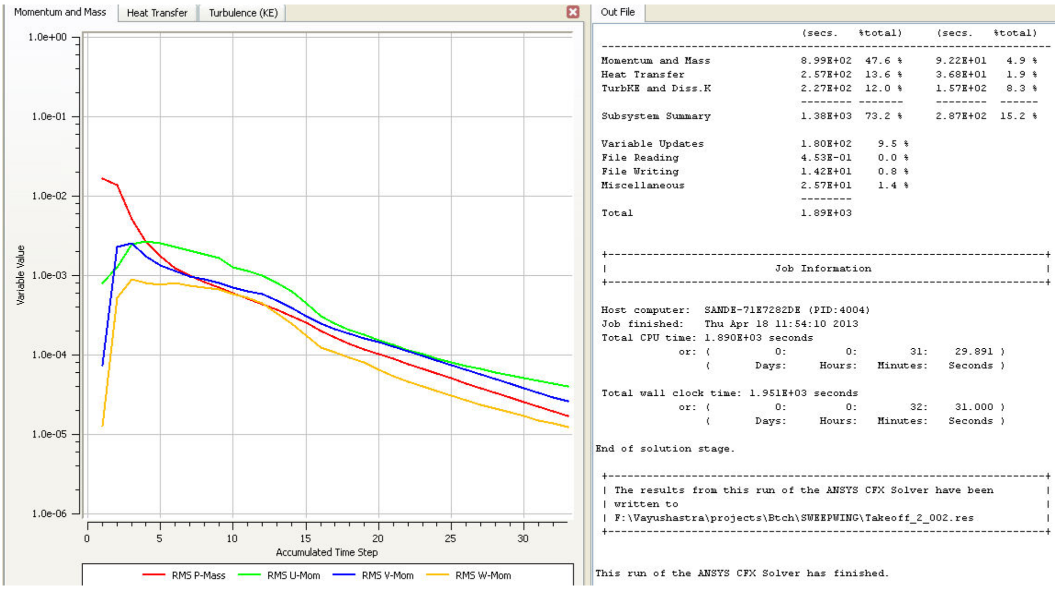
normal velocity : 90 m/s.

- Boundary → Name: → boundary type:Select outlet→Edit→Relative pressure as: 0 Pa
- Boundary→Name:Wall→ boundary type: wall→location: W a l l s →boundary

details→mass momentum →free slip →ok.

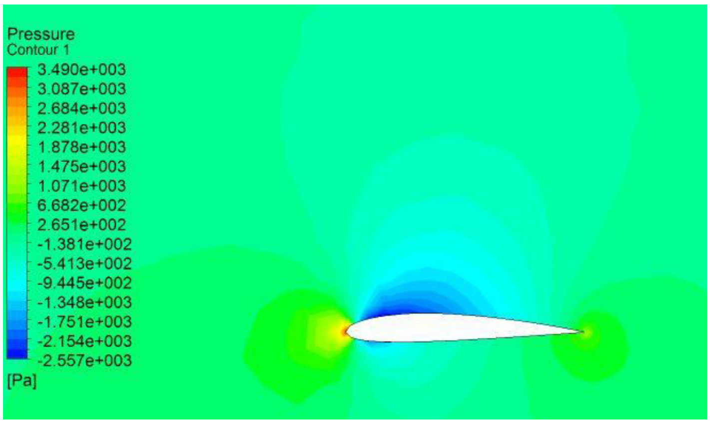
7) Solver:

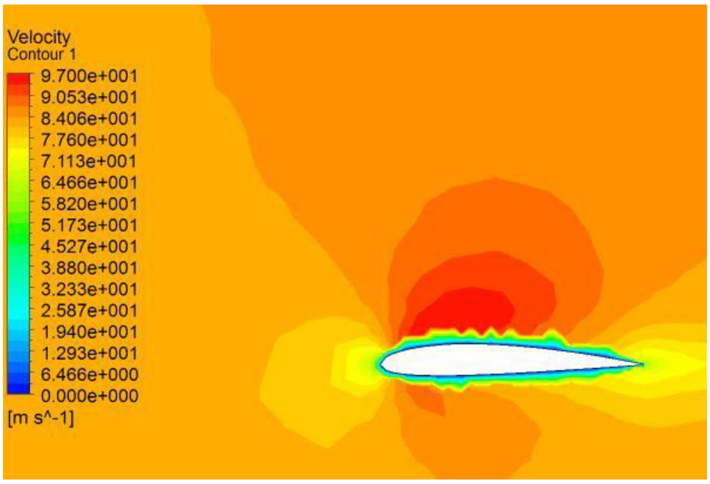
- ➢ Select solver control
- ➢ Advection scheme: high Resolution
- ➢ Convergence control Max iterations:100
- ➢ Convergence criteria :RMS Taret:1E-4
- ➢ Tools →Solver → start solver →Define Run→name file
- ➢ Once solver reaches convergence CFD POST opens.



8) Results:

- ➢ Select →Location →Plane →XY Plane → Apply
- ➢ Contours →Location → plane→Variable→ Velocityand pressure →Apply





Exercise Problems:

5.1 Evaluate aerodynamic characteristics of aerofoil at 15 degrees angle of attack.

5.2 Perform the grid independence study over airfoil.

EXPERIMENT 6

SUPERSONIC FLOW OVER A WEDGE

Aim: To find changes in flow properties across shock when supersonic flow passes over wedge.

Description: Consider air flowing over wedge. The free stream Mach number is 3 and the angle of attack is 5°. Assumestandard sea-level values for the free stream properties:

Pressure =101,325Pa

Density =1.2250kg/m^3

Temperature =288.16K

Kinematic viscosity v = 1.4607e-5 m^2/s

Software: ICEM and CFX

Steps Involved In ICEM CFD:

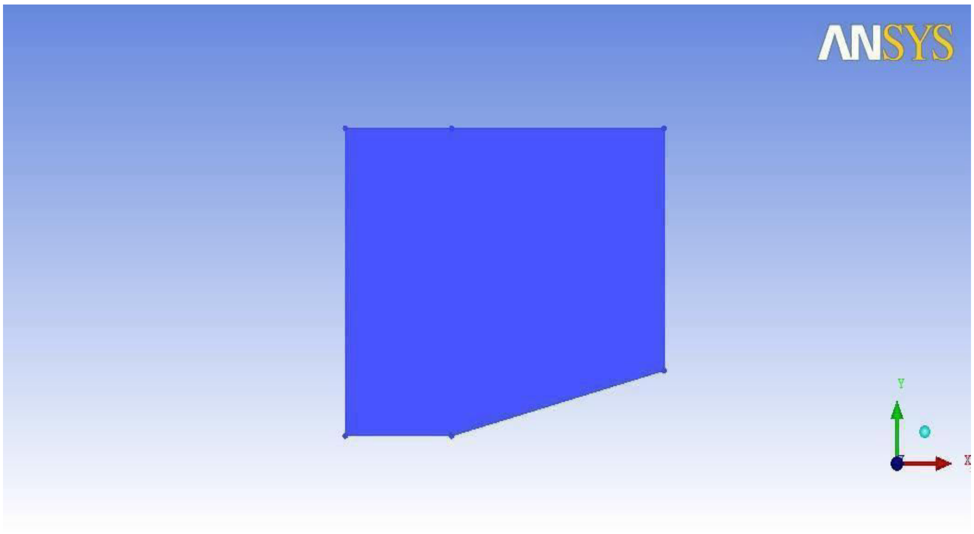
1) Creation of Geometry in ICEM CFD:

- Geometry→Create point→Explicit coordinates→Enter the coordinates as given in table shown:

X	0	0	0.5	1.5	1.5	0.5
Y	0	1.259	1.259	1.259	0268	0
Z	0	0	0	0	0	0

Geometry→Create/modify curve→From points→Select any 2 points→ok→Similarly create the curves to all points

- Geometry→Create/Modify surface→Simple surface→Select all the lines of domain→ok
- Create a point in z direction using Geometry→create point→Explicit coordinate→x=0,y=0,z=1→Apply.
- Join points (0,0,0) and (0,0,1) using create/modify curves→from points.
- Create/Modify surfaces→curven driven→driving select curve along z axis →driven curves select airfoil→Apply



2) Creation of parts:

➢ Parts in the tree→Right click→Create part→Select Left surfae: **Inlet**

Select Right surface: **Outlet**

Select Top surface: **Top**

Select inclined surface: **Wedge**

Select bottom surface: **Front_wedge**

3) Saving the Geometry:

➢ File→Change working directory→Choose the folder

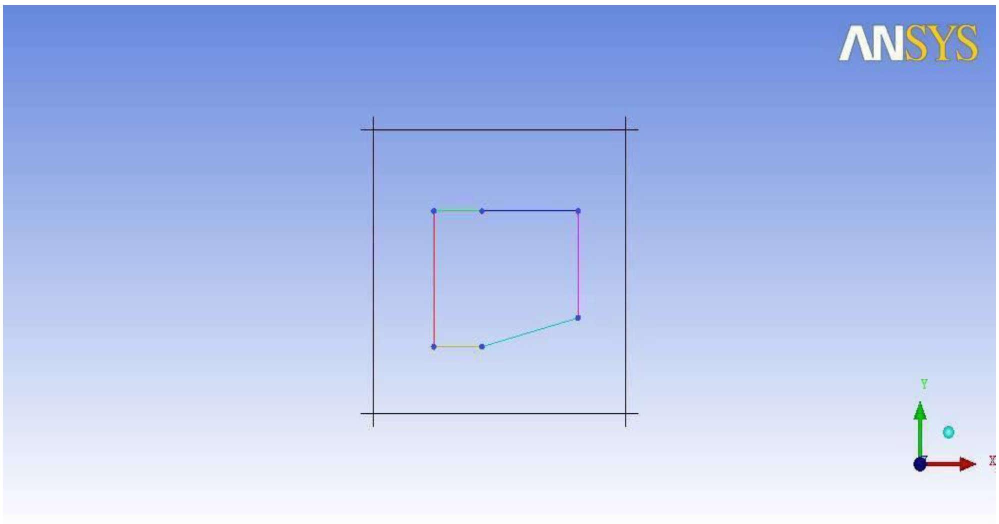
➢ File→Geometry→Save Geometry as→Give the name.

4) Creation of Blocking and Association:

➢ Blocking→Create block→Initialize blocks→Type as:3D Bounding Box→ok

➢ Associate→Associate vertex to point→Select a vertex and a point→Apply→ Similarly associate remaining 3 vertices to points

➢ Associate→Associate edge to curve→Select a edge and a curve →Apply→ Similarly associate remaining all edges to all curves

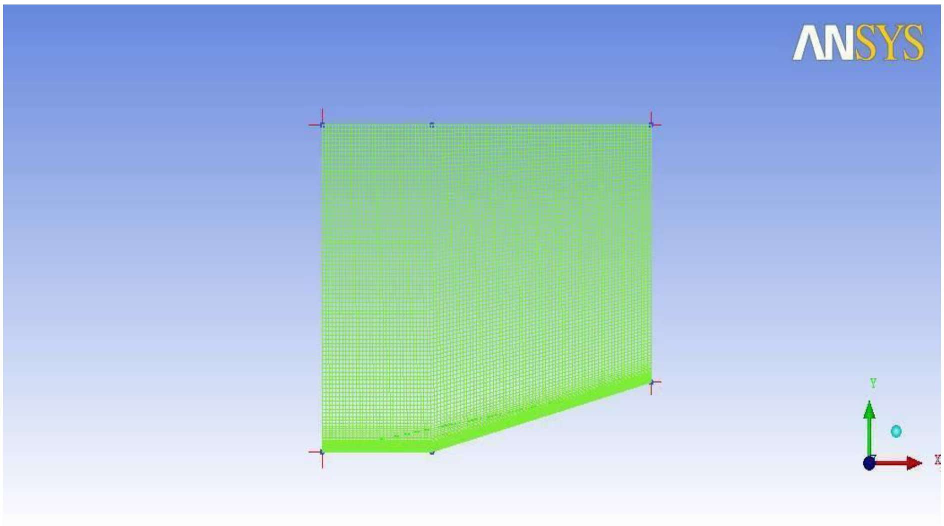


5) Generation of Mesh:

- Pre-mesh parameters→Edge parameters→Switch ON the Copy Parameters→ Select the Horizontal edge and give no. of nodes as: 100→ok
- Pre-mesh parameters→Edge parameters→Switch ON the Copy Parameters→ Select the Vertical edge and give no. of nodes as: 100→Spacing as: 0.001→Ratio as: 1.1→ok

- Switch ON Pre-mesh in the tree→click yes to compute the meshing
- Pre-mesh→Right click→Convert to unstructured mesh

Now the required mesh has been generated as shown in below fig:



6) Saving the Project:

- File→Save Project as→Give the name

.

7) CFX:

- CFX Pre – New file – general.
- Mesh - Import mesh – ICEM CFD – open .msh file.
- Default Domain Basic Settings Materials Air at 25^{0}c, Reference Pressure 0 atm

Domain - Default Domain Modified	
Type	Fluid
Location	FLUID
Materials	
Air Ideal Gas	
Fluid Definition	Material Library
Morphology	Continuous Fluid

Settings	
Buoyancy Model	Non Buoyant
Domain Motion	Stationary
Reference Pressure	1.0000e+00 [atm]
Heat Transfer Model	Isothermal
Fluid Temperature	2.5000e+01 [C]
Turbulence Model	K- epsilon
Turbulent Wall Functions	Scalable

Domain	**Boundaries**	
Default Domain Modified	**Boundary - inlet**	
	Type	INLET
	Location	IN_1
	Settings	
	Flow Regime	Supersonic
	Mass And Momentum	Normal Speed
	Normal Speed	1.04000e+02 [m s^-1]
	Rel Static Pressure	1 atm
	Static temp	300 K
	Boundary - outlet	
	Type	OUTLET
	Location	OUT_1
	Settings	
	Flow Regime	Supersonic
	Mass And Momentum	Average Static Pressure
	Boundary - WALL	
	Type	WALL

	Location	Top_1, Bot_1, Wed_1
	Settings	
	Mass And Momentum	No Slip Wall
	Wall Roughness	Smooth Wall

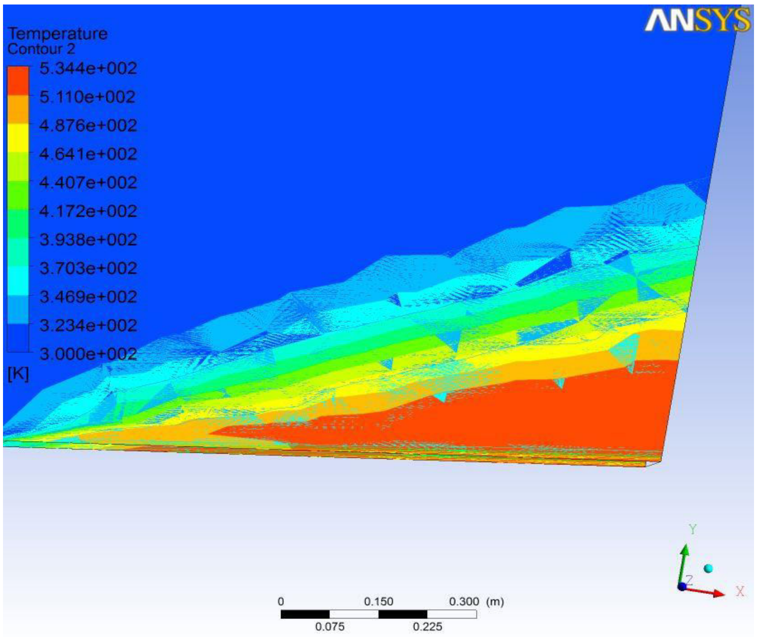
- Click ok and define run to start solution.

POST PROCESSING

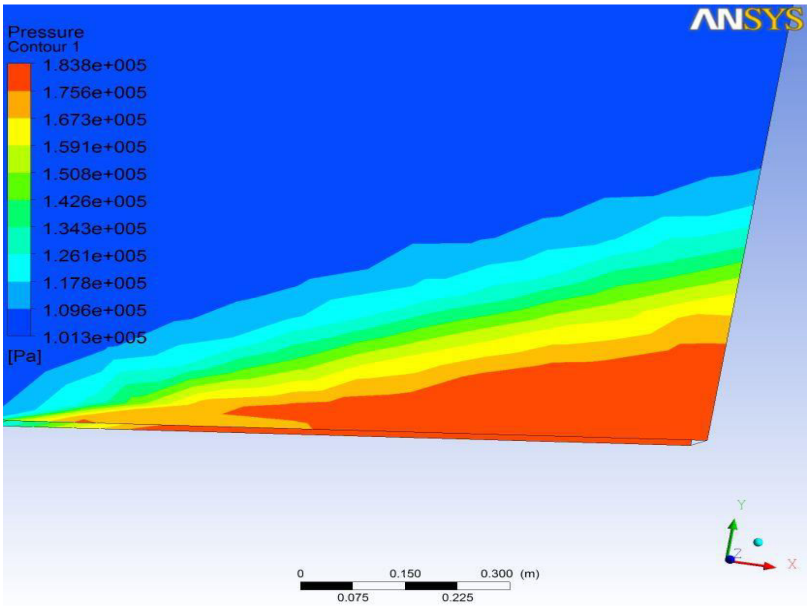
8) Analyze results in CFX POST:

- Create plane
- Create contour – pressure, mach number
- Create streamlines
- Create chart for velocity along plate.

RESULTS:



Temperature contour



Pressure contour

Exercise Problems:

5.1 Evaluate flow properties across oblique shock of wedge at 15 degrees angle of attack.

5.2 Compare the results with analytical formulas.

EXPERIMENT – 7

FLOW OVER A FLAT PLATE

Aim: To study the characteristics of flow over a flat plate

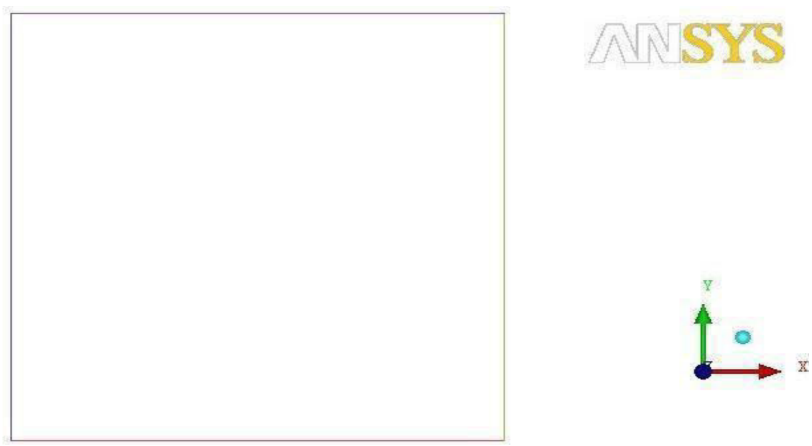
Description: Consider a plate of 1m and the flow of air is 0.00133 m/s. The plate is an stationary solid wall having no slip as its boundary condition.

Software: ICEM and CFX

Procedure:

1) **Steps in ICEM CFD**

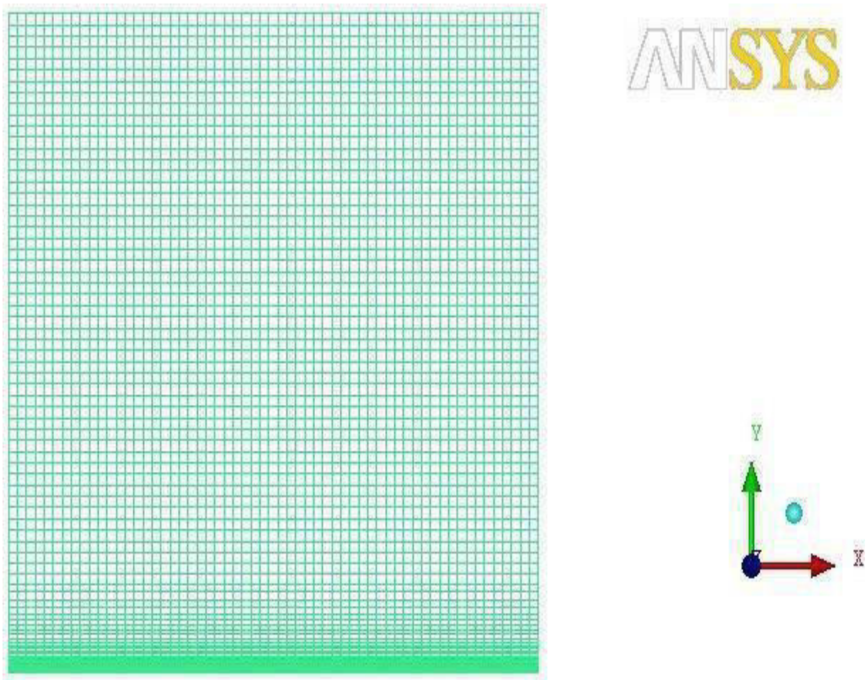
- Geometry→ create point→ explicit coordinates→ 1(0,0,0), 2(1,0,0), 3(1,1,0) and 4(0,1,0) → ok
- Create/modify curve→ select 2 points→ middle click Select all points to make a rectangle.
- Create a point in z direction using Geometry→create point→Explicit coordinate→x=0,y=0,z=1→Apply.
- Join points (0,0,0) and (0,0,1) using create/modify curves→from points.
- Create/Modify surfaces→curven driven→driving select curve along z axis →driven curves select airfoil→Apply



- Create/modify surface→ select all the lines→ surface is created
- Create part→ name inlet→ select the left edge→ middle click Similarly create outlet, top and bottom
- Switch off points and curves→ create part→ name surf→ click on surface→ ok Blocking→ create block→ select entities→ click spectacles→ middle click→ switch on points and curves
- Go to association→ associate vertex→ select the point→ double click on the point

Associate→ edge to curve→ select the edge→ ok→ again select the edge→ ok
Similarly for the remaining edges

- Premesh parameters→ edge parameters→ select any edge→ click on copy
- parameters→ nodes-60, spacing-0.01, ratio-1.1→ ok
- Blocking tree→ premesh→ right click→ convert structured to unstructured mesh



- Change the working directory
- output→ output solver CFX→ common-ansys→ ok

2) Steps in CFX:

- CFX Pre – New file – general.
- Mesh - Import mesh – ICEM CFD – open .msh file.
- Default Domain Basic Settings Materials Air at 25^0c, Reference Pressure 1 atm

Domain - Default Domain Modified	
Type	Fluid
Location	FLUID
Materials	
Air Ideal Gas	
Fluid Definition	Material Library
Morphology	Continuous Fluid

Settings		
Buoyancy Model	Non Buoyant	
Domain Motion	Stationary	
Reference Pressure	1.0000e+00 [atm]	
Heat Transfer Model	Isothermal	
Fluid Temperature	2.5000e+01 [C]	
Turbulence Model	SST	
Turbulent Wall Functions	Scalable	
Domain	**Boundaries**	
Default Domain Modified	**Boundary - inlet**	
	Type	INLET
	Location	IN_1
	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Normal Speed
	Normal Speed	0.3000e+02 [m s^-1]
	Turbulence	low Intensity = 1%
	Boundary - outlet	
	Type	OUTLET
	Location	OUT_1
	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Average Static Pressure
	Pressure Profile Blend	5.0000e-02
	Relative Pressure	1.0132e+05 [Pa]
	Pressure Averaging	Average Over Whole Outlet
	Boundary - WALL	
	Type	WALL
	Location	flat_ns_1, back_ns_1 and front _ ns_1
	Settings	

	Mass And Momentum	No Slip Wall
	Wall Roughness	Smooth Wall

- Click ok and define run to start solution.

3) Analyze results in CFX POST:

- Create plane
- Create contour – pressure, mach number
- Create streamlines
- Create chart for velocity along plate.

RESULTS:

EXERCISE PROBLEMS

7.1 Find out the effect of viscosity of water on flat plate.
7.2 Find the material of the plate on velocity profile.

EXPERIMENT 8
LAMINAR FLOW THROUGH PIPE

Aim: To study characteristics of laminar flow through a pipe.

Description: Consider a pipe of radius 0.05 and 1 mt length. The freestream velocity considered is 40m/s.

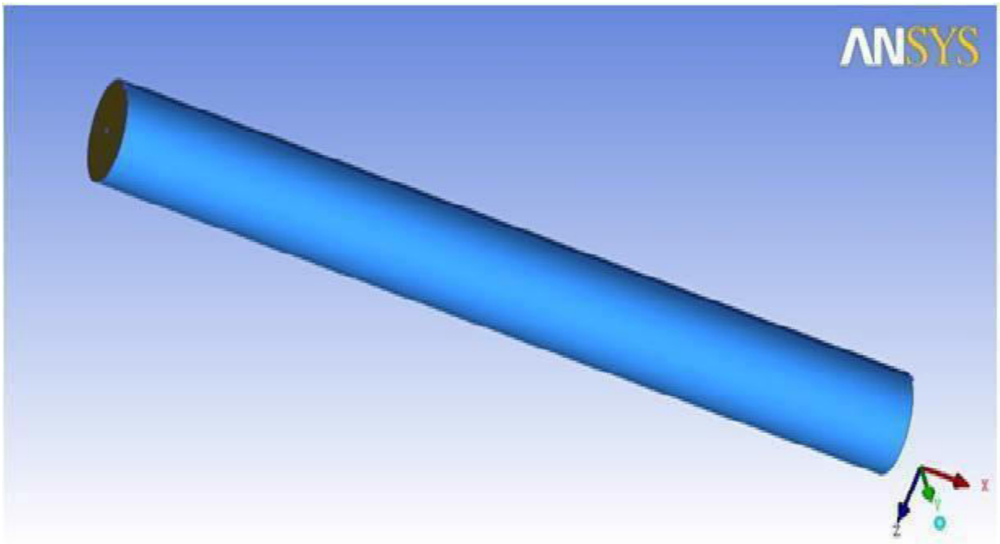
Software: ICEM and CFX

STEPS INVOLVED:

1) **Create A Geometry:**

a) **Create a point:** Geometry →create point →explicit coordinates→(X, Y, Z) = (0, 0, 0) →apply→(X, Y, Z) = (1, 0, 0) →ok.

b) **Create a pipe:** Geometry→Create/modify surfaces→Standard shapes→ cylinder→radius1=radius2 = 0.05→select the 2 points →ok



2) **Generation of parts:**

- Part →create part→inlet→select inlet→ok.
- Part →create part →outlet →select outlet →ok.
- Part →create part →pipe →select pipe without inlet and outlet →ok.

3) **Generation of blocking:**

- Blocking →create block →solid →select pipe element with inlet and outlet→ apply→ok.
- Blocking →associate →edge to curve →select the 4 edges of the

blocking at inlet →apply →select the inlet curve→ok.

- Blocking→associate →edge to curve→select the 4 edges of the blocking at outlet → apply→select the outlet curve→ok.

- Associate→faces to surface→select inlet face→ apply →select as inlet→accept →ok.

- Associate →faces to surface →select outlet face →apply →select as outlet →accept →ok.

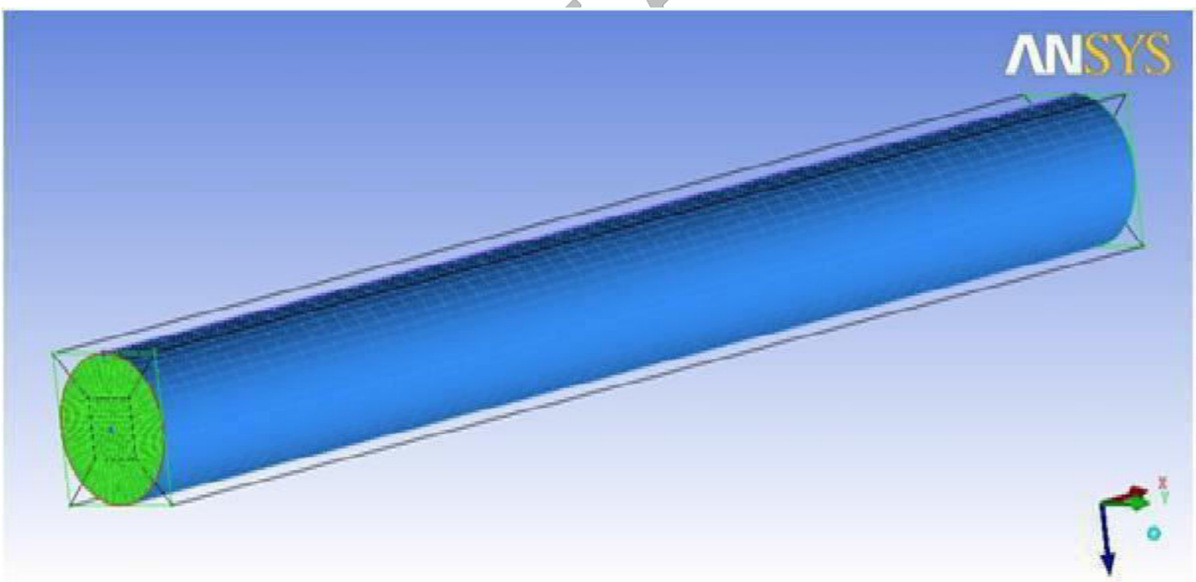
- Associate →face to surface →select pipe faces →apply →select as pipe → accept →
- Blocking →split block →O grid block →select the 2 faces (inlet & outlet) →apply →ok.

4) Generation of Meshing:

- Blocking →pre-mesh parameters →edge parameters →switch on the copy parameters →select 1 edge →give no. of nodes =20→ok.

- Repeat the above steps to the remaining edges also and then apply.

- Blocking →pre-mesh →compute.



- Blocking →pre-mesh →convert to unstructured mesh.

5) Generation of Solver file :

- Output→select solver→ansys.cfx→ANSYS→ok.

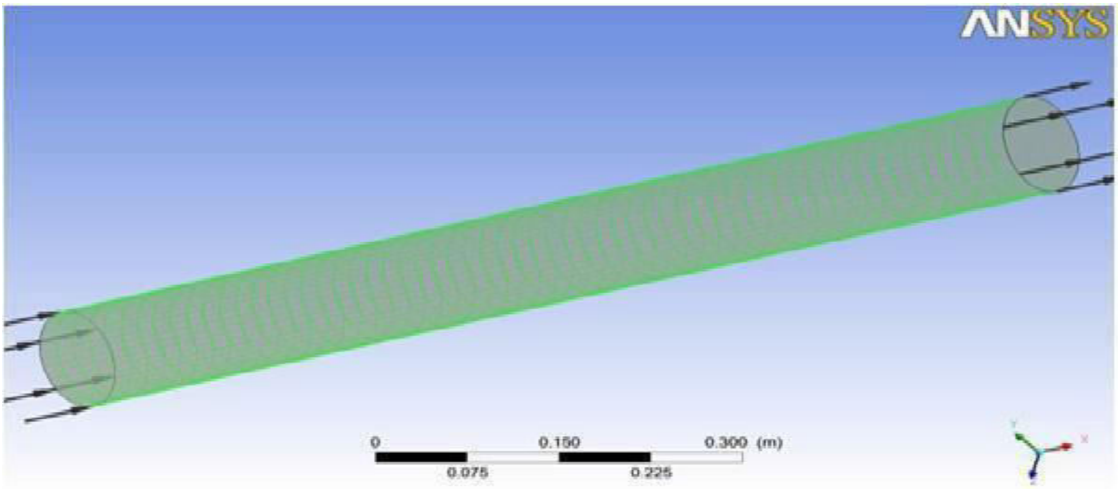
- Output →write input →done →check the file is saved folder →ok.

6) Solution in CFX Solver:

- Start →programs →ANSYS →fluid dynamics →CFX →ok.

7) CFX-PRE:

- File→new case→general→ok.
- File→import→mesh→select the meshed filed→ok.
- Boundary→Boundary1→Boundary type as: inlet→location as: inlet→boundary conditions as: velocity=40m/s→ok.
- Boundary→Boundary2→boundary type as: outlet→location as: outlet→boundary conditions as: static pressure=0→ok.
- Boundary→Boundary3→boundary type as: wall→location as: pipe→boundary conditions as: no slip condition, smooth wall→ok.
- Domain→basic settings→location as: solid→domain type as: fluid domain→material as: air→ok.
- Solver control→basic settings→ max.itterationsas:1000→residual target as: 0.000000001→ok.
- Write solver input file→give the name of the file→ok.

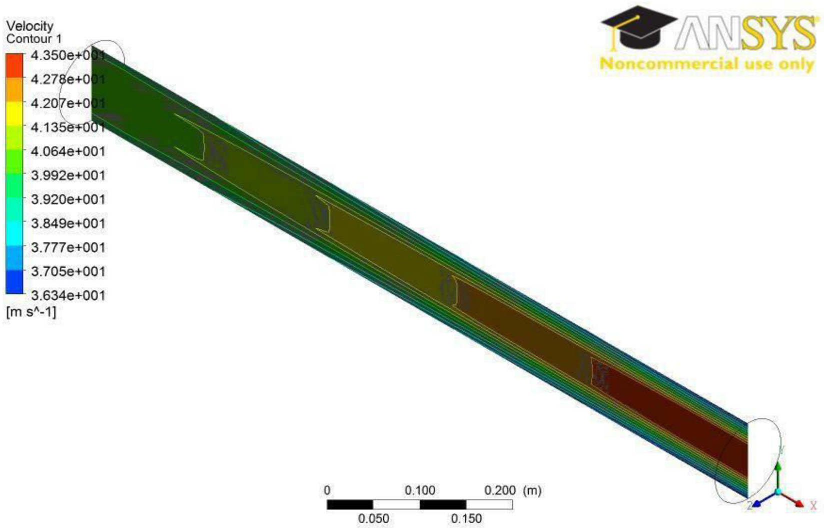


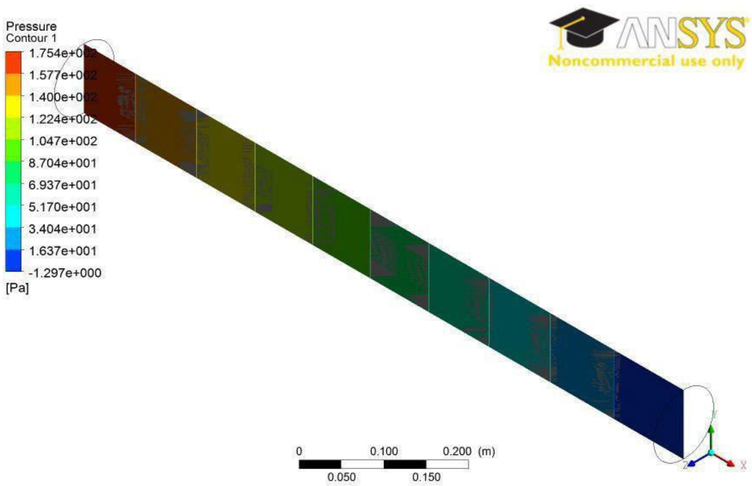
8) CFX-Solver Manager:

- File→Define run→Solver input file→Select the file→Start run.

9) CFD POST

- Select →Location →Plane →XY Plane → Apply
- Contours →Location → plane→Variable→ Velocityand pressure →Apply.





Exercise problems

8.1 Find out the effect convergent section on velocity.

8.2 Find the minor losses in a bend pipe.

EXPERIMENT 9

FLOW OVER A CYLINDER

Aim: To obtain flow over a cylinder of 10cm radius and 1m length consider the flow with velocity 34m/s.

Software: ICEM and CFX

Procedure: **Pre processing:**

1. **Create geometry in ICEM**

- Create the points 1(0,0,0); 2(0,10,0); 3(10,0,0); by selecting geomtry tab → Create point → Explicit coordinate method.
- Geometry → Create curve → circle on arc → select the center as point 1 → select other two points as 2 and 3 → Apply.
- Create the point 4(0,0,100); by selecting geomtry tab → Create point → Explicit coordinate method.
- Join the points 1-4 by selecting geomtry tab → Create/modify curve from points
- Geometry → Create surface → Curve driven → select driving curve 1-4 and then the circle lines under the driven curve → Apply
- Geometry → Create surface → Simple surface → select lines → apply.
- Create domain over cylinder to analyze flow field.
- Delete unwanted surfaces, lines and points.
- From the model tree → right click on parts → create part → Name
    - Inlet – in_1
    - Outlet – out_1
    - wall – wall_1
    - Cylinder- cyl_1
    - Apply
- Geometry → Create body → Location → centroid of 2 points → select the two diagonally opposite points on the model → Apply.

2. **Mesh geometry:**

- Click Mesh Function tab
- Give global mesh parameters in global mesh setup like element scale factor and element max size = 10→ display on → apply.
- Create partmesh setup for inlet, outlet and wall element max size = 10 → height 0.5 → height ratio 1.2, and cylinder element max size = 1.
- Compute mesh → volume mesh → mesh type → tetra/mixed → create prism layers → create hexa-core → mesh method → Robust[octree] → select geometry – compute.
- Edit mesh – check mesh – quality mesh – smooth mesh.

3. **Export mesh**
   - Output – output to cfx – save project – output type – output scale factor - .msh file created.
4. **CFX:**
   - CFX Pre – New file – general.
   - Mesh - Import mesh – ICEM CFD – open .msh file.
   - Default Domain Basic Settings Materials Air at 25^0c, Reference Pressure 0 atm

<table>
<tr><th colspan="2">Domain - Default Domain Modified</th><td></td></tr>
<tr><td>Type</td><td>Fluid</td><td></td></tr>
<tr><td>Location</td><td>FLUID</td><td></td></tr>
<tr><td colspan="2">Materials</td><td></td></tr>
<tr><td colspan="2">Air Ideal Gas</td><td></td></tr>
<tr><td>Fluid Definition</td><td>Material Library</td><td></td></tr>
<tr><td>Morphology</td><td>Continuous Fluid</td><td></td></tr>
<tr><td colspan="2">Settings</td><td></td></tr>
<tr><td>Buoyancy Model</td><td>Non Buoyant</td><td></td></tr>
<tr><td>Domain Motion</td><td>Stationary</td><td></td></tr>
<tr><td>Reference Pressure</td><td>1.0000e+00 [atm]</td><td></td></tr>
<tr><td>Heat Transfer Model</td><td>Isothermal</td><td></td></tr>
<tr><td>Fluid Temperature</td><td>Ideal gas</td><td></td></tr>
<tr><td>Turbulence Model</td><td>K- epsilon</td><td></td></tr>
<tr><td>Turbulent Wall Functions</td><td>Scalable</td><td></td></tr>
<tr><th>Domain</th><th colspan="2">Boundaries</th></tr>
<tr><td rowspan="10">Default Domain Modified</td><th colspan="2">Boundary - inlet</th></tr>
<tr><td>Type</td><td>INLET</td></tr>
<tr><td>Location</td><td>IN_1</td></tr>
<tr><td colspan="2">Settings</td></tr>
<tr><td>Flow Regime</td><td>Supersonic</td></tr>
<tr><td>Mass And Momentum</td><td>Normal Speed</td></tr>
<tr><td>Normal Speed</td><td>34.01000 [m s^-1]</td></tr>
<tr><td>Rel Static Pressure</td><td>1 atm</td></tr>
<tr><td>Static temp</td><td>300 K</td></tr>
<tr><th colspan="2">Boundary - outlet</th></tr>
</table>

	Type	OUTLET
	Location	OUT_1
	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Average Static Pressure
	Boundary - WALL	
	Type	WALL
	Location	Wall_1
	Settings	
	Mass And Momentum	free Wall
	Wall Roughness	Smooth Wall

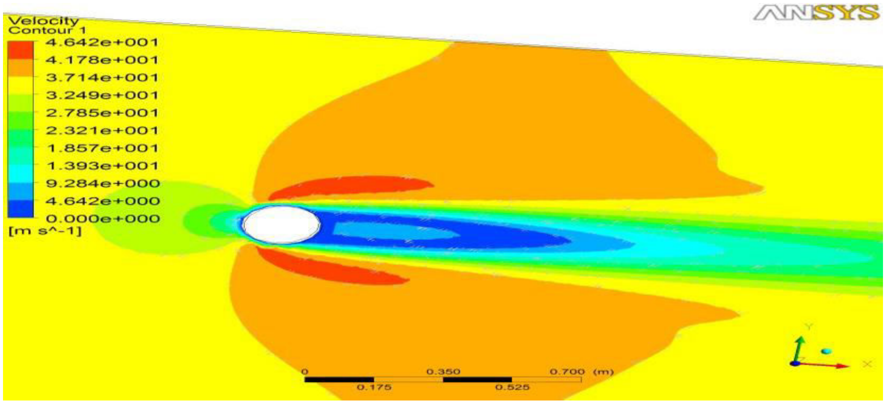
Boundary – Cyl_1	
Type	WALL
Location	Wall_1
Settings	
Mass And Momentum	No slipWall
Wall Roughness	Smooth Wall

- Click ok and define run to start solution.

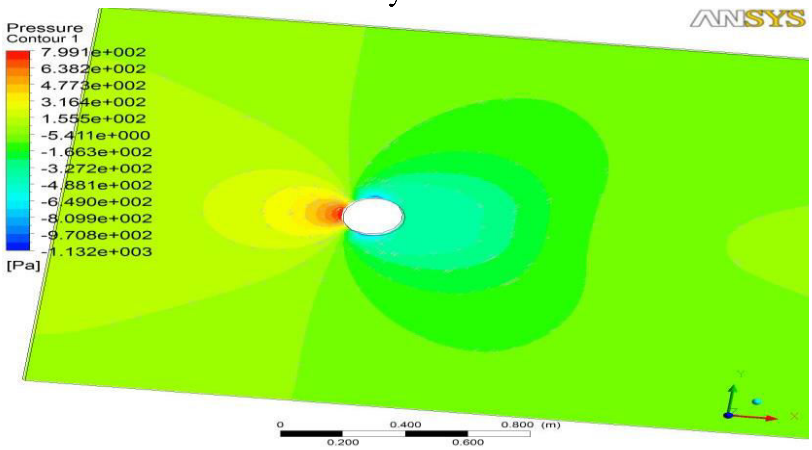
POST PROCESSING

5. **Analyze results in CFX POST:**
   - Create plane
   - Create contour – pressure, mach number
   - Create streamlines
   - Create chart for velocity along pipe.

RESULTS:



velocity contour



pressure contour

EXPERIMENT 10

SIMULATION OF COMPRESSIBLE FLOW IN A CONVERGENT-DIVERGENT NOZZLE

Aim: Consider the nozzle having a cross sectional area A varies with axial distance from the throat, according to the formula $A = 0.1+X^2$; where X varies from $-0.5<X<0.5$. Stagnation pressure P_o = 101325 pa; stagnation temperature T_o = 300K;

Software: ICEM and CFX

Procedure:

Pre processing:

1. **Create geometry in ICEM**
   - Create the vertex data of Nozzle contour variation along axis line with vertices (-0.5, 0) and (0.5, 0) and $A = 0.1+X^2$ in excel sheet.
   - Import vertex data into ICEM by creating .dat file.
   - Create edges by create/modify curves and face by create/modify faces of nozzle using geometry function tab.
   - Create body point.
   - Create parts inlet, outlet and wall.

2. **Mesh geometry:**
   - Click Mesh Function tab
   - Give global mesh parameters in global mesh setup like element scale factor and element max size.
   - Create partmesh setup for inlet, outlet and wall.
   - Compute mesh – volume mesh – mesh type – select geometry – compute.
   - Edit mesh – check mesh – quality mesh – smooth mesh.

3. **Export mesh**
   - ➢ Output – output to cfx – save project – output type – output scale factor - .msh file created.

4. **CFX:**
   - ➢ CFX Pre – New file – general.
   - ➢ Mesh - Import mesh – ICEM CFD – open .msh file.

Domain - Default Domain Modified	
Type	Fluid
Location	FLUID
Materials	
Air Ideal Gas	
Fluid Definition	Material Library
Morphology	Continuous Fluid
Settings	
Buoyancy Model	Non Buoyant
Domain Motion	Stationary
Reference Pressure	1.0000e+00 [atm]
Heat Transfer Model	Isothermal
Fluid Temperature	2.5000e+01 [C]
Turbulence Model	k epsilon
Turbulent Wall Functions	Scalable

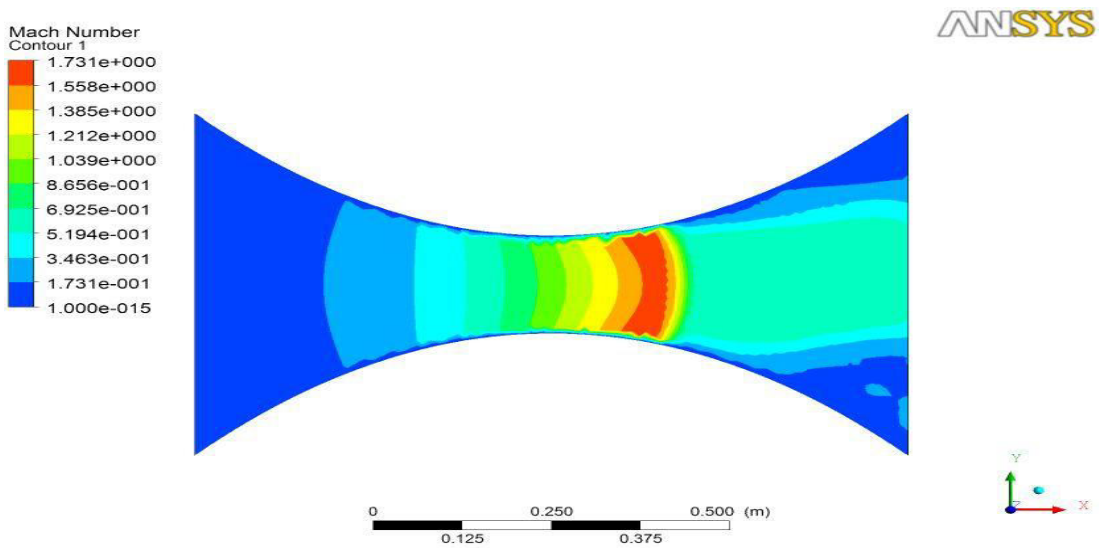
Domain	**Boundaries**	
Default Domain Modified	**Boundary - inlet**	
	Type	INLET
	Location	IN
	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Normal Speed
	Normal Speed	2.8000e+02 [m s^-1]
	Turbulence	Medium Intensity and Eddy Viscosity Ratio
	Boundary - outlet	
	Type	OUTLET
	Location	OUT
	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Average Static Pressure
	Pressure Profile	5.0000e-02

	Blend	
	Relative Pressure	1.0132e+05 [Pa]
	Pressure Averaging	Average Over Whole Outlet
	Boundary - WALL	
	Type	WALL
	Location	Wall
	Settings	
	Mass And Momentum	No Slip Wall
	Wall Roughness	Smooth Wall

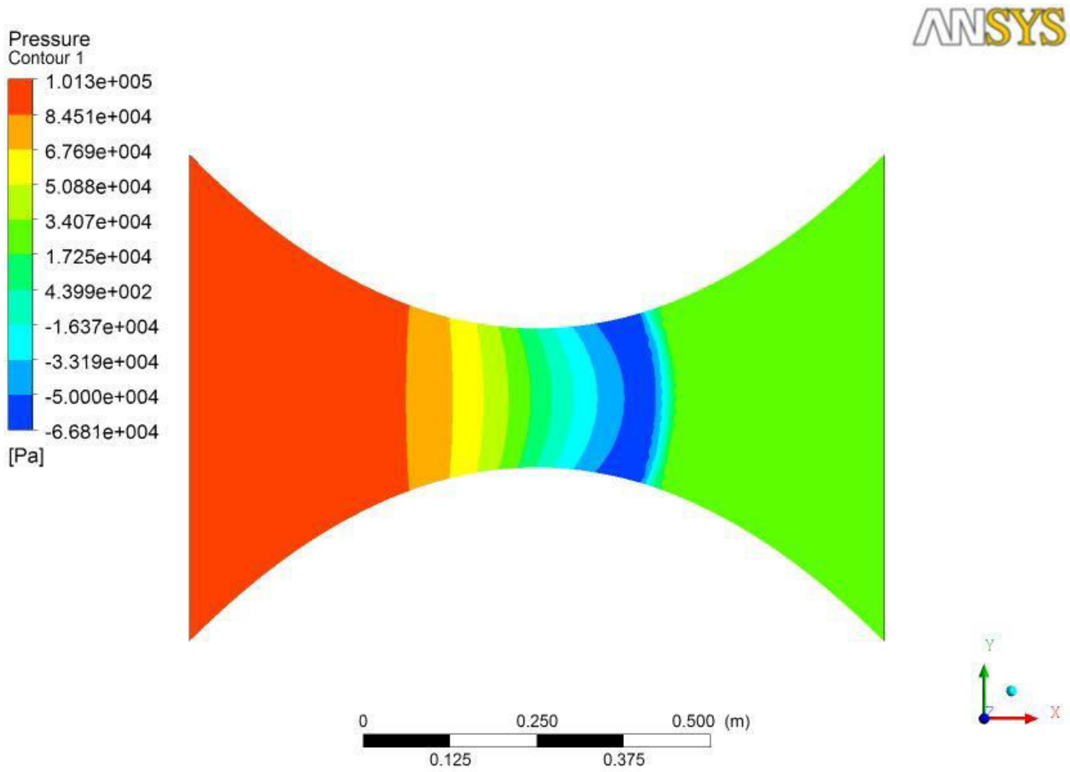
5. **Solve the problem:**
    - Solver Control – Basic setting
    - Advection scheme
    - Turbulence Numerics
    - Convergence controls
    - Fluid timescale control
    - Convergence criteria

POST PROCESSING

6. **Analyze results:**
    - Create plane
    - Create contour – pressure, mach number
    - Create streamlines
    - Create chart for temperature along nozzle axis.



Mach number contour



Pressure contour

EXPERIMENT -11

FLOW OVER A WING

Aim: To obtain flow field over a finite rectangular wing with incoming flow conditions velocity 120 m/s, Pressure 1 atm and air at 25^0c.

Software: ICEM and CFX

Procedure: **Pre processing:**

1. **Create geometry in ICEM**
   - **Import vertex data of airfoil coordinates from File** → Import Geometry → Formatted point data.
   - Join the points by selecting geomtry tab, Create curve from points.
   - Create a line in Z direction to extrude airfoil.
   - Geometry → Create surface → Curve driven → select driving curve and then the remaining lines under the driven curve → Apply.
   - Geometry → Create surface → Simple surface → for upper and lower surface.
   - Create domain for the flow analysis.
   - Delete unwanted surfaces, lines and points.
   - From the model tree → right click on parts → create part → Name

     Inlet – in_1

     Outlet – out_1

     Top – top_1

     Side – sidewall_1

     Bottom – Bot _ 1

     Airfoil – Airfoil_1

     Apply
   - Geometry → Create body → Location → centroid of 2 points → select the two diagonally opposite points on the model such that point should not be within airfoil→ Apply.
2. **Mesh geometry:**
   - Click Mesh Function tab
   - Give global mesh parameters in global mesh setup like element scale factor and element max size = 0.1 → display on → apply.
   - Create partmesh setup for inlet, outlet and wall.
   - Compute mesh → volume mesh → mesh type → tetra/mixed → create prism layers → create hexa-core → mesh method → Robust[octree] → select geometry – compute.
   - Edit mesh – check mesh – quality mesh – smooth mesh.
3. **Export mesh**

- ➢ Output – output to cfx – save project – output type – output scale factor - .msh file created.

4. CFX:

- ➢ CFX Pre – New file – general.
- ➢ Mesh - Import mesh – ICEM CFD – open .msh file.

Domain - Default Domain Modified	
Type	Fluid
Location	FLUID
Materials	
Air Ideal Gas	
Fluid Definition	Material Library
Morphology	Continuous Fluid
Settings	
Buoyancy Model	Non Buoyant
Domain Motion	Stationary
Reference Pressure	1.0000e+00 [atm]
Heat Transfer Model	Isothermal
Fluid Temperature	2.5000e+01 [C]
Turbulence Model	SST
Turbulent Wall Functions	Scalable

Domain	**Boundaries**	
Default Domain Modified	**Boundary - inlet**	
	Type	INLET
	Location	IN_1
	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Normal Speed
	Normal Speed	1.2000e+02 [m s^-1]
	Turbulence	low Intensity = 1%
	Boundary - outlet	
	Type	OUTLET
	Location	OUT_1

	Settings	
	Flow Regime	Subsonic
	Mass And Momentum	Average Static Pressure
	Pressure Profile Blend	5.0000e-02
	Relative Pressure	1.0132e+05 [Pa]
	Pressure Averaging	Average Over Whole Outlet
	Boundary - WALL	
	Type	WALL
	Location	Airfoil_1, Top_1, Bot _1
	Settings	
	Mass And Momentum	No Slip Wall
	Wall Roughness	Smooth Wall

- Click ok and define run to start solution.

POST PROCESSING

5. **Analyze results in CFX POST:**
    - Create plane
    - Create contour – pressure, mach number
    - Create streamlines
    - Create chart for velocity along plate.

RESULTS:

VIVA QUESTIONS:

1. Define CFD?
2. What are the three major steps of CFD?
3. What are the governing equations of CFD?
4. What is meant by Discretization?
5. Which type of Discretization is used in CFD?
6. Difference between forward and backward differencing scheme?
7. What is Explicit method and Implicit method?
8. What is LAX method?
9. What is a stability criterion?
10. What is thermal diffusivity?
11. Define Grid?
12. Difference between Structured and Unstructured grid?
13. What is meant by Grid Independence study?
14. What is linspace command in MATLAB?
15. How to give titles to X and Y axis of a graph?
16. How to create Hybrid mesh in ICEM?
17. How to create structured grid in ICEM?
18. What is the importance of Body point in ICEM?
19. How to define material properties in CFX or FLUENT?
20. What is meant by convergence criteria?
21. How to define supersonic inlet conditions in CFX?
22. What is Grid adaption technique?
23. What is meant by parallel and serial processing?
24. In how many ways CFD results can be presented?
25. How to define formulas in CFD Post?
26. What are the causes for reverse flow or diverged flow during CFD iterations?
27. What are the relaxations factors in FLUENT?
28. What is courant number and how does it affects the solution?
29. What are the different types of turbulence models in CFD?
30. Difference between free slip and no-slip conditions?