King Fahd University of Petroleum and Minerals

Aerospace Engineering Department

Senior Design Project I

(AE 412)

Air Particle Separator

SALMAN AL-FIFI

ID # 991694

ABSTRACT:

Flight operations in desert and sandy environments often encounter sand and dust, which have been known to create a number of operating problems for the power plant. Low altitude aircraft and helicopters are particularly vulnerable to sand and dust entering the engine. Sand ignition causes severe performance degradation, excessive were, increased maintenance and eventually premature failure of the engines.
This project concentrates on investing the aerodynamic performance of Engine particle separators (Figur-1) in preventing such damage and exploring an experimental set up design that can simulate flow of air in this unit.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE page

LIST OF TABLES

ACKNOWLEDGEMENT

"In the name of Allah (God), Most Gracious, Most Merciful. Read, In the name of thy lord and Cherisher, Who created man from a [leech - like] clot. Read, and thy Lord Is Most Bountiful, He Who taught [the use of] the pen. Taught man that which he know not. Nay, but man doth Transgress all bounds. In that he looketh upon himself as self- sufficient. Verily, to thy Lord is the return [of all]. " (The Holy QURAN, Surah No. 96).

Above and first of all, I thank and pray to Allah for His guidance and protection throughout my life. Including the years of this study.

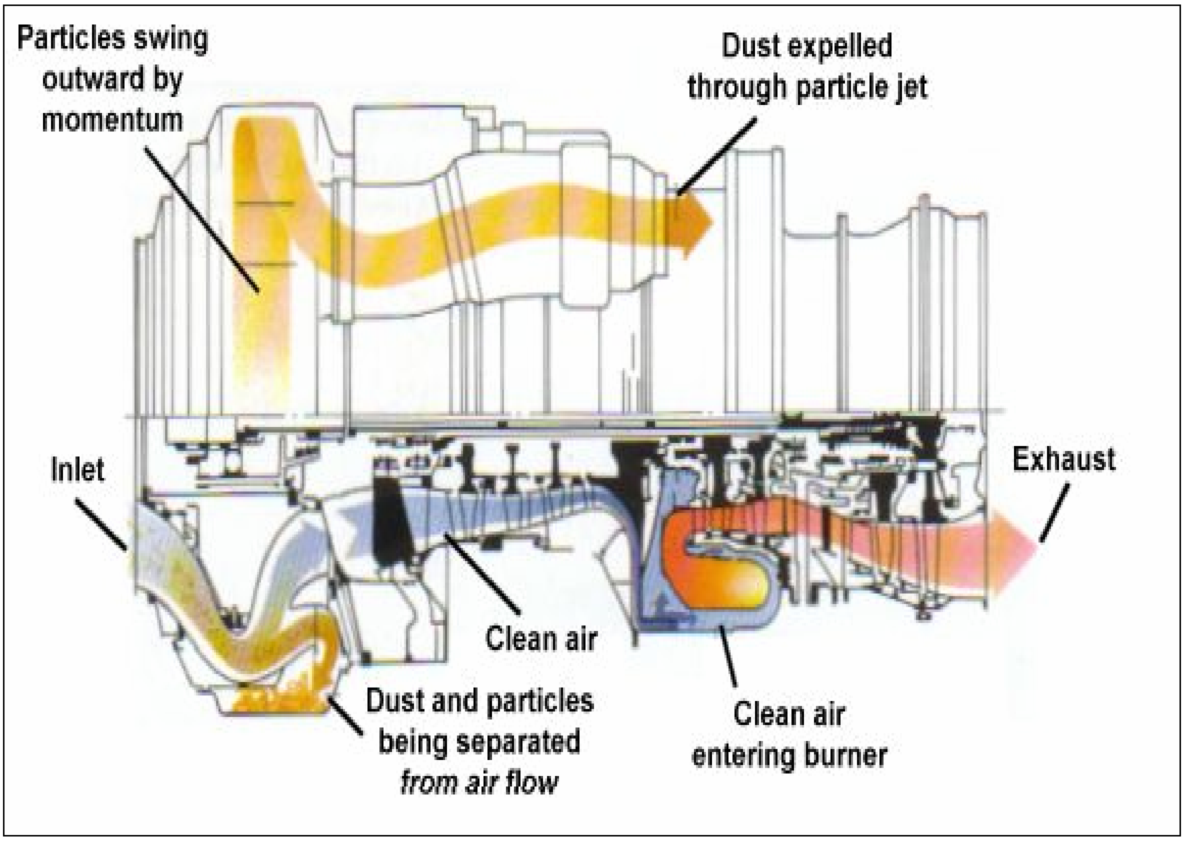
I am happy to have had a chance to glorify His name, in the sincerest way, through his small accomplishment, and I ask Him, with hope in Him, to accept my efforts. And secondary my peace upon His Prophet, MOHAMMED (salla Allah alihe wa sallam).

I wish to thank my direct supervisor **Mr:. Mueyyet Tozan** for providing me with some sources, which deal with the air particle separator, and for his support, comments, suggestions, constructive criticism, and encouragement. I also would like to express my sincere thanks to my supervisor **Proof. Ahmad Z. Al - Garni** for sharing with me his knowledge and helping me through this project.

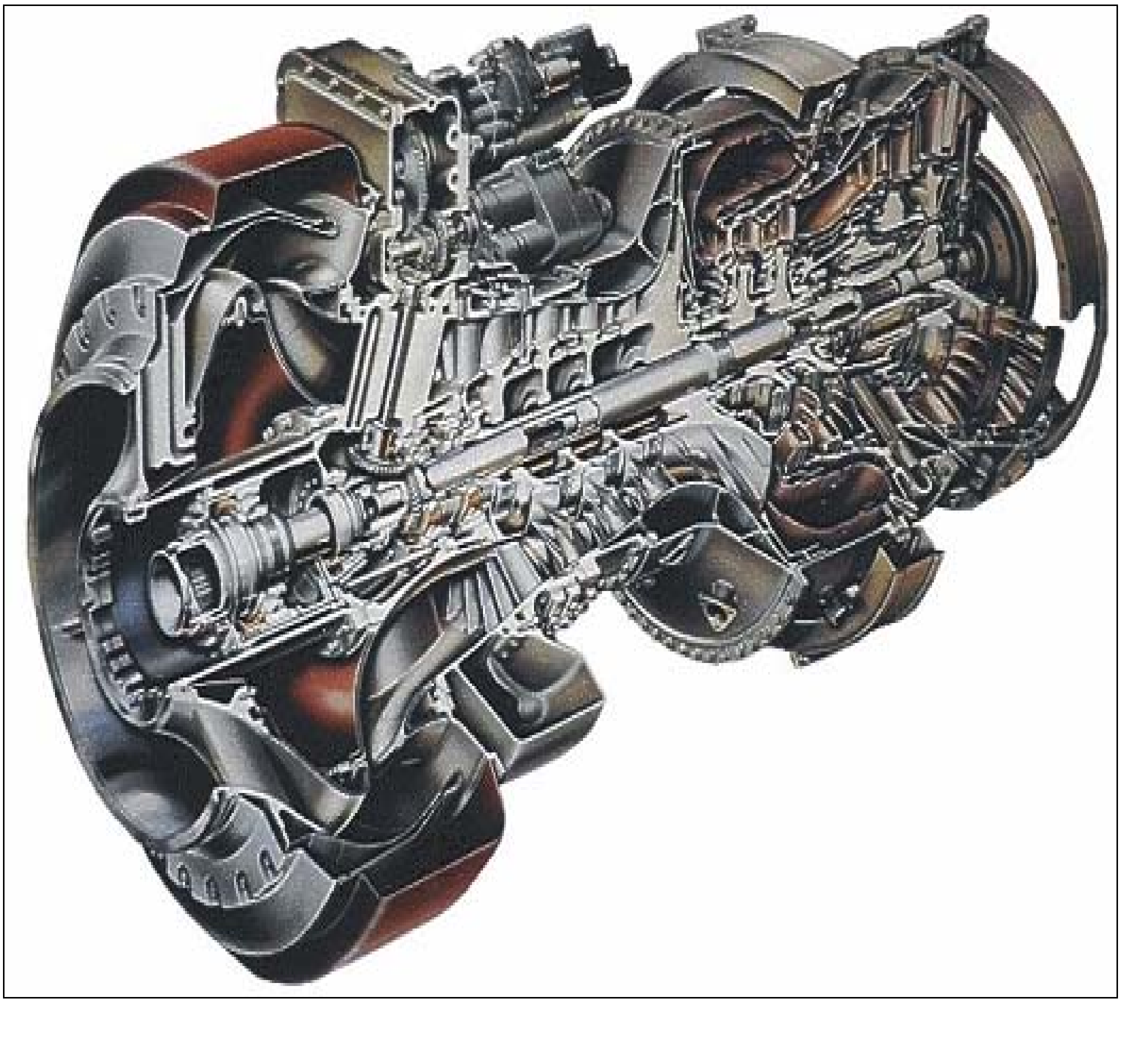
INTRODUCTION:

Flight operations in desert and sandy environments often encounter sand and dust, which have been known to create a number of operating problems for the power plant. Low altitude aircraft and helicopters are particularly vulnerable to sand and dust entering the engine due to the constant take offs and landing, and map of the earth' flight operations which required flying close to the ground.

Ingestion of certain coarse sand into gas turbine engines can cause erosion (Figure 1and Figure 2) of the blades and vanes resulting in a loss of power and serge margin. Axial compressors usually suffer from roughening of the blade surfaces, rolling of blade leading edges and the wearing of blade tips and shroud. Turbine stages can also incur similar problems. In centrifugal compressors, wearing of the impeller leading edges and trailing edge root often result in structural failure. Ingestion of fine dust causes little or no erosion but they tend to cause other problems in the turbines where they melt in the combustion process and deposit on the turbine blades thus reducing the throat area and resulting in a mismatched engine. Erosion of the turbine by large particles can cause related problems but here the throat areas are generally enlarged. The glass like deposits can also shed causing structural damage to engine components.


Particles swing
outward by
momentum
Dust expelled
through particle jet
Inlet
Exhaust
Clean air
Dust and particles
being separated
from air flow
Clean air
entering burner

(Figure 1)



(Figure 2)

NATURE OF PROJECT:

The nature of this project is going to be computational. As a next step, a model of engine particle separator would be manufactured to test its aerodynamic performance. Actually after collecting enough
Actually after collecting enough data, experiments will be carried out and computations will be done over these experiments to achieve the desired objectives of this project.
The teamwork of this project will be consisting of:

1. Mr. Mueyyet Tozan
2. Dr. Ahmed Z. AL-Garni
3. Salman AL-Fifi

OBJECTIVE:

The objectives of this project are to:

- Investigate the effect of dust and sand ingestion into engines.
- Investigate operation of the engine air particle separators.
- Investigate and model the flow field in engine particle separators.
- Develop a test set up witch will be able to simulate the flow of air-sand mixture in the separator that allows further research in the particle trajectories to increase the separating efficiency.
- Looking for conceptual design of engine particle separator with minimum pressure loss.

PROGRESSIVE PLAN FOR PROJECT:

The teamwork decided to follow the following actions in performing the project:

1. Literature survey and review of engine particle separators

2. Field survey to find out problems due to sand ingestion in flight operations in the kingdom

3. Collect data on environmental parameters regarding with airborne particulates in the kingdom (such as size, composition and concentration of sand)

4. Modeling of airflow through an engine air particle separator.

5. Determining basic specifications (size, velocity range, flow rate, sand concentration, and etc.) for experimental set-up to be used to simulate flow of air-sand mixture.

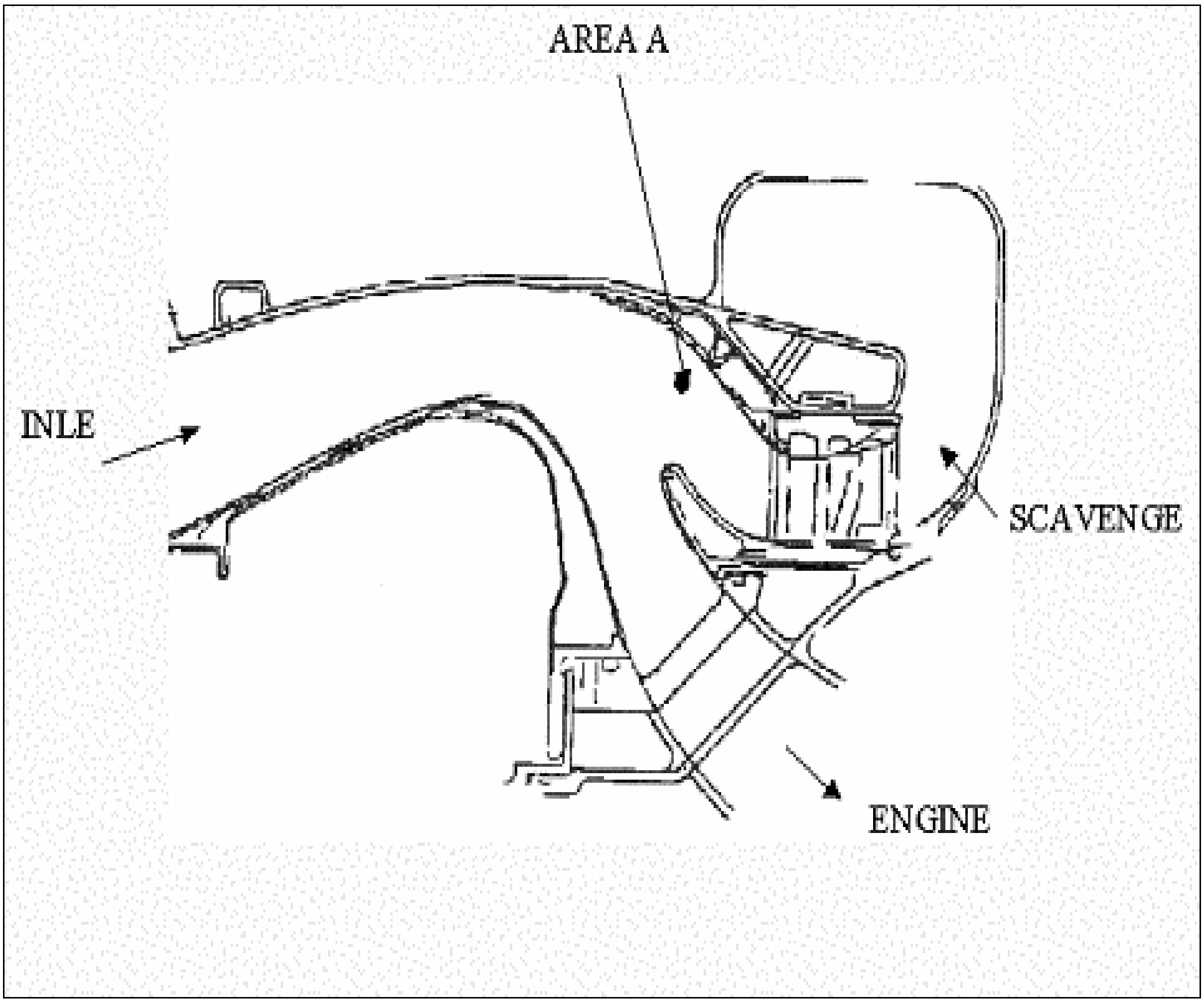
BACKGROUND:

Air particle separators operate on the principle of particle inertia; air coming into the separator is taken around a sharp bend, which the sand and dust particles, because of their inertia, are not able to turn quickly enough to follow. They are hence forced to pass into the scavenge duct and away from the engine.

As of today, every helicopter in the world fitted with an engine-mounted separator, is fitted with one, which looks and operates in exactly this way. That's a lot of helicopters and an awful lot of R&D to develop the separator technology to a mature state. In all probabilities, several tens of millions of pounds worth of R&D.

The separator, however, has a problem. it does not work very well. All of the contaminated air which passes along the scavenge duct has to be pumped. The energy required to provide the pumping force can be quite large- and on an aircraft every little bit counts. Therefore, the designs all attempt to minimize the amount of scavenge air. Unfortunately, but hardly surprising, the less air scavenged, the worse the separator performance becomes.

The fundamental problem here is that the scavenge annuals area has to be large in order to trap as many particles as possible, but on the other hand needs to be small because there is not very much flow passing through it and so that will tend to stagnate. So we will try in our own design of the air separator to compromise between all elements that give better efficiency of this separator. Figure 3 shows the shape such separator.

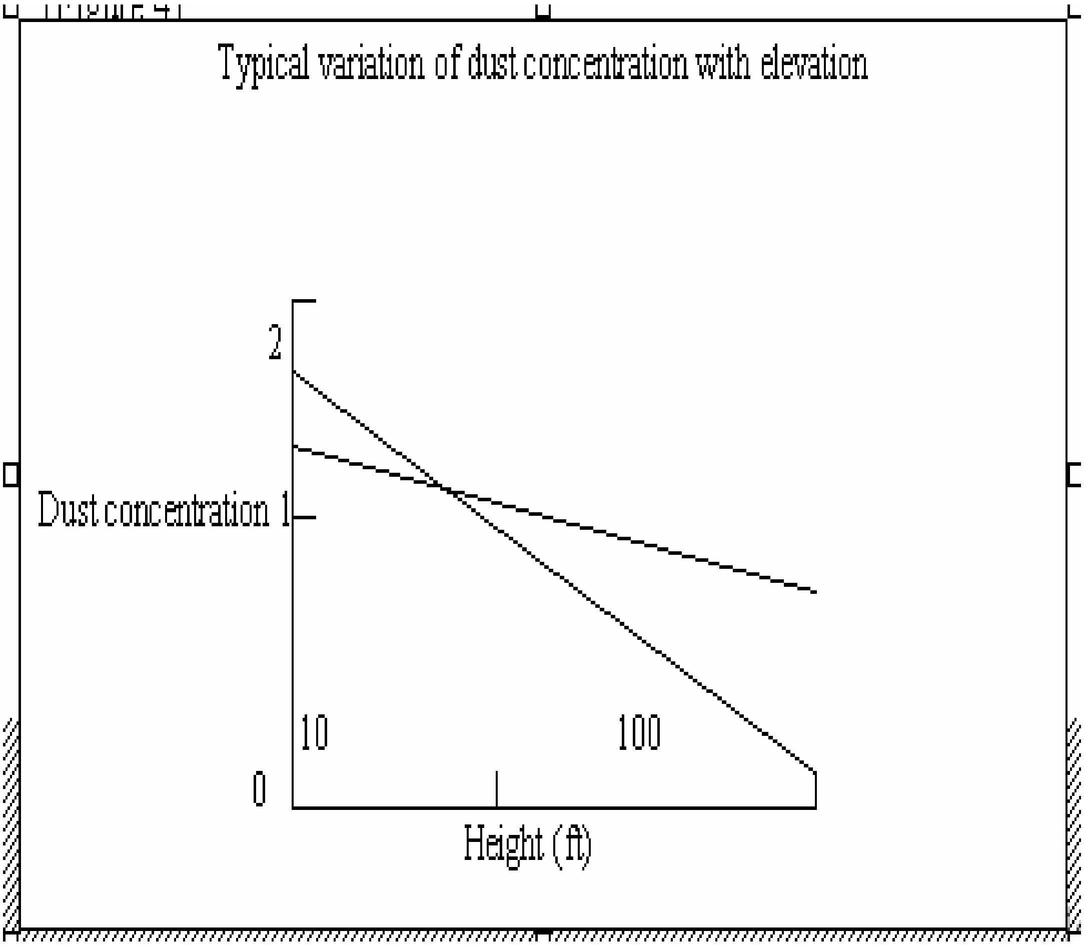


(Figure 3)

Environment:

Ambient air can be contaminated by solids liquids, or gases. Of these three, contamination by solids is the most common, usually the most serious situation. The quantity of solids can be defined in many ways, such as milligrams per cubic meter of air or grains per 1000 cubic feet. A measure General Electric finds convenient is parts per million (ppm), i.e., the mass of contaminants per million units mass of air. The fact that this is a convenient measure immediately demonstrates that the quantity of dust is generally quite small compared to the mass of air. In the United States, the Environmental Protection Agency samples airborne particulates periodically at some 4000 locations. Results of annual surveys are published.

Dust loading in desert regions, particularly those subject to sand and dust storms, is much higher than those usually experienced in the United States, and this is what makes the difficulty arises here in the kingdom to overcome this problem of sands. Concentrations in sand storms may reach several hundred ppm for periods of several hours, while long-term levels may average one to five ppm. When the wind blows in these regions, the larger soil particles become airborne first, smaller particles being more adherent. When the large particles fall back to earth, they disturb the surface and ‘splash out’ fine particles. By Stokes law, fine particles settle more slowly; so they remain airborne longer. The results are that the dust concentrations are highest close to the ground, and that the particles there tend to be coarser than at higher elevations. There is no exact relationship between dust concentrations and elevation above the ground, but available data generally tend to fall within the range in (Figure 4).


Typical variation of dust concentration with elevation
2
Dust concentration 1
0
10
100
Height (ft)

(Figure 4)

The size distributions of airborne dusts are variable with respect to time and place. In general, high values of dust concentrations tend to be associated with coarse dust and low values with fine dust. Large particles tend to fall quickly, while smaller particles are more likely to stay airborne. Consequently, dust samples taken near the source of contamination tend to be coarser than those samples taken at a distance.

Table 1 shows the idea of size distributions by reference to the standardized dusts, Arizona Coarse and Arizona Fine, which are widely used in the testing of air filtration devices. Table 1 shows their mass distribution as a function of particle size.

Particle size range (microns)	Nominal percentage of total mass of particles	
	Coarse dust	Fine dust
0 -5	12	39
5 -10	12	18
10 -20	14	16
20 -40	23	18
40 -80	30	9
80 -200	9	

(Table 1)

Types of sands and dust:

There is no single dust size, which can represent the full range of sand and dust encountered in flight operations, but the following sizes are defined as standard test dust

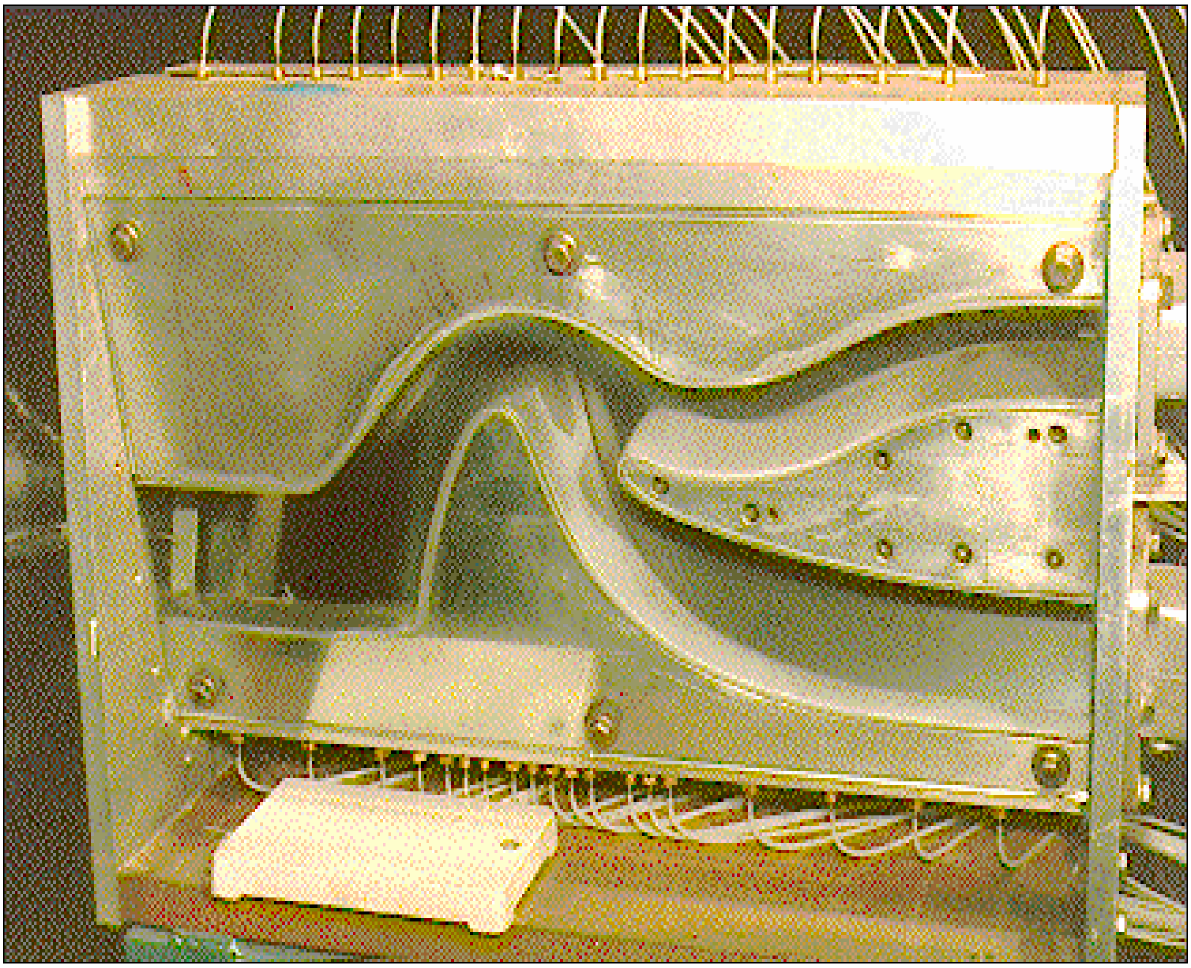
- MIL-E-5007C o ‘C-Spec’ sand- consists of crushed quartz with particle size from 0-1000 microns and mean diameter of about 200 microns. This type of sand is considered as highly erosive causing severe engine component damage.

- Coarse Arizona Road Dust (AC-Coarse)- has of low quartz content with particle sizes from 0-200 microns and mean diameter of about 80 microns. This type can cause erosion, glazing and clogging particularly in the hot section of the engine.

- Fine Arizona Road Dust (AC- Fine) consists of particle sizes from 0-80 microns and mean diameter of about 8 microns. Glazing and clogging particularly in the hot section of an engine are the usual phenomenon.

Inertial particle separator (IPS):

Inertial particle separators (Figure 5) are capable of moving large particles leaving the smaller ones to be trapped by the filters which greatly enhanced the life of the filter and offers maximum engine protection. The main advantage is the large installed area required for such a system thus increasing the overall intake area.

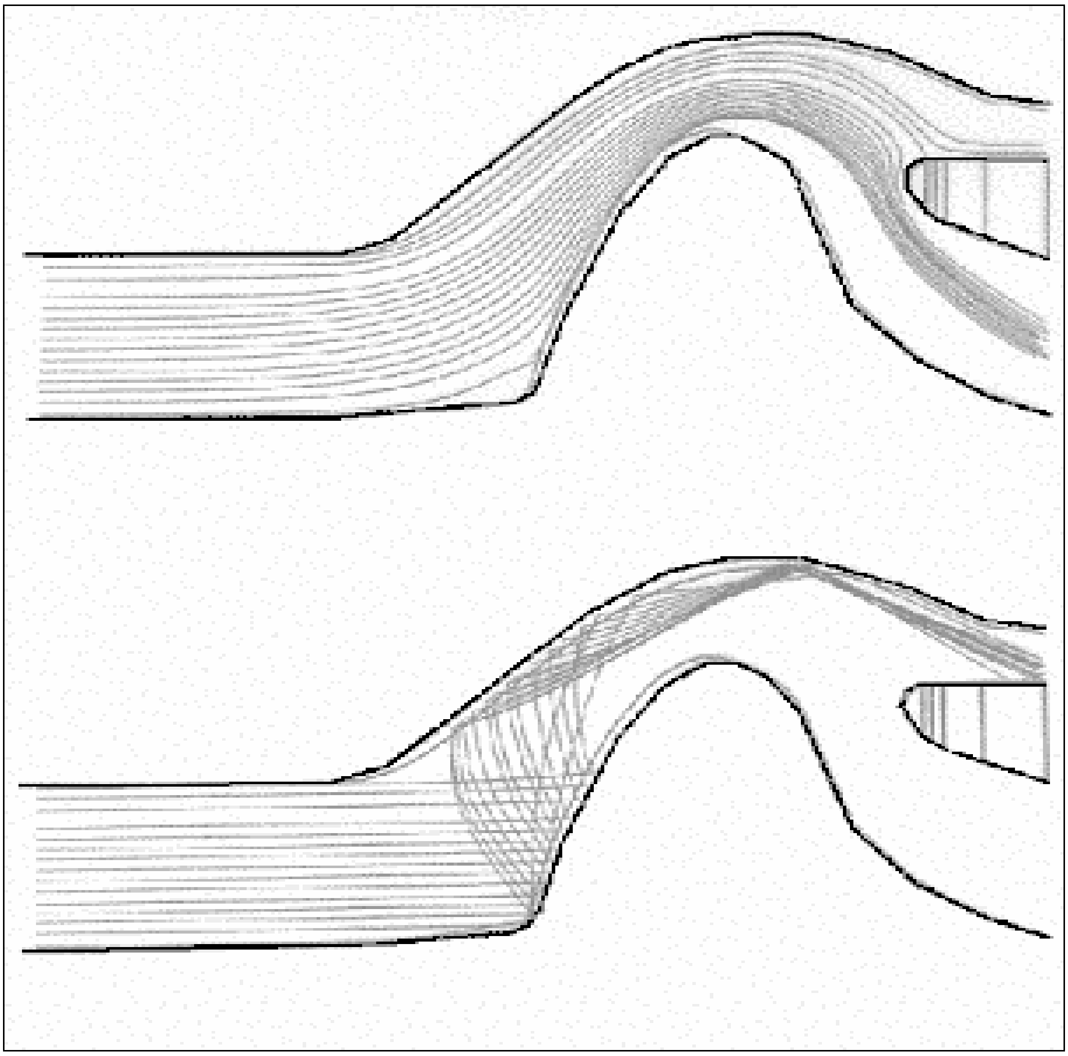


(Figure 5)

- In this project we will manufacture our own (IPS) according to the available compressor we have with certain specifications.

- The study will involve the experimental simulations of particle trajectories in an inertial particle separator in order to determine the sand separating efficiency for a range particle sizes. The mechanism of sand removal derives from the inertia and bounce characteristic of sand particles, which vary with the particle size, impact velocity and angle, and the target and particle materials.

A particle-air mixture is fed locally at the inlet into the (IPS) (shown in figure 5) , which has contoured shapes to deflect sand particle into the scavenge section while only clean air enters the compressor. The sand particles used have narrow size bandwidth. The ratio of the particles colleted from the scavenge to those ingested represents the separation efficiency for the particular dust size. Optimization of the separation efficiency is carried using 3D tracking computer code, which includes the particle bounce characteristics, shape of contoured surfaces and aerodynamic performance of the separator. Figure 6 shows how the particles propagate inside (IPS)



(Figure 6)

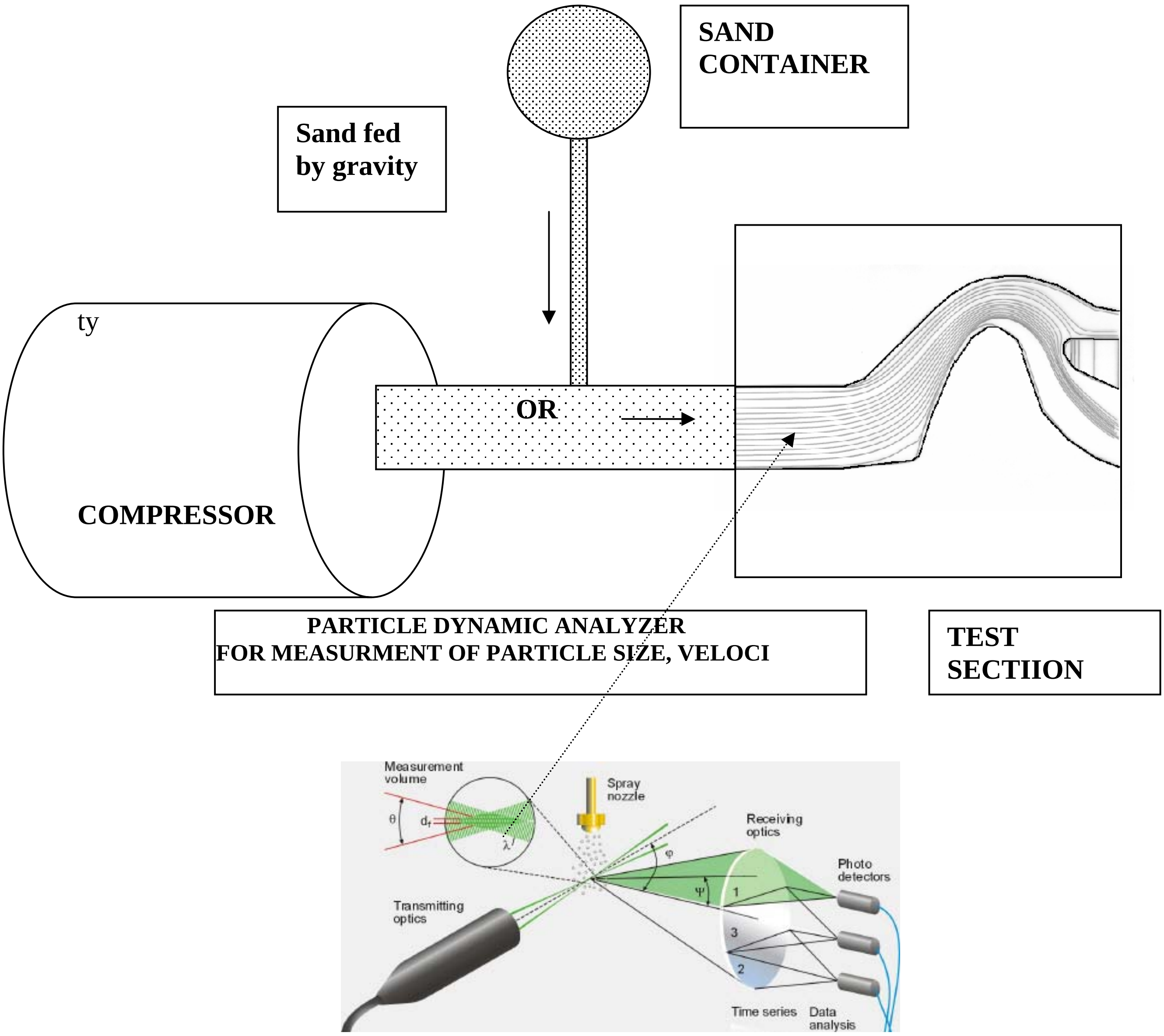
PRELIMINARY EXPERIMENTAL SET-UP FOR STUDYING INLET PARTICLE SEPARATOR:

Actually at this stage, we are concern of investigating the floe field characteristics inside the duct of air particle separator but for next stage the following suggested experiment set-up is suggested. For more details, look at the appendix.

We will make the set-up of our experiment as shown in figure 7 below with the following specifications:

Velocity Range	: 50-100 m/s
Sand concentration	: Variable (in mg/m3)
Model cross-section area	To be determined

(Table 2)


SAND
CONTAINER
Sand fed
by gravity
ty
COMPRESSOR
OR
PARTICLE DYNAMIC ANALYZER
FOR MEASURMENT OF PARTICLE SIZE, VELOCI
TEST
SECTIION
Measurement
volume
Spray
nozzle
Receiving
optics
Photo
detectors
Transmitting
optics
Time series
Data
analysis

(Figure 7)

PRELIMINARY FINDINGS REGARDING WITH TEST SET-UP :

(1) To obtain a velocity at the range of 50-100 m/s at cross-section area of minimum (5cmx10cm) 0.005m2, a compressor should supply an airflow of 15 m3/sec min.Preliminary data collected from various compressor manufacturers indicates that 15 m3/sec. is a high flow rate that adds the cost (More than 30,000 USD). After a survey in the local market (to be conducted by Mr.Salman) and examining the available compressor in the lab, a decisison will be made in a week or so on this matter.

(2) An instrument (particle dynamic analyzer or laser anemometer) to track the particle paths is needed. Responses from some manufacturers regarding with the prices of these types of instruments are expected nowadays.

SUGGESTIONS :

It is strong probability that purchasing a new compressor and measuremet instrument is costly and probably takes time.Under this conditions followings may be suggested:

- Seek for the possibility to improve the capacity of existing compressor by some modifications.

- One of the following two ways may be chosen to go on :

1) Project may be concentrated on ***aerodynamic performance*** i.e. experimental & theoretical investigation of flow field in the inlet particle separator without sand.This will help to develop and verify computer codes for the solution of the flow field which is needed for further steps of the project.

2) Or, a simple set up for studying ***sand erosion*** on surfaces may be designed by using available compressor.

I. Aerodynamic Performace of Inlet Particle Separators:

Procedure in performing experiment:

(1) Obtain the drawing of an existing particle separator

(2) Build a model of inlet particle separator.

(3) Test the model in the wind tunnel to obtain pressure and/or velocity distribution along the model and to determine the ratio of pressure before and after the separator. (If sufficient time is available)

(4) Establish model for air flow in the seperator

(5) Develop computer code for solution of equations

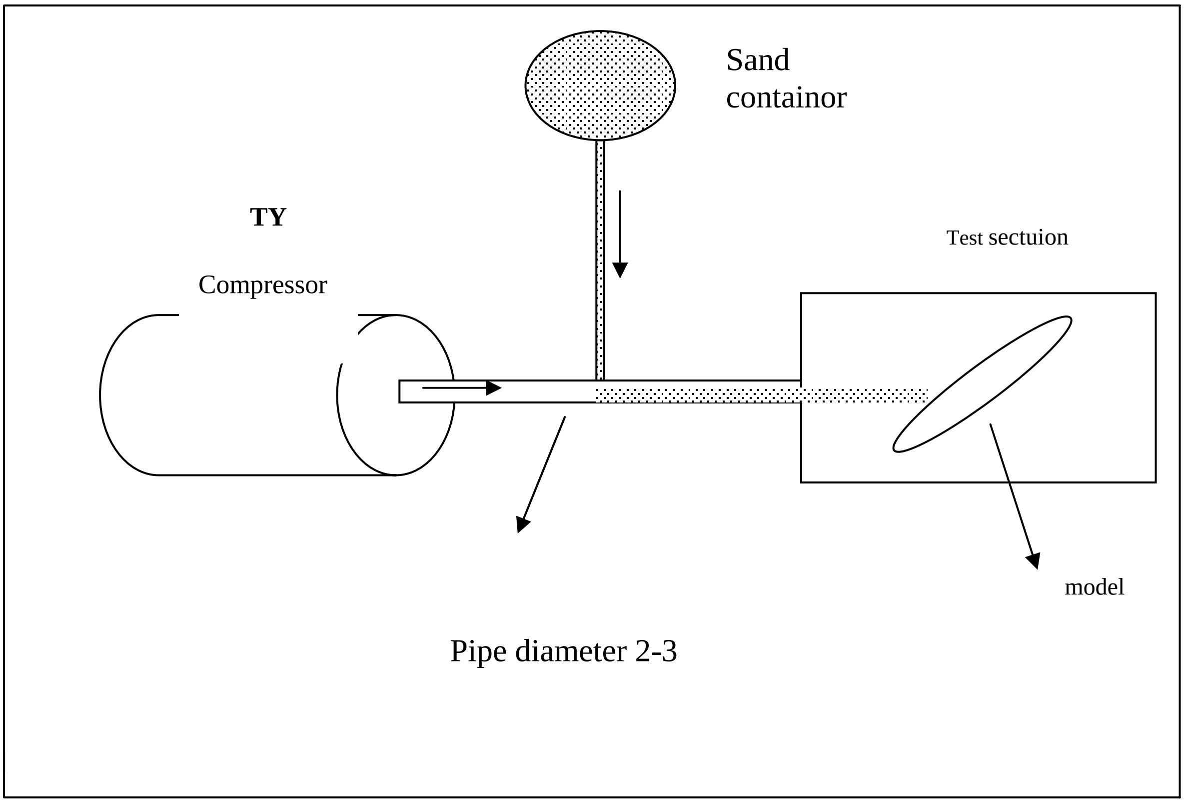
(6) Compare results with experimental findings

II. Sand particle erosion

(1) Establish the experiment set-up with a room for further improvements

(2) Build a particle feed system that supplies variable particle flow rate (some help may be needed from industry)

(3) Conduct preliminary test to study the effect of impact angle and impact velocity on the erosion pattren.figure 8 shows Simple experimental set-up for particle erosion study.


Sand
containor
TY
Compressor
Test sectuion
model
Pipe diameter 2-3

(Figure 8)

Theory and Analysis:

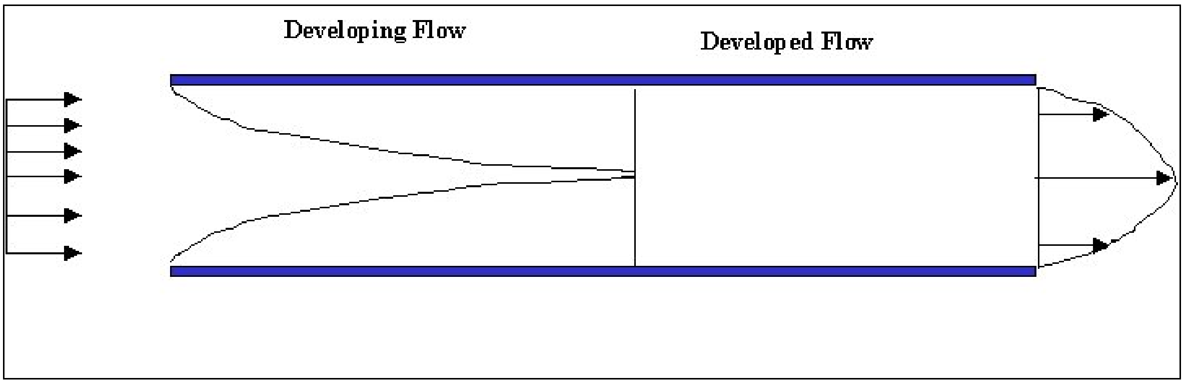
Pressure Losses in Ducts:

Pressure Drop Calculation for Rectangular Channels - Laminar Flow

The installation of air distribution systems and the type of duct fittings used play a major role in the overall system performance. It is crucial for the designer and the contractor to realize the impact pressure drop in flexible ducts and fittings can have on the power consumption and the overall performance of the HVAC system. To satisfy the ARI 210/240 requirements for minimum external pressure on indoor air-moving equipment, a proper estimate of pressure drops in the air-distribution system is critical. Proctor and Parker (2000) noted that the measured external static pressure in seven field tests (245 north American houses) were two to four times higher than the standard DOE assumptions. Pressure losses in an air distribution system are balanced by pressure increases at the installed fan. It is very important that every feasible means be used to control the fan power use. Increased flow resistance in the ducts results in increase in pressure drop, therefore lower airflow. This leads to increased fan power use and a lower heat exchanger efficiency (due to a lesser capacity). The combination of these effects can significantly increase power and energy consumption. Designing and properly installing duct systems that are energy efficient is therefore instrumental in achieving an overall energy efficient HVAC system. The available literature lacks sufficient description of pressure losses in flexible ducts and loss coefficients for other commonly used residential duct fittings such as, splitter boxes, outside air intake hoods and air supply boots.

Pressure drop affects the output of your fan or pump - and that is the volume of flow that is going do your cooling. Over the next couple of months, we will address several elements of a fan or pump-cooled system by looking at how to do some analytical calculations. These analytical calculations are just that - they're not perfect. But they're better than rules of thumb, and they're better than waiting for a test to tell you that you're in big trouble. And when you code the equations into a spreadsheet, they're really valuable for design work.

There are many examples of flow in rectangular channels in the cooling world. You have your basic heat exchangers, heat sinks (a misnomer if there ever was one - really they are heat exchangers too), stacks of two or more circuit boards, convection belt furnaces, even your car radiator. The analysis that follows might underestimate the pressure drop for circuit boards - they tend to be kind of rough topologically.



(Figure 9)

The most commonly applied correlations are for fully developed or developing laminar flow in parallel-plate or rectangular channels. Fully developed implies no change in the width-wise velocity profiles along the flow length (see Figure).

Typically these correlations are for a dimensionless form of the pressure drop, the friction factor f. The pressure drop is Dp=4f(L/Dh)(½rV2) where L is the flow length, Dh is the hydraulic diameter, r is the fluid density, and V is the average fluid velocity in the channel. Table 1 summarizes the correlations.

A quick note here: if you look at your basic fluid mechanics text, you might see the friction factor labeled as either Cf or f (or maybe even both). Also, the friction factor can be named after either "Fanning" or "Darcy." The Darcy version is four times the Fanning version, so the pressure drop equation either has a 4 in it or not. I'm using the Fanning version here, which puts the 4 in the pressure drop equation instead of in the friction factor equation.

Fully developed flow, parallel plates

There is an exact solution to the Navier-Stokes equations for fully developed laminar flow between parallel plates. The resulting velocity profile is parabolic. Don't worry, I'm not going to make you go through the calculus. For now, just think of this case as a rectangular channel with infinite aspect ratio.

Fully developed flow, rectangular channels

The pressure drop depends on the channel aspect ratio. The parallel plate value is modified by a function of the aspect ratio (see Table 1) to obtain the values of the pressure drop parameter,fRe, commonly tabulated in textbooks, for example White (1991). This tabulation also appears in the classic reference, Shah and London (1979). The polynomial equation in Guyer (1989) is more useful for spreadsheet work.

Developing flow, parallel plates

The correlations for developing flow between parallel plates assume a uniform velocity profile at the channel inlet. The velocity profile gradually changes from uniform to having a parabolic ("fully developed," see Figure) profile. The transition region is the hydrodynamic entry length, usually measured in terms of x+; x+ is the actual flow length scaled by the hydraulic diameter and the Reynolds number (see Table).

To obtain the pressure drop across the entire channel, an apparent (vs. local) friction factor is defined in Guyer (1989), based on the effective channel length. I find this to be a lot of trouble to go to, and not that accurate if I'm looking at thick plates (like extruded heat sink fins), where the area contraction accelerates the flow development. To simplify, stick with fully developed flow, and treat the result as a lower limit if you think you have a long region of developing flow. Of course, you can go to the extra trouble and make yourself a spreadsheet that has all the detail in it. The beauty of writing this stuff into a spreadsheet is that you only have to do the coding once. After that, the computer does the repetitive, boring calculations for you.

Entrance and exit effects

The typical correlations given in the literature for pressure drops across changes in channel area are in the form of a pressure loss coefficient, K=Dp/(½rV2). The coefficient usually depends on the free area ratio. See the Table. The velocity to use is the one in the small cross-section (i.e. inside the channel). The pressure drop is the above equation rearranged (one of my college profs called it "turned inside out").

Fully developed, parallel plates, spacing s	$f = 24/Re$ $Re = \rho V D_h/\mu$ $D_h = 2s$	White (1991)
Fully developed,rectangular channels, area A and perimeter P	$f = (24/Re)(1 - 1.3553a + 1.9467a^2 - 1.7012a^3 + 0.9564a^4 - 0.2537a^5)$ $D_h = 4A/P$	aspect ratio a < 1 Guyer (1989) Shah and London (1979)
Developing, parallel plates	$f_{pp,dev} = 3.44(\frac{1}{Re})/\sqrt{x^+} + (24 + 0.674/x^+ - 3.44/\sqrt{x^+})/(1 + 2.9e^{-5}/(x^+)^2)$	x+= L / (Dh Re)Guyer (1989)
Entrance loss coefficient	Kloss_in = $0.8 + 0.04\sigma - 0.44\sigma^2$	σ=free area ratio Adapted from Kays & London (1964)
Exit loss coefficient	Kloss_out = $1 - 2.4\sigma + \sigma^2$	Adapted from Kays & London (1964)

(Table 3)

Total pressure drop:

Calculate the pressure drop for all the flow lengths and area changes that an air particle would see as it flows through your system. Add 'em all up, and you're there! At least, you've got the pressure drop for a specific value of the channel velocity, V. Therein lies the rub - you usually don't know V. It's what you're after in the first place, because that's the fluid speed that is going to do your cooling. The whole reason you are going through this little rigmarole is to find the temperature of some surface or other. So let's sketch out a sequence of calculations (we'll cover these in the next several columns).

The test objectives were to measure the **pressure** drop in various residential duct components and expressing the results in terms of power law coefficients for straight ducts, and local loss coefficients for the fittings. The test procedures were based on proposed ASHRAE Standard 120P and involved different lengths of sheet metal duct sections installed upstream and downstream of the test specimens, air moving fans, airflow measuring devices, and data loggers and hand-held manometers. As required by ASHRAE Standard 120P, piezometer rings, the same diameter as the test ducts, were used that each have four equidistant static **pressure** taps for upstream and downstream **pressure** measurements. Airflows were measured using either a 6” (150 mm) nozzle flowmeter (±0.5% of reading accuracy) or a combined fan/flowmeter device with ±3% accuracy. Flow straighteners that reduce swirl and turbulence were incorporated into the experimental apparatus and the flow meters. For the splitter box tests a fan/flowmeter was mounted on each downstream leg of the splitter boxes to suck air through the test system and measure the flow through each leg. The 6” (150 mm) nozzle flowmeter was used to measure the total airflow through the main branch upstream of the splitter box. All pressure and flow measurements were averaged for five seconds and the readings were recorded using a data logger. In addition to the data loggers, hand-held electronic digital pressure gauges were used in the supply boot and the splitter box tests to modulate different pressure/flow stations. The experimental results were corrected for temperature changes during the test and for changes in flowmeter calibrations with temperature. Also elevation corrections were made because some of the tests were performed at sea level and others at several hundred feet elevation. Throughout the tests, the volumetric flow rate ranges were those that are encountered in typical residential systems. The results of the component analysis were compared with available references, whenever a similar duct fitting was reported in the literature. A further evaluation of the components analysis results was performed through a comparison with measured **pressure** drop in an installed air distribution system.

Potential Flow Theory:

$$\lim_{\delta t \to 0} \frac{\partial \theta_2}{\partial t} = -\frac{\partial u}{\partial y}$$

The average of the two quantities is known as the **rotation**, ω

$$\omega = \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)$$

when ω is non zero the flow is said to be rotational and **'vorticity'** ζ (zeta) is said to exist

$$\zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

Thus the condition for two dimensional flow to be irrotational is

$$\frac{\partial v}{\partial x} = \frac{\partial u}{\partial y}$$

In general for three dimensional irrotational flow

$$\frac{\partial w}{\partial y} = \frac{\partial v}{\partial z} \;;\; \frac{\partial u}{\partial z} = \frac{\partial w}{\partial x} \;;\; \frac{\partial v}{\partial x} = \frac{\partial u}{\partial y}$$

For a given point $P(x,y)$ on a streamline, making an angle θ to the horizontal, the relationship

$$\frac{v}{u} = \tan\theta = \frac{dy}{dx}$$

and for three dimensional flow

$$\frac{dx}{u} = \frac{dy}{v} = \frac{dz}{w}$$

Potential function and Stream function

The kernel of potential flow theory is the definition of a velocity potential ϕ, such that

$$u = -\frac{\partial \phi}{\partial x} \left[\frac{\partial \phi}{\partial x}\right] \quad v = -\frac{\partial \phi}{\partial y} \left[\frac{\partial \phi}{\partial y}\right] \qquad \textbf{4}$$

where u is the velocity in the x direction and v is the velocity in the y direction.
As an alternative a stream function ψ, may be defined similarly as

$$u = -\frac{\partial \psi}{\partial y} \quad \left[\frac{\partial \psi}{\partial y}\right] \quad v = \frac{\partial \psi}{\partial x} \quad \left[-\frac{\partial \psi}{\partial x}\right] \qquad \textbf{5}$$

Other texts use a positive sign for the stream function and opposite signs for the stream function , terms in brackets).

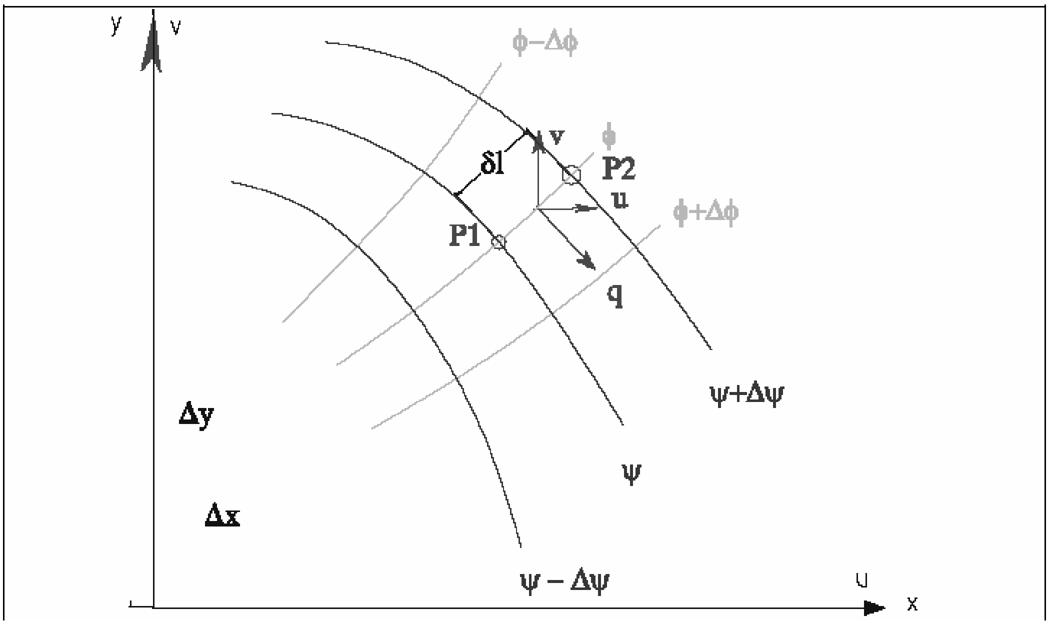
from previous equations:

$$v dx - u dy = 0$$
$$\frac{\partial \psi}{\partial x} dx + \frac{\partial \psi}{\partial y} dy = 0$$
$$\frac{\partial \psi}{\partial x} \delta x + \frac{\partial \psi}{\partial y} \delta y = 0 = \delta \psi$$

thus along any streamline ψ must be constant.

For a pair of streamlines the flow normal to a line, of length δl, joining both stream lines is

$$\delta Q = (u \sin\theta - v \cos\theta)\delta l$$
$$= \left(u \frac{\delta y}{\delta l} - v \frac{\delta x}{\delta l}\right)\delta l$$
$$= u\delta y - v\delta x$$
$$= -\frac{\partial \psi}{\partial y}\delta y - \frac{\partial \psi}{\partial x}\delta x = -\delta \psi$$



(Figure 10)

Similar relationships are obtained by considering the potential function. Equi-potential lines are a series of lines drawn at right angles to the streamlines. For steady flow along any potential line

$$\delta\phi = \frac{\partial \phi}{\partial x}\delta x + \frac{\partial \phi}{\partial y}\delta y = 0$$
$$-udx - vdy = 0$$

so that along a given equi-potential

$$\frac{dy}{dx} = -\frac{u}{v} \qquad \textbf{6}$$

hence from equation (3) potential lines and stream lines are orthogonal.

If any of the expressions for u and v are substituted into equation (5) it is seen that

$$\frac{\partial^2 \phi}{\partial x \partial y} - \frac{\partial^2 \phi}{\partial x \partial y} = 0$$

The use of a velocity potential to model a flow implies certain assumptions must be made, these are -

1 The fluid behaves as an ideal fluid (inviscid and incompressible), and consequently the flow is irrotational since there can be no shear forces applied to an inviscid fluid.
2 There is no separation between fluid and solid boundaries.
3 The continuity equation must be satisfied. (see below)
4 Newtons second law of motion applies at all time and at every point. "*F=ma*"

Continuty Relationship

The general expression for three dimensional flow is

$$\frac{1}{\rho}\left(\frac{\partial \rho}{\partial t}+u\frac{\partial \rho}{\partial x}+v\frac{\partial \rho}{\partial y}+w\frac{\partial \rho}{\partial z}\right)+\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \qquad \mathbf{8}$$

Thus for incompressible steady flow in two dimensions

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \qquad \mathbf{9}$$

Laplace Equation

By substitution of equation 4 into equation 9 yields

$$\frac{\partial^2 \phi}{\partial x^2}+\frac{\partial^2 \phi}{\partial y^2}=0 \quad \text{or} \quad \nabla^2 \phi = 0 \qquad \mathbf{10}$$

which is the well known Laplace equation.

Polar Coordinates

for a point *P(x,y)*, the following holds

$$x = r\cos\theta \;;\; y = r\sin\theta \;;\; r = \sqrt{x^2+y^2} \;;\; \theta = \tan^{-1}\frac{y}{x}$$

Two velocities may be defined, one radially outwards v_r and one tangential component, v_t. Recalling that the flowrate between two streamlines is numerically equal to the difference in the value of the streamfunction, then

$$-\delta\psi = v_r\, r d\theta - v_t\, dr$$

Also since ϕ is a function of r and θ then

$$-\delta\psi = v_r\, r d\theta - v_t\, dr$$

Also since ϕ is a function of r and θ then

$$\delta\psi = \frac{\partial\psi}{\partial\theta}\delta\theta + \frac{\partial\psi}{\partial r}\delta r$$
$$= \frac{1}{r}\frac{\partial\psi}{\partial\theta} r\delta\theta + \frac{\partial\psi}{\partial r}\delta r$$

thus it is evident that

$$v_r = -\frac{1}{r}\frac{\partial\psi}{\partial\theta} \;;\; v_t = \frac{\partial\psi}{\partial r} \qquad \textbf{11}$$

In terms of the potential function

$$v_r = -\frac{\partial \phi}{\partial r} \;;\; v_t = -\frac{1}{r}\frac{\partial \phi}{\partial \theta} \qquad \textbf{12}$$

In terms of polar coordinates then the Laplace equation is

$$\frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r}\frac{\partial \phi}{\partial r} = 0 \qquad \textbf{13}$$

or

$$\frac{\partial^2 \psi}{\partial r^2} - \frac{1}{r}\frac{\partial \psi}{\partial r} = 0 \qquad \textbf{14}$$

Finite Difference Method:

The finite difference method is the most commonly used numerical method for solving differential equations. The derivatives in the differential equation are replaced by appropriate difference-quotient approximations.
By Taylor's series expansion of $u(x+h)$, we have

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2!}u''(x) + \cdots \qquad (2\text{-}2)$$

Evaluating (2-2) at $x = x_i$ yields

$$u_{i+1} = u_i + hu_i' + \frac{h^2}{2!}u_i'' + \cdots \qquad (2\text{-}3)$$

where u_i and u_{i+1} are the values of u at x_i and x_{i+1} respectively.
By the same argument, we have

$$u_{i-1} = u_i - hu_i' + \frac{h^2}{2!}u_i'' - \cdots \qquad (2\text{-}4)$$

By adding (2-3) and (2-4) and rearranging, we have

$$u_i'' = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O(h^2) \qquad (2\text{-}5)$$

By subtracting (2-4) from (2-3) and rearranging, we have

$$u_i' = \frac{u_{i+1} - u_{i-1}}{2h} + O(h^2) \qquad (2\text{-}6)$$

where $O(h^2)$ is the error introduced by truncating the series. Since h is very small, both terms can be neglected. Equations (2-5) and (2-6) are called the central-difference formulae for $u''(x_i)$ and $u'(x_i)$ respectively.
By substituting (2-5) and (2-6) into (2-1), we have

$$u_{i+1}\left(\frac{p}{h^2} + \frac{q}{2h}\right) + u_i\left(r - \frac{2p}{h^2}\right) + u_{i-1}\left(\frac{p}{h^2} - \frac{q}{2h}\right) = f_i \qquad (2\text{-}7)$$

for i = 1, 2, ..., n - 1, where f_i is the value of f at x_i.

At the two boundary points, we have

$$\begin{cases} u_0 &= \alpha \\ u_n &= \beta \end{cases} \tag{2-8}$$

By substituting (2-8) into (2-7), we have a system of linear equations with unknowns,

$u_1, u_2, \cdots, u_{n-1}$. Let $a = \frac{p}{h^2} - \frac{q}{2h}$, $b = r - \frac{2p}{h^2}$, and $c = \frac{p}{h^2} + \frac{q}{2h}$, the system of equations may be written in matrix vector form as follows

$$\begin{bmatrix} b & c & & & \\ a & b & c & & \\ & \ddots & \ddots & \ddots & \\ & & a & b & c \\ & & & a & b \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} f1 - a\alpha \\ f_2 \\ \vdots \\ f_{n-2} \\ f_{n-1} - c\beta \end{bmatrix} \tag{2-9}$$

If p<0 , $r \geq 0$ and $h \to 0$, the coefficient matrix of (2-9) is tridiagonal and diagonally dominant, the solution by LU decomposition is numerically stable. By solving the system of linear equations, (2-9), the solution of (2-1) on each node point can be determined. The approximated function value of u on any point within the [0, 1] can be estimated by linear interpolating two nearby node points.

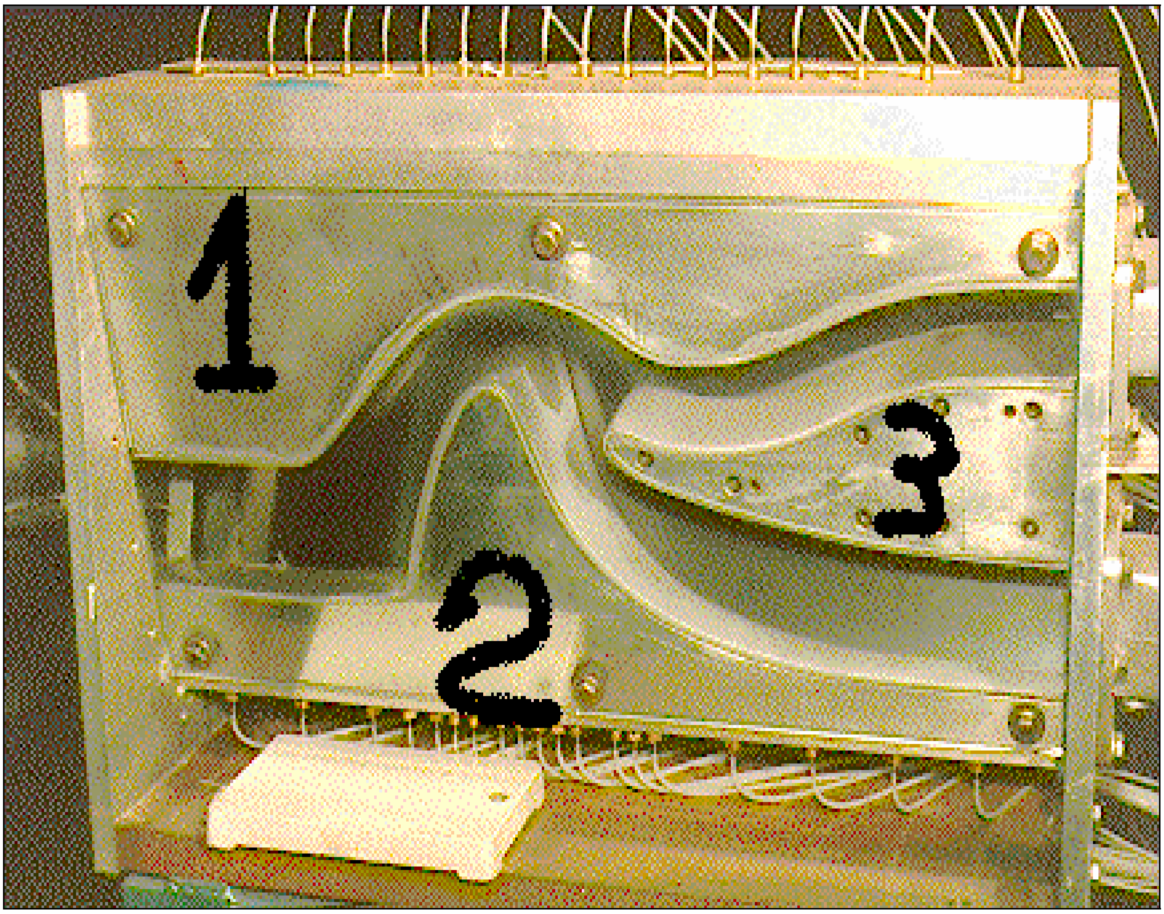
It is known that the a priori error estimates for finite difference method is given by

$$\|u - u_h\| \leq Ch^2 \tag{2-10}$$

Where C is a constant which depends on the exact solution u and its derivatives.

DRAWINGS:

All the parts of our inlet particle separator will be drawn individually we numbered these parts as hown in figure 9 below .



(figure 11)

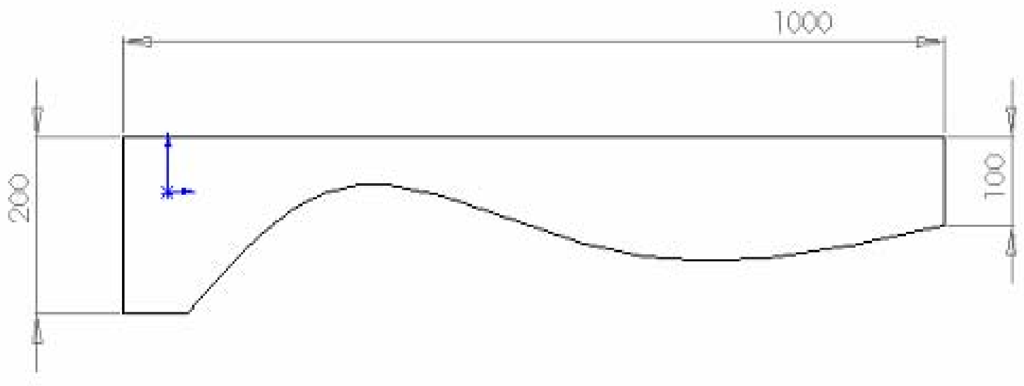
We present the drawing of each part above as well as the assembed drawing from different faces as follows:

a) front
b) Top
c) Bottom
d) Right
e) Left
f) Isometric
g) Dimetric
h) Trimetric

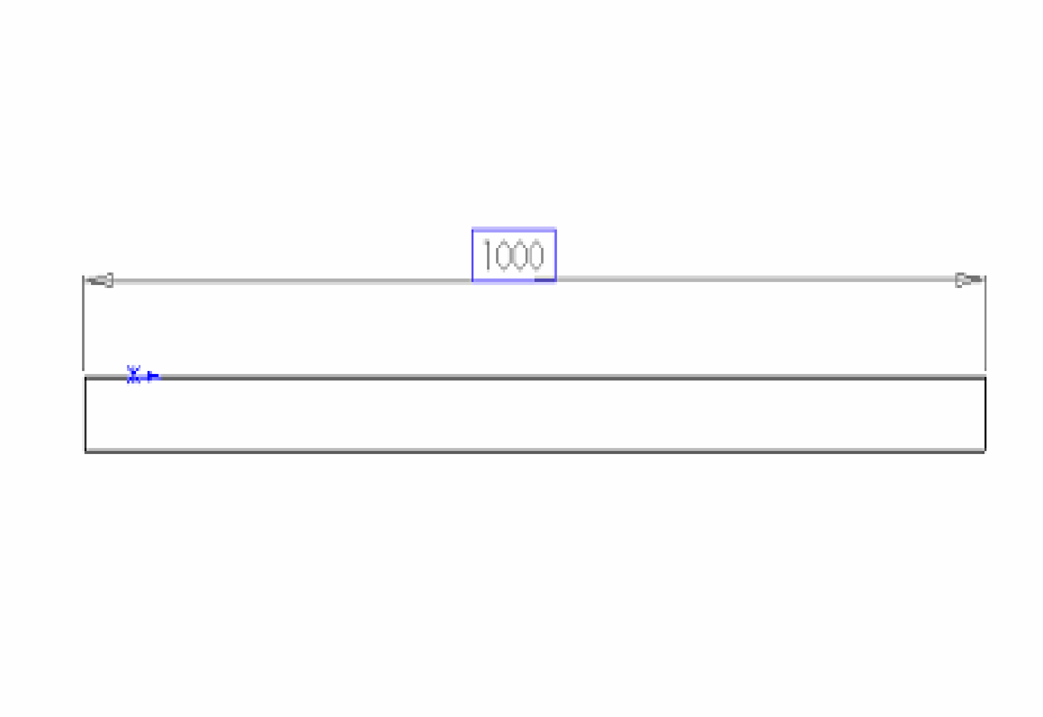
For all detailed drawing, look at Appendix B.

Part 1:

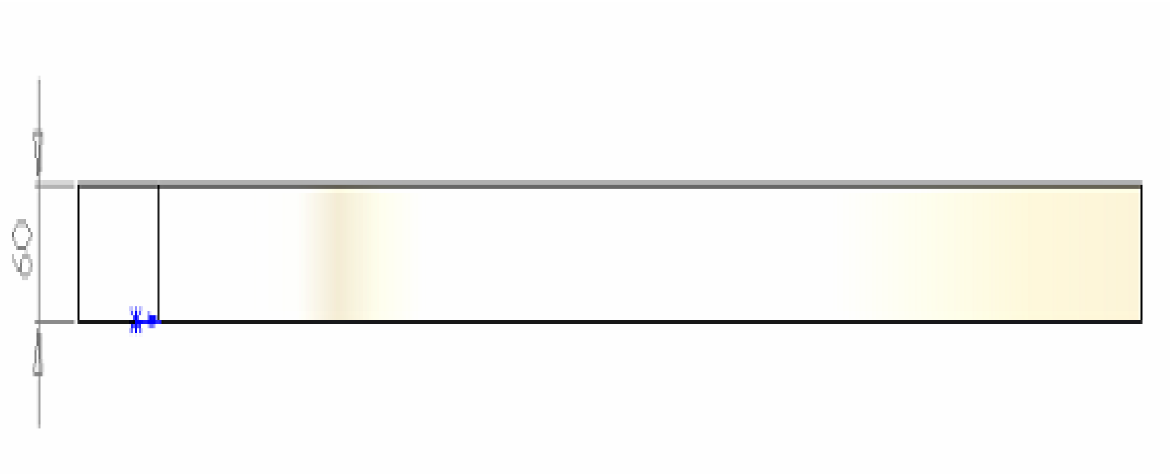
Front:



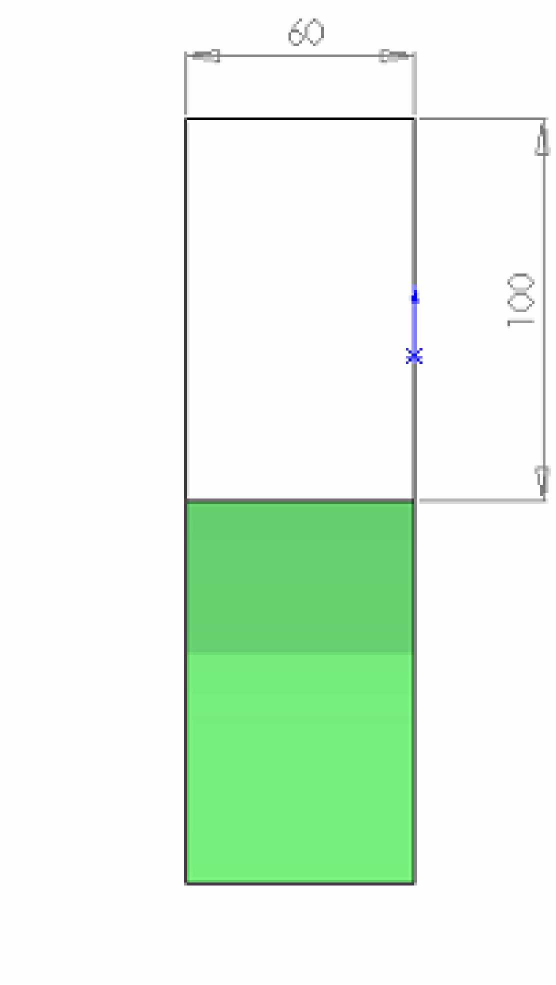
Top:



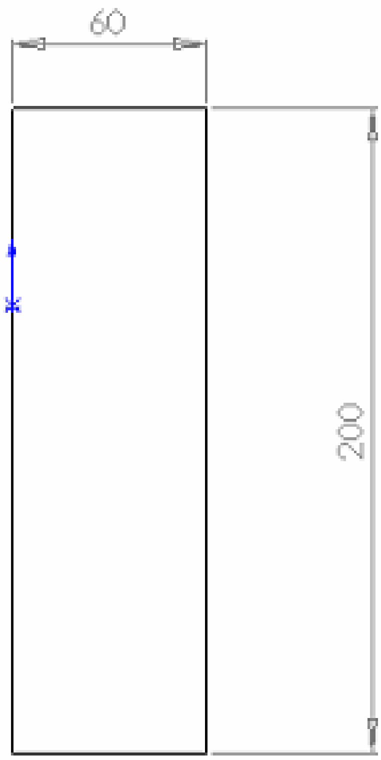
Bottom:



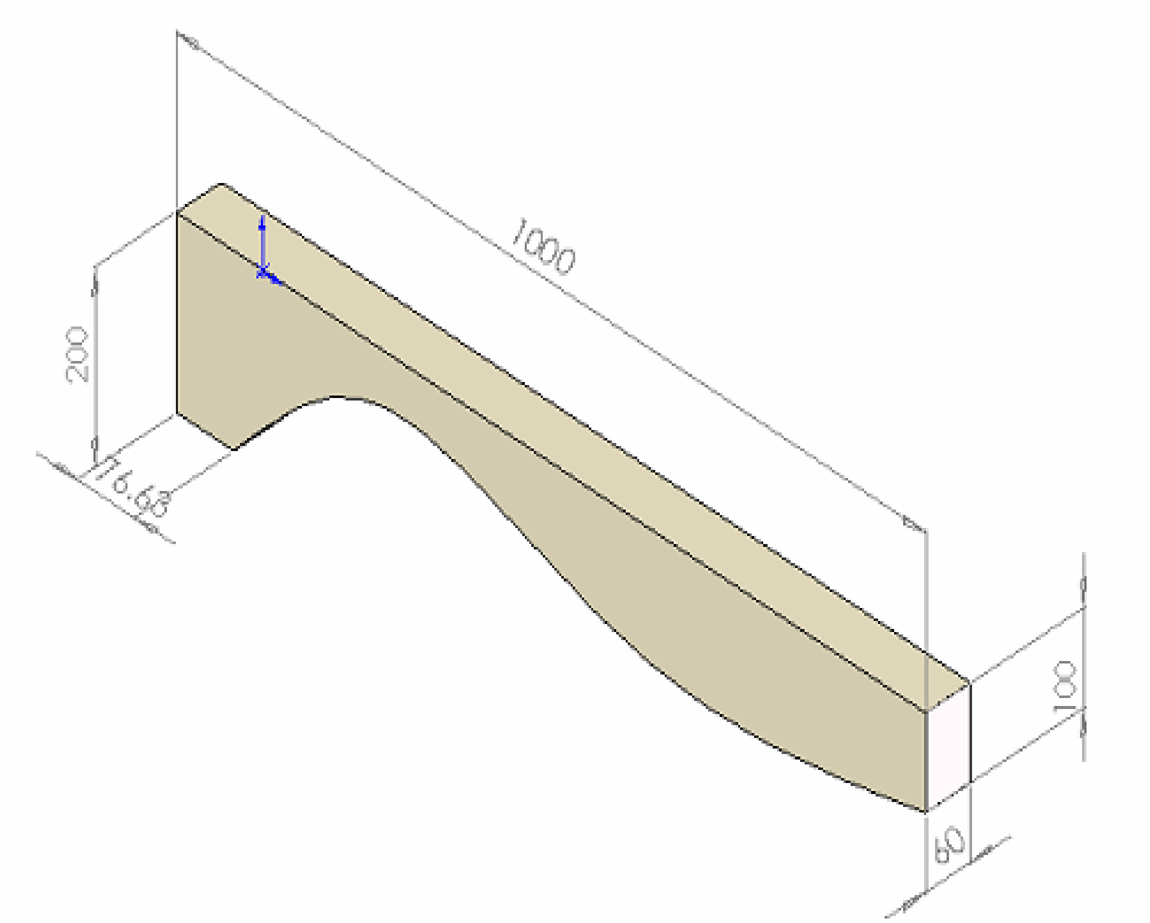
Right veiw:



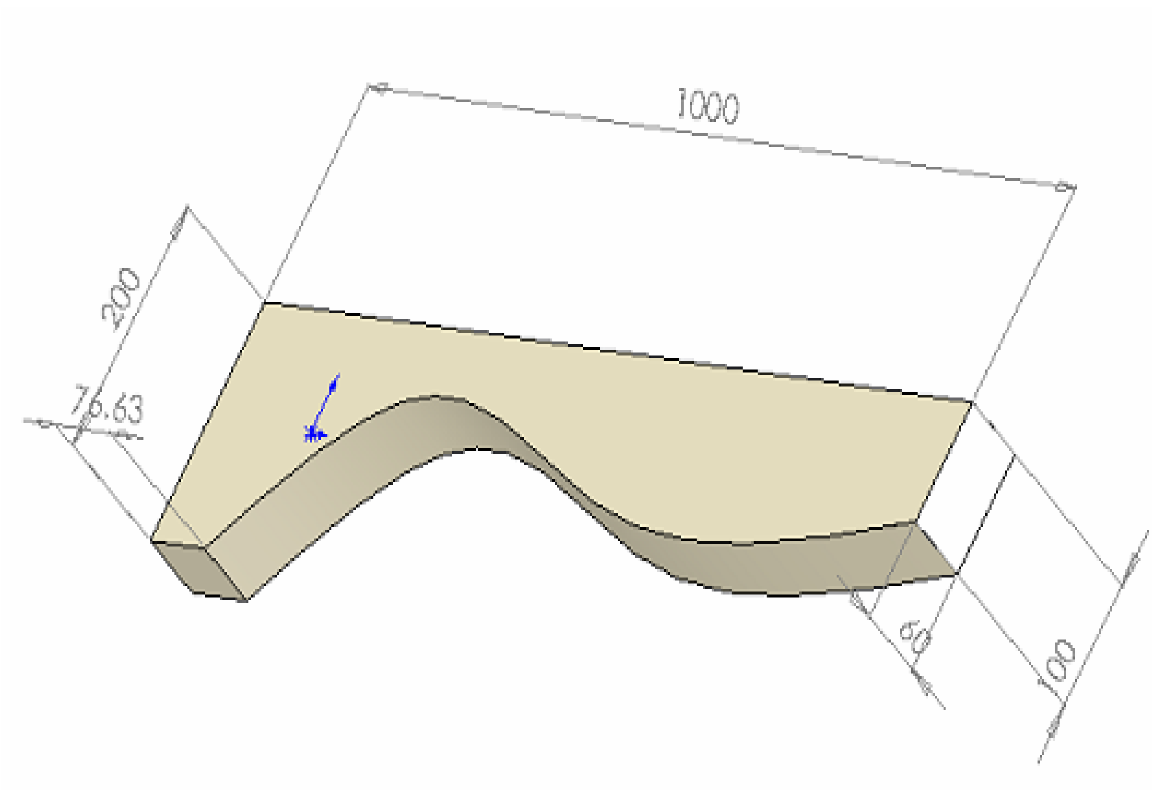
Left view:



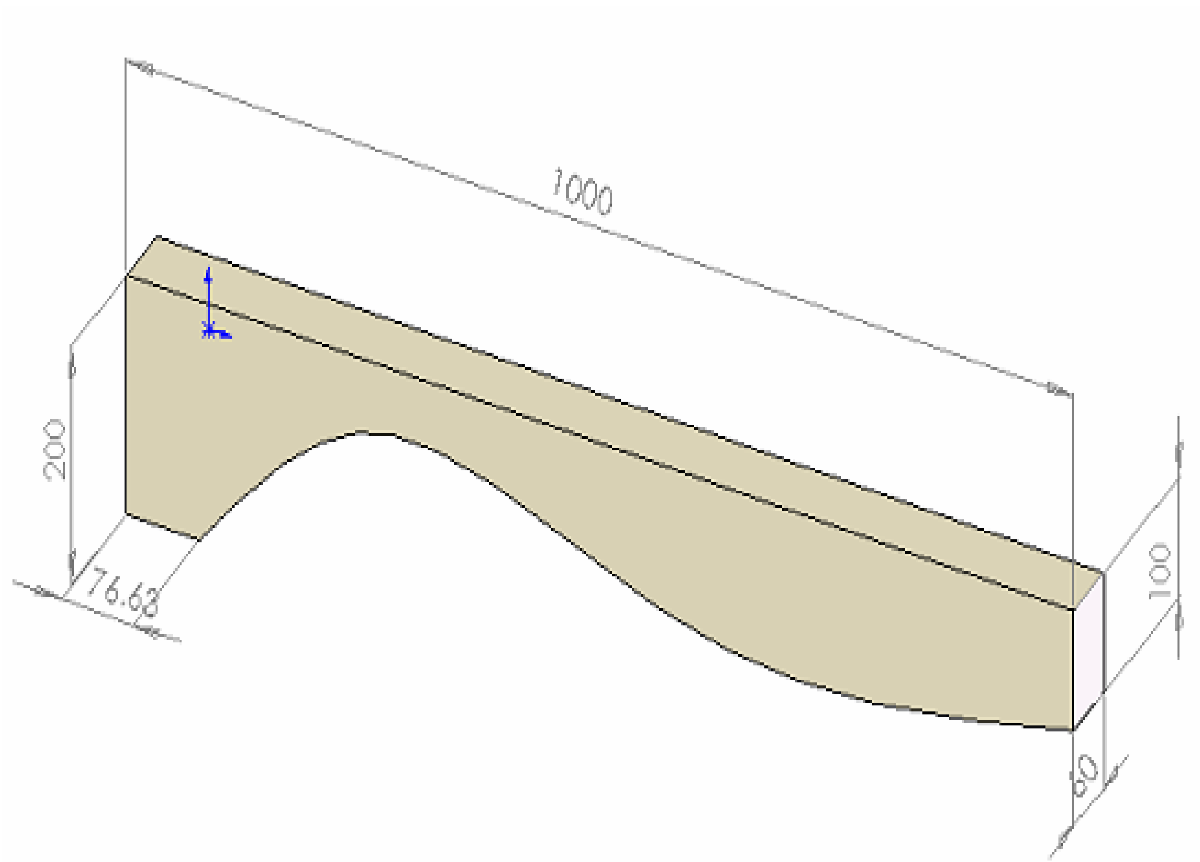
Isometric view:



Trimetric view:

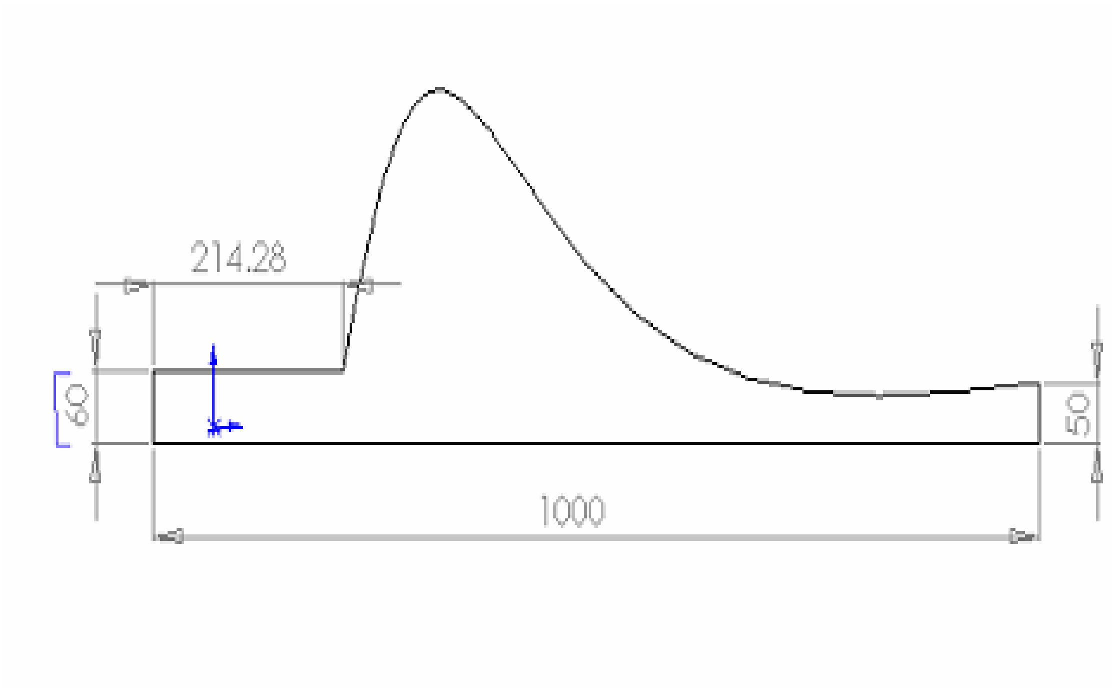


Diametric view:

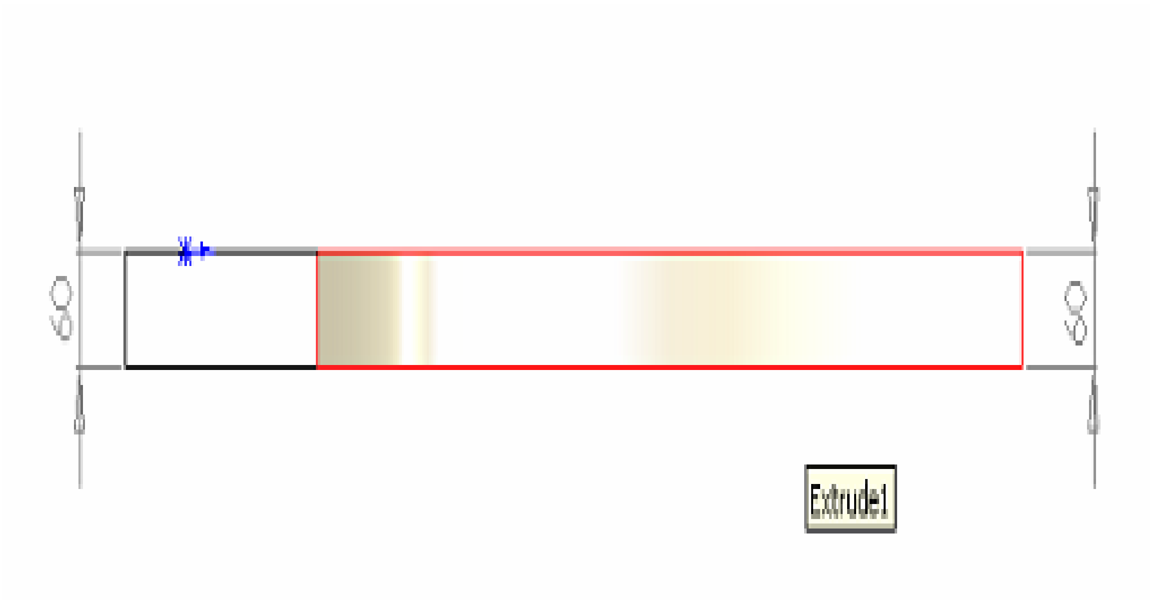


Part 2:

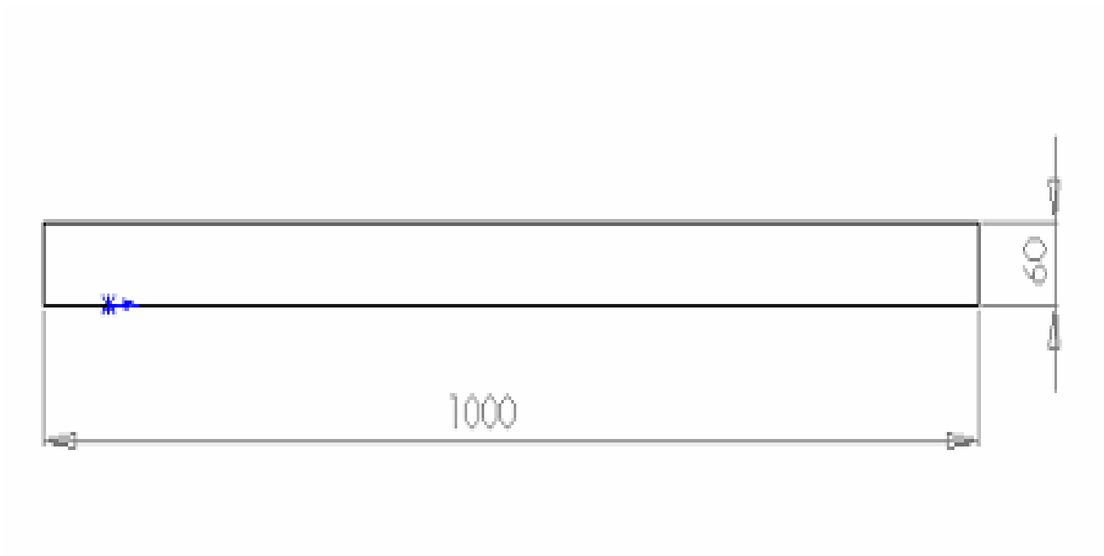
Front view:



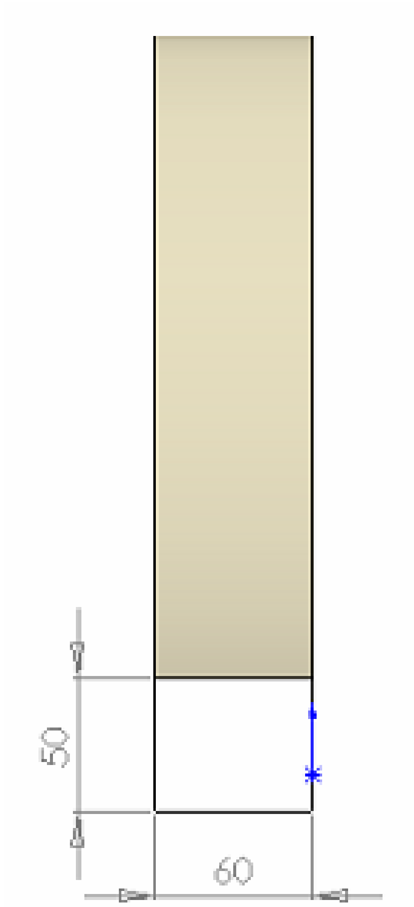
Top view:



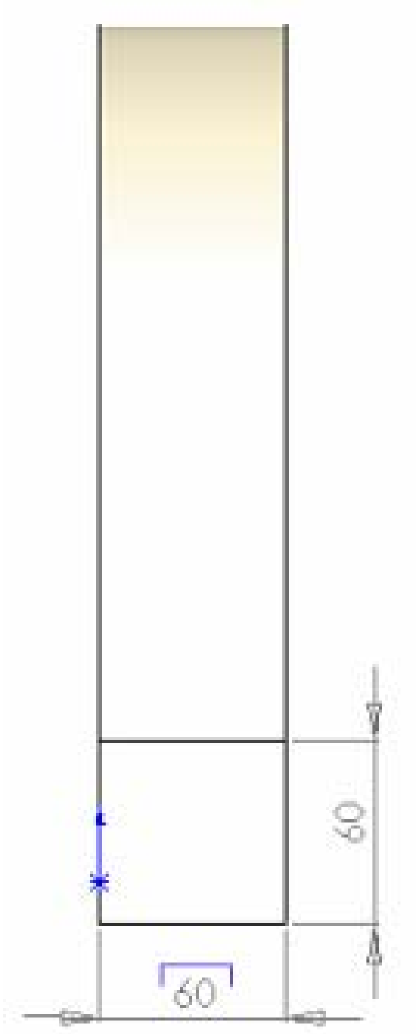
Bottom view:



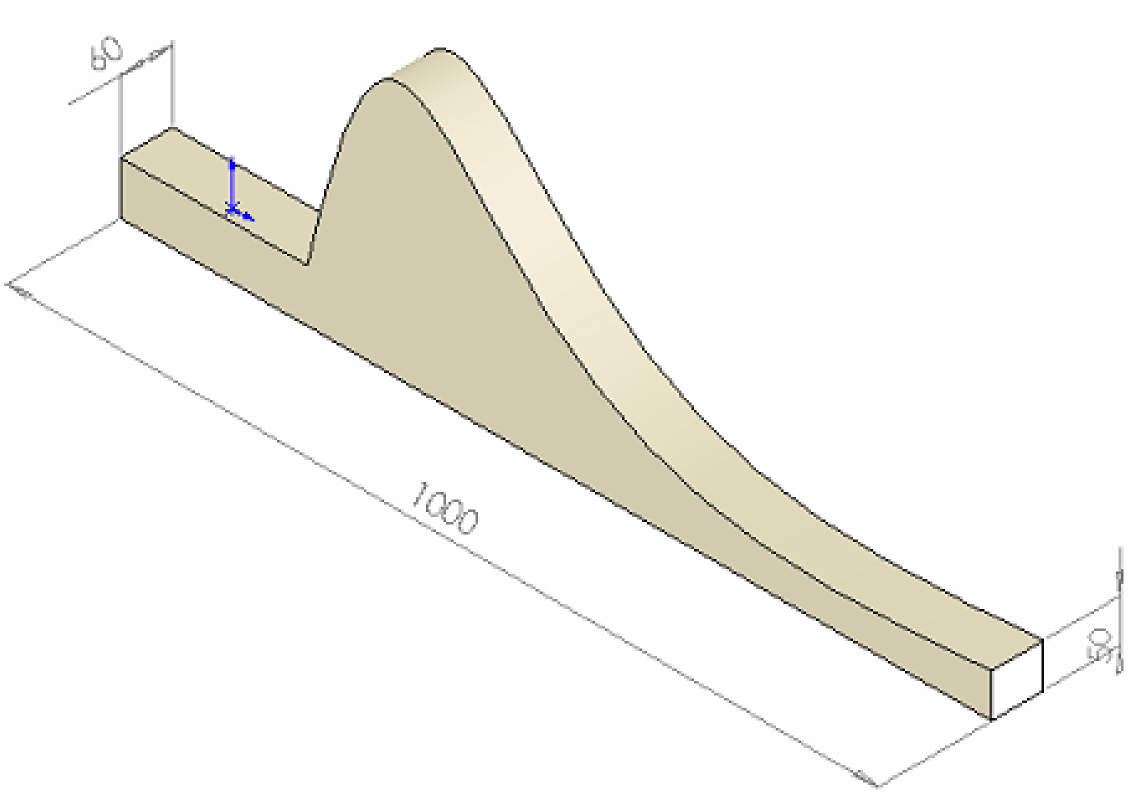
Right view:



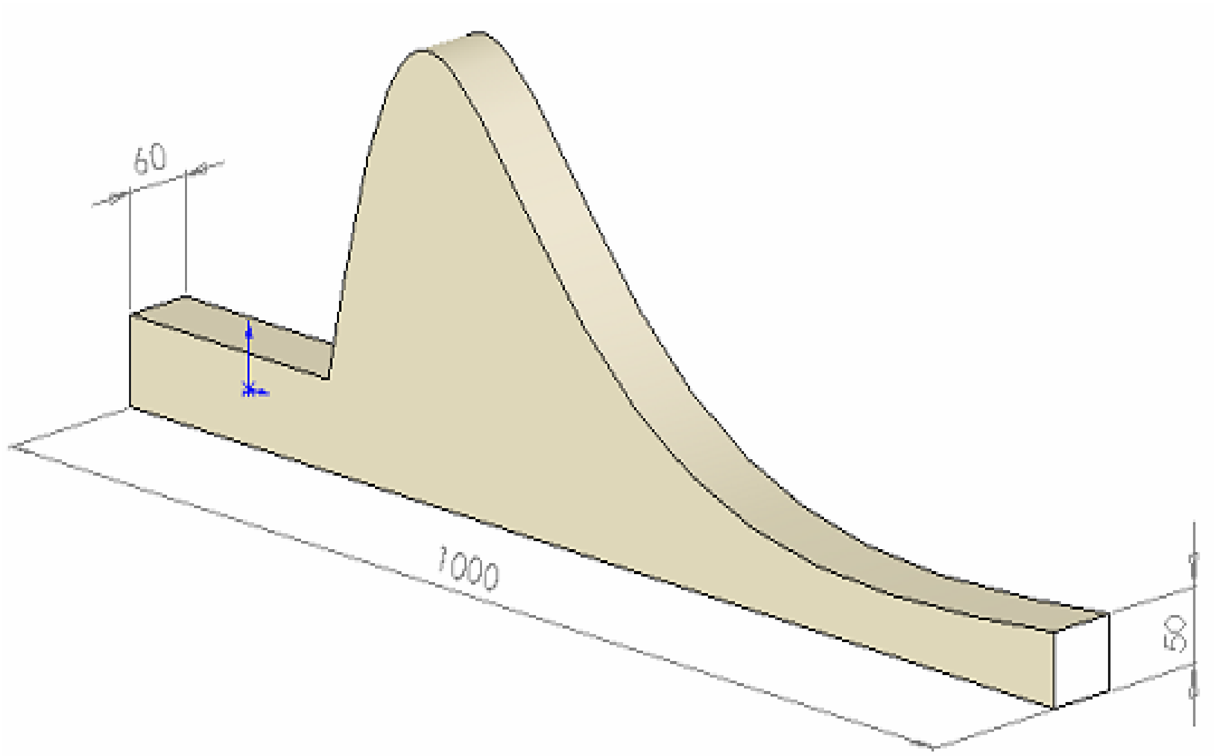
Left view:



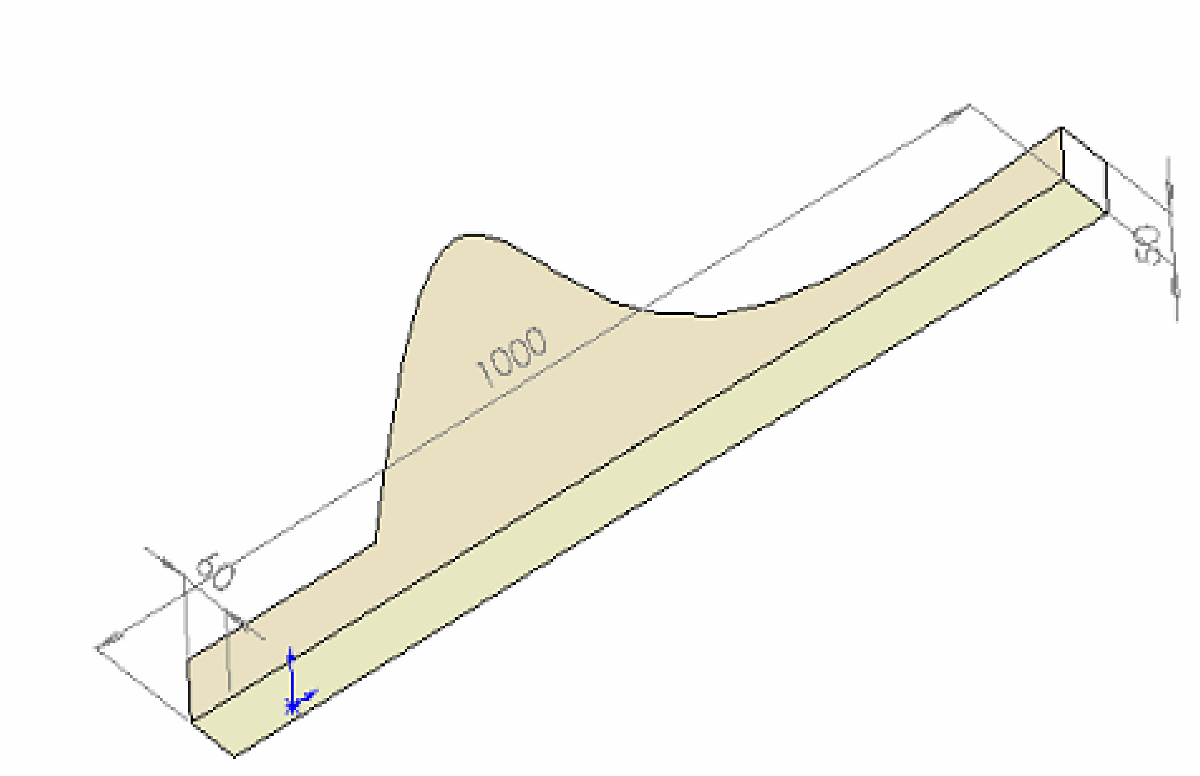
Isometric view:



Diametric view:

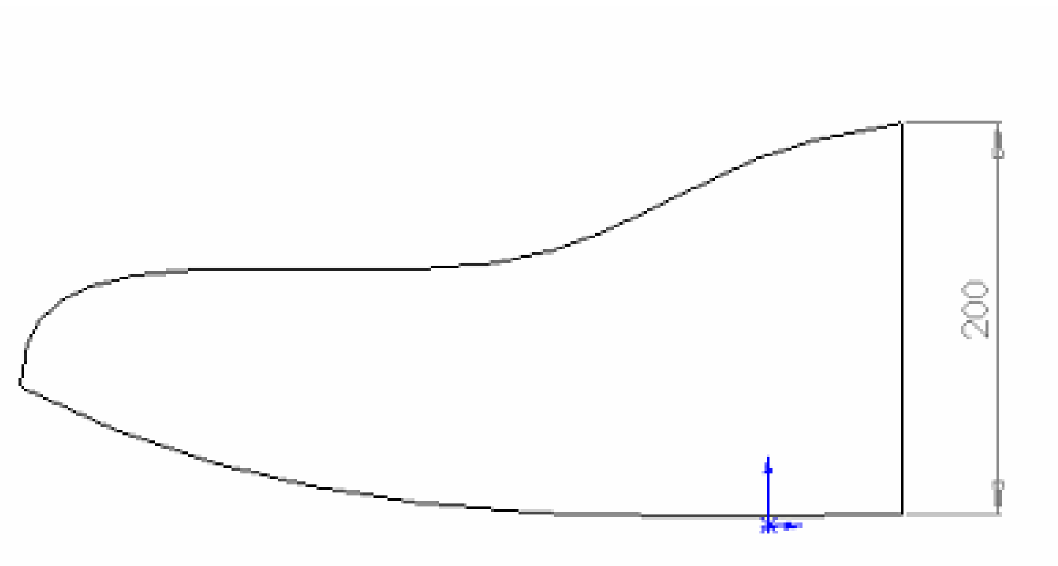


Trimetric view:



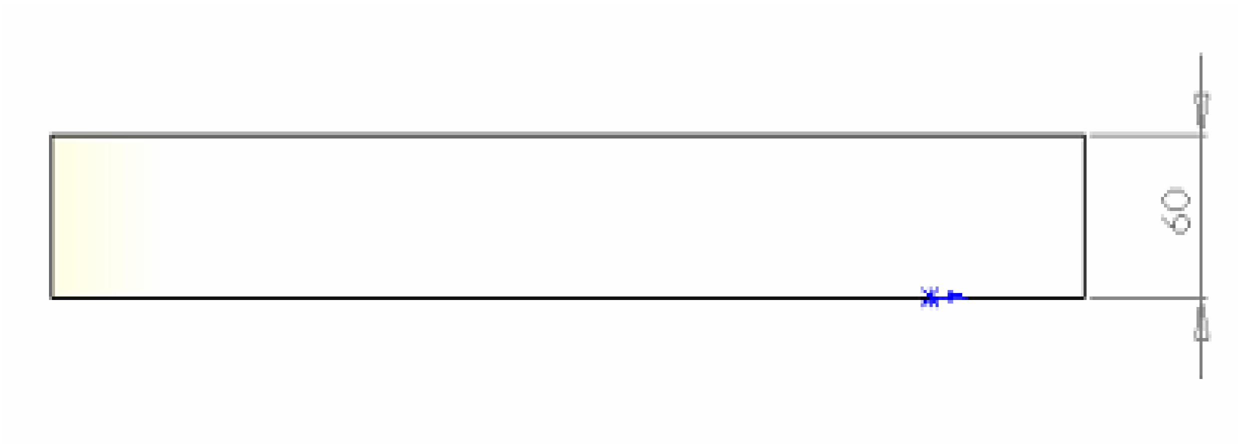
Part 3:

Front view:

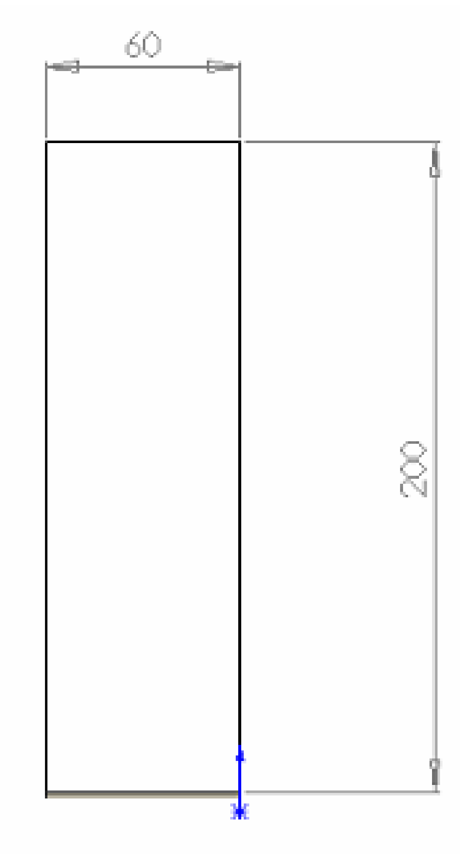


Top view:

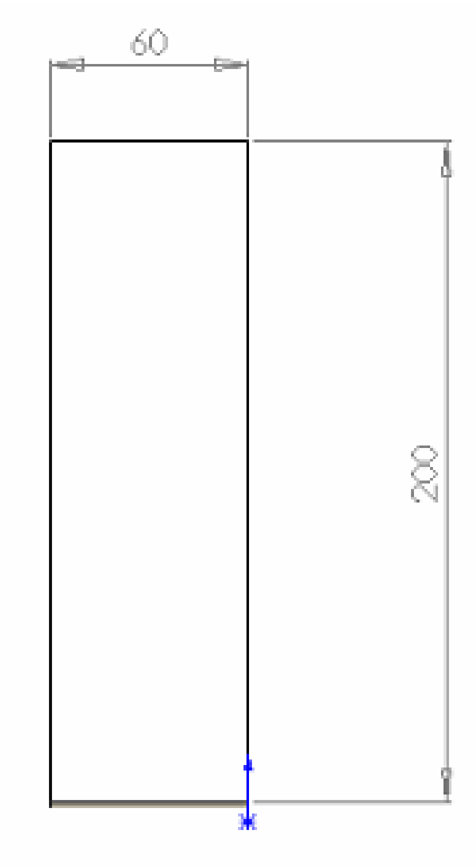
Bottom view:



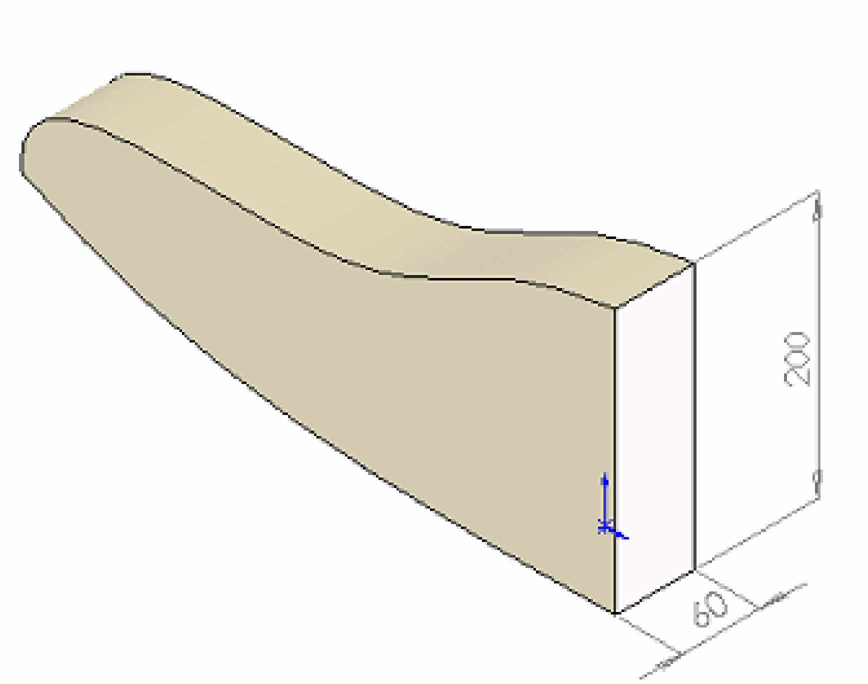
Right view:



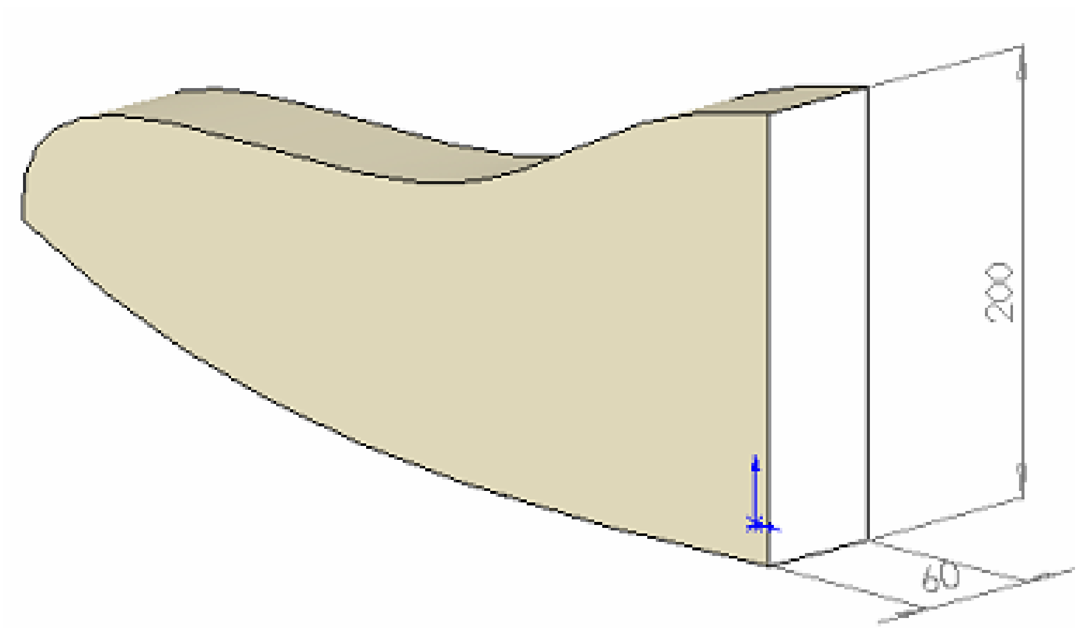
Left view:



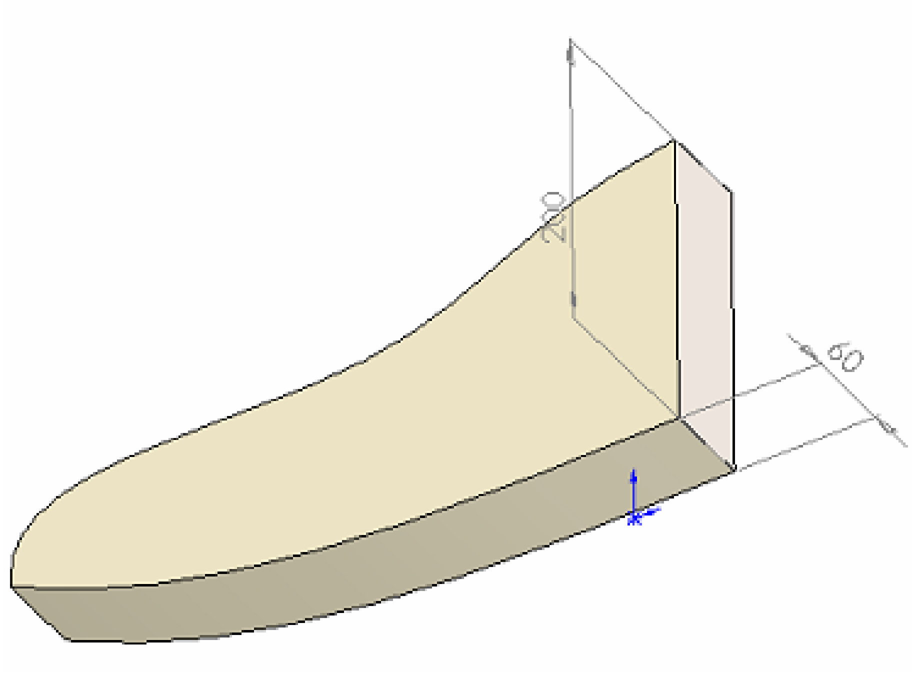
Isometric view:



Diametric view:

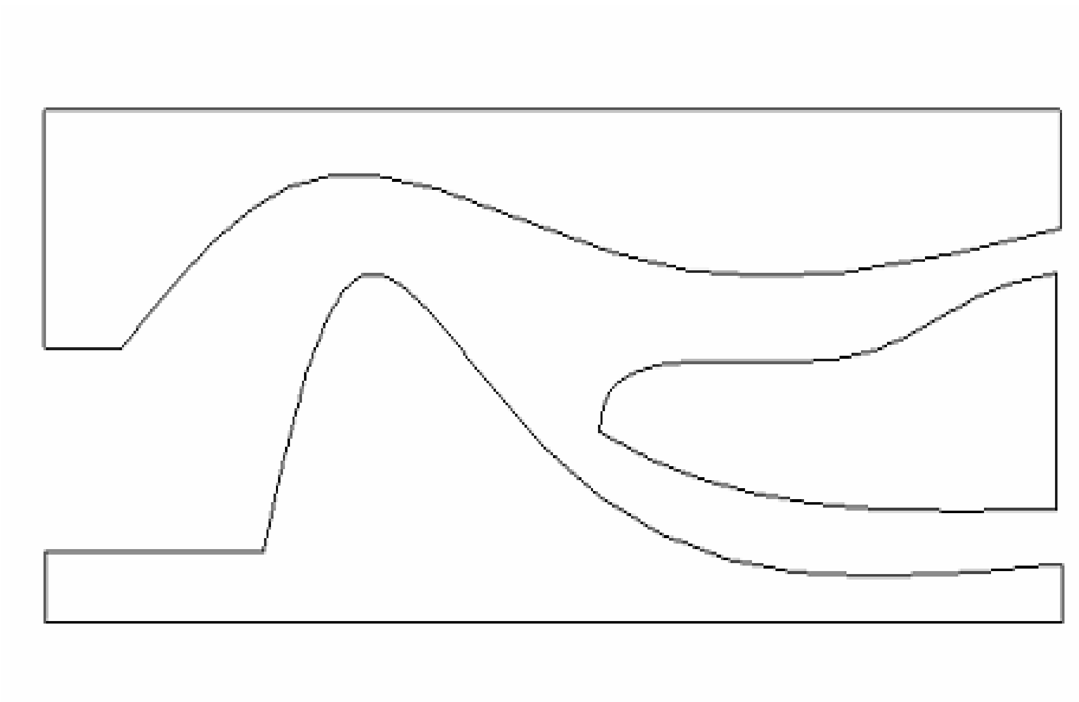


Trimetric view:

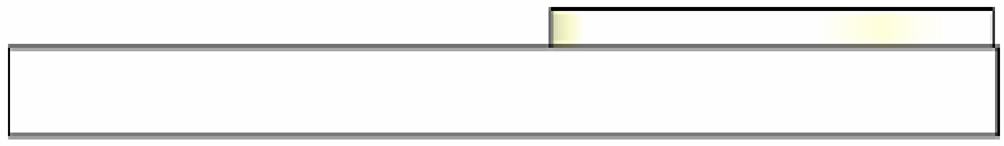


Assembly drawing:

Front view:

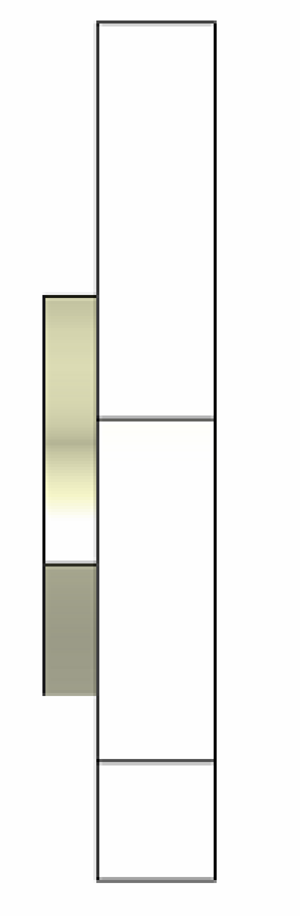


Top view:

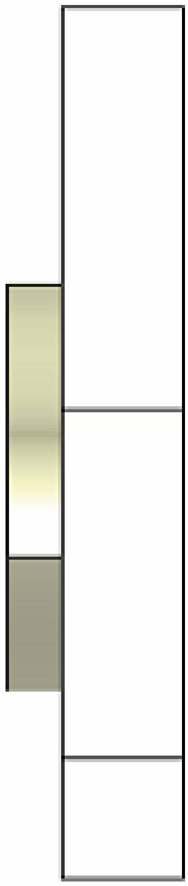


Bottom view:

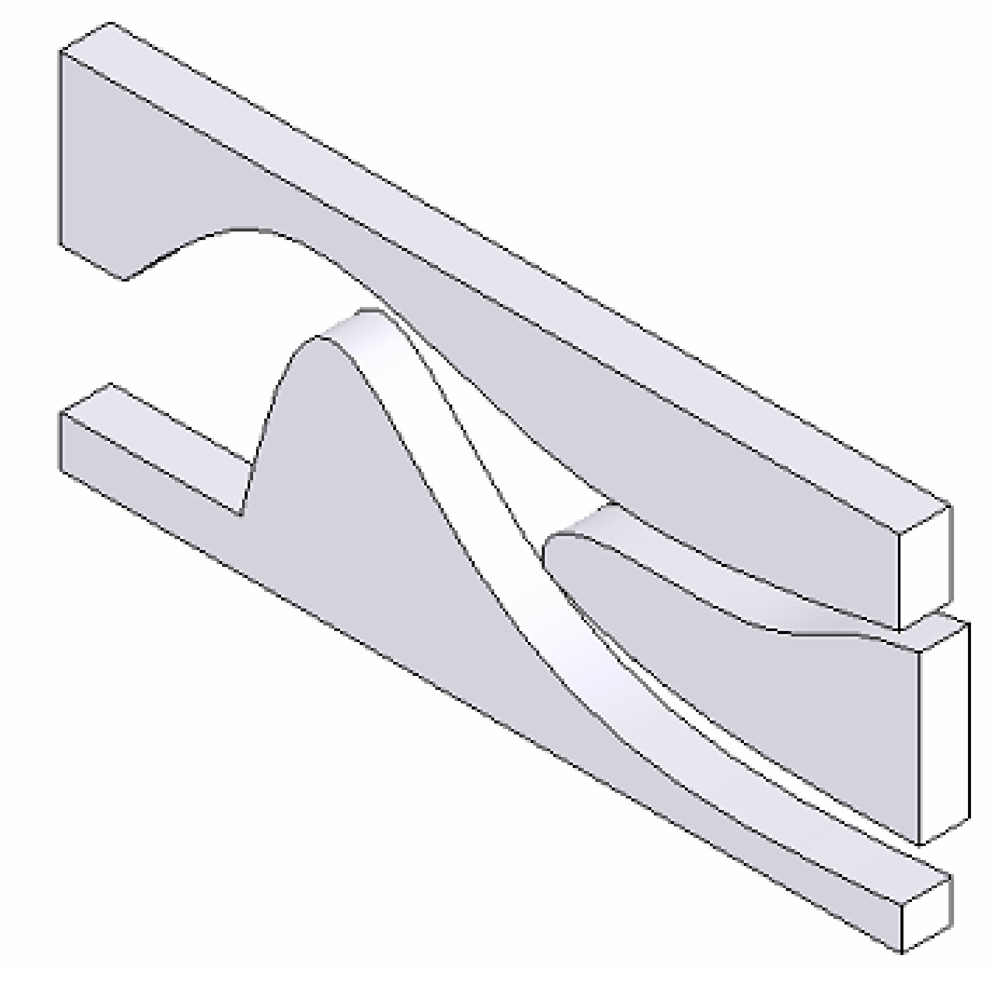
Right view:



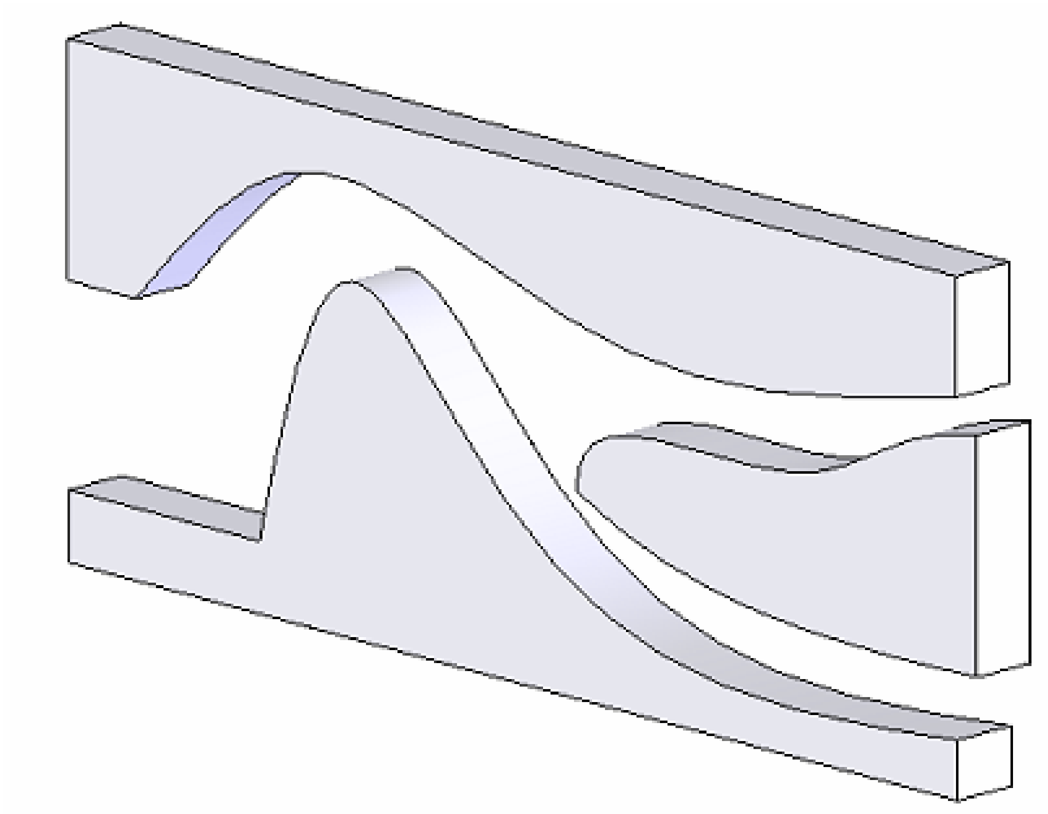
Left view:



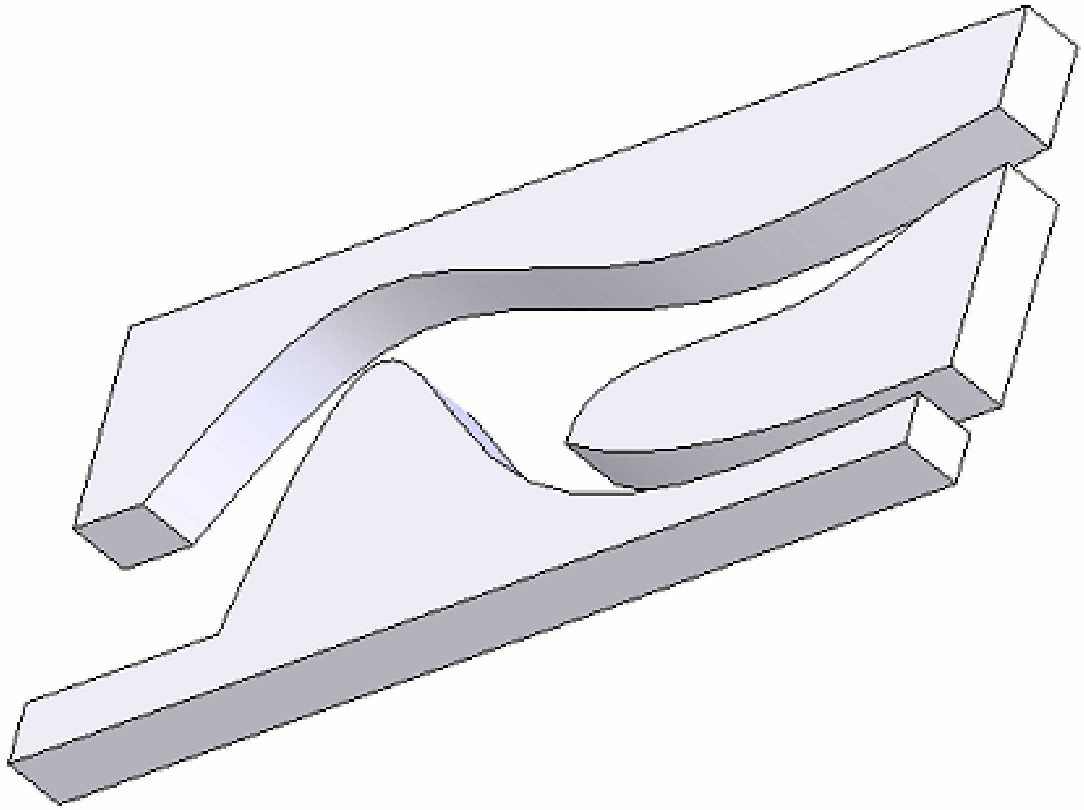
Isometric view:



Diametric view:



Trimetric view:



CALCULATION ANALYSIS:

- **ASSUMPTIONS:**

For simplicity and to make the calculations easy, the following assumptions may be raised:

1. Invisid flow
2. Incompressible flow
3. No particle or dust effect

In our calculations, we will consider two **cases:**

CASE 1:

In this case, we will consider a **crude separator with 2 x 90 bend** and calculate the velocity profile and the pressure loss in this duct.

Flow around a bend in a duct theory:

Inside the bend duct, the following equations are used to describe the flow:

$u = C/r$

where u is the stream velocity at radius r from the center of the curvature of the bend. Separation and secondary flow will be neglected.
The constant C may be found by applying the equation of continuity.

$Q = Ub(r_2 - r_1) = b \int_{r1}^{r2} u\, dr$

Where b is the width of the section of the duct. By appluing the integral and substituting in the previous equation we get:

$$C = U (r_2 - r_1)/(\ln (r_2 / r_1))$$

The corresponding pressure distribution may be found by assuming the Bernoulli equation between the updtream section and a section within the bend.

$$P_0 + ½ \rho U^2 = P + ½ \rho U^2$$

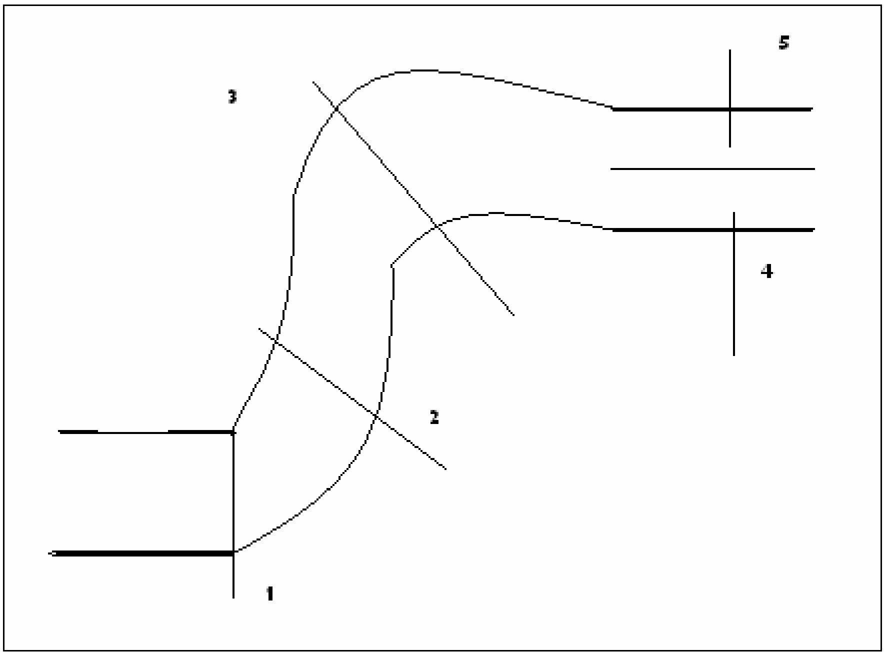
Where the P_0 is the static pressure and p is the pressure at the radius r in the bend.
It is convenient to express p in the form of a dimensionless pressure coefficient Cp.

$$Cp = (P_0 - P)/ \tfrac{1}{2} \rho U^2$$

From Bernoulli equation:

$$Cp = 1 - (u / U^2)$$

We will consider the following sketch to be our duct that is similar somehow to our air particle separator with different sections and different radiuses.



(Figure 12)

To perform our calculation analysis, We assume the following:

The velocity inlet = 20 cm/s

Station 2	Station 3	Station 4	Station 5
R_1 = 5 cm	R_1 = 3 cm	R_1 = 6 cm	R_1 = 6 cm
R_2 = 10 cm	R_2 = 8 cm	R_2 = 8 cm	R_2 = 11 cm

(Table 4)

- According to the equations that were presented in the theory part, the **velocity profile** is calculated by writing the following matlab program.

➢ **For cross section (perpendicular to the flow):**

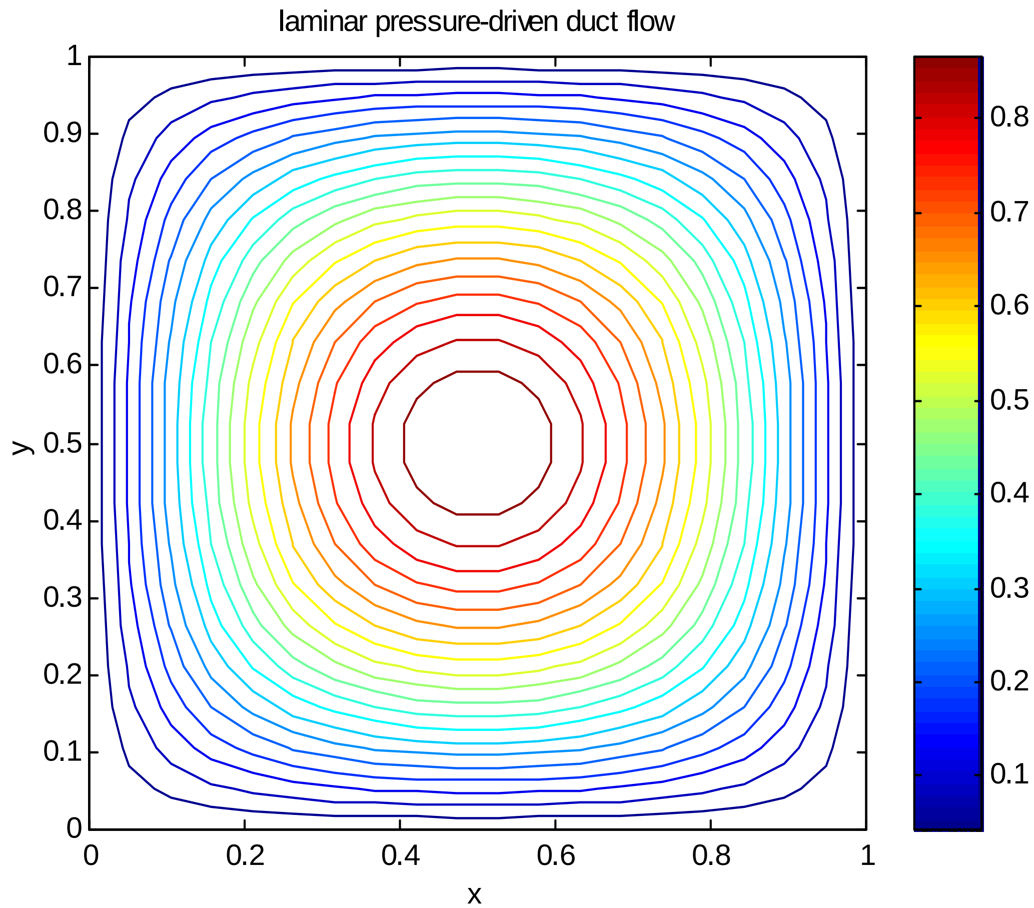
The M-file matlab program is:

```
%first set the truncation order of the expansion
N=10
resolution=20

%next, set the physical properties
G=-1;% pressure gradient
mu=1;%viscousity
Lx=1;%duct lengh in x direction
Ly=1;%duct lengh in y direction

%next, make  a 2D grid
x_vect=linspace(0,Lx,resolution);
y_vect=linspace(0,Ly,resolution);
[X,Y]=meshgrid(x_vect,y_vect);

%now we calculate the values of the velocity at each
point
VZ=0*X;

%sum over the orders of the expansion
var1=-16*G/mu/pi^4
for i=0:N
    m=2*i+1;
    for j=0:N
        n=2*i+1;
        factor1=var1/m/n/(m^2/Lx^2+n^2/Ly^2);
        %for every poni in the grid
        for p=1:size(X,1)
             for q=1:size(X,2)
                 x=X(p,q);
                 sin_mx=sin(m*pi*X(p,q)/Lx);
                 y=Y(p,q);
                 sin_ny=sin(m*pi*Y(p,q)/Ly);
                 VZ(p,q)=VZ(p,q)+...
                     factor1*sin_mx*sin_ny;
             end
         end
     end
 end
 figure;
 contour(X,Y,VZ,20)
 colorbar;
 title('laminar pressure-driven duct flow')
 xlabel('x');
 ylabel('y');
```

The output of the previous program is:

```
» %first set the truncation order of the expansion
N=10
resolution=20

%next, set the physical properties
G=-1;% pressure gradient
mu=1;%viscousity
Lx=1;%duct lengh in x direction
Ly=1;%duct lengh in y direction

%next, make  a 2D grid
x_vect=linspace(0,Lx,resolution);
y_vect=linspace(0,Ly,resolution);
[X,Y]=meshgrid(x_vect,y_vect);

%now we calculate the values of the velocity at each point
VZ=0*X;

%sum over the orders of the expansion
var1=-16*G/mu/pi^4
for i=0:N
    m=2*i+1;
    for j=0:N
        n=2*i+1;
        factor1=var1/m/n/(m^2/Lx^2+n^2/Ly^2);
        %for every poni in the grid
        for p=1:size(X,1)
             for q=1:size(X,2)
                 x=X(p,q);
                 sin_mx=sin(m*pi*X(p,q)/Lx);
                 y=Y(p,q);
                 sin_ny=sin(m*pi*Y(p,q)/Ly);
                 VZ(p,q)=VZ(p,q)+...
                     factor1*sin_mx*sin_ny;
             end
         end
     end
 end
 figure;
```

```
 contour(X,Y,VZ,20)
 colorbar;
 title('laminar pressure-driven duct flow')
 xlabel('x');
 ylabel('y');


N =

    10


resolution =

    20


var1 =

    0.1643
```


laminar pressure-driven duct flow
1
0.9
0.8
0.7
0.6
y 0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
x
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

(Figure 13)

Now taking each part of the duct.

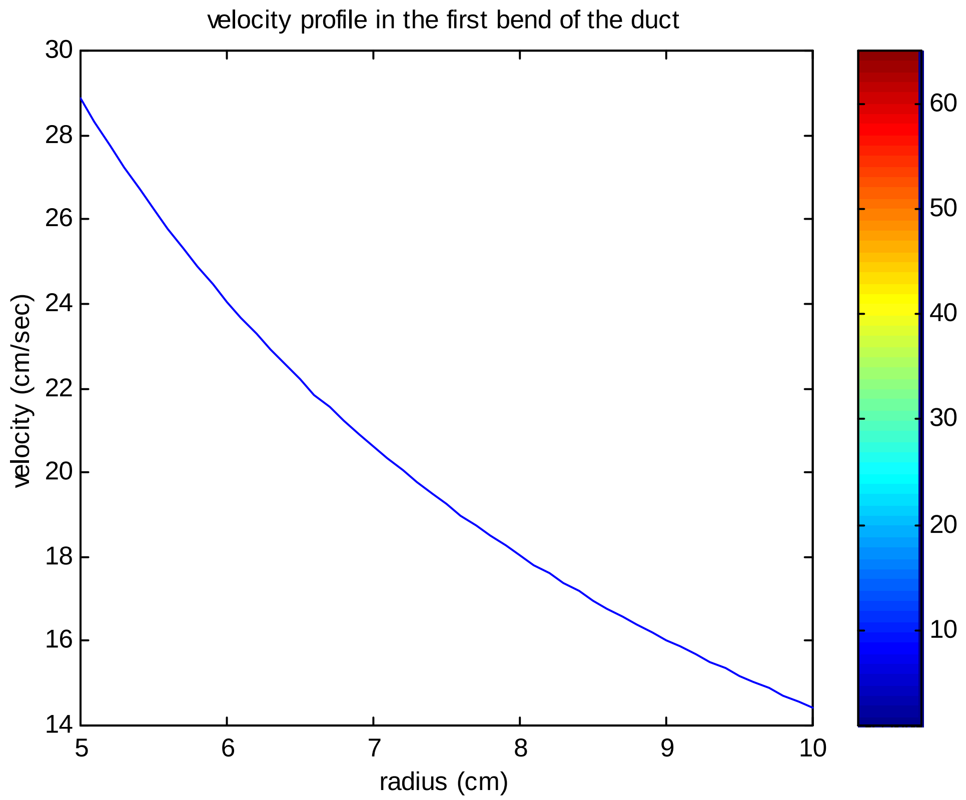
- **First portion:**

The M-file matlab program is:

```
%this is for the first bend(r11=5 cm and r12=10 cm)
initial_vel_1=20
r11=5
r12=10
r1=r11:0.1:r12;
velocity_prof_1=initial_vel_1.*(r12-
r11)./(r1*log(r12./r11));
plot(r1,velocity_prof_1);title('velocity profile in the
first bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» %this is for the first bend(r11=5 cm and r12=10 cm)
initial_vel_1=20
r11=5
r12=10
r1=r11:0.1:r12;
velocity_prof_1=initial_vel_1.*(r12-r11)./(r1*log(r12./r11));
plot(r1,velocity_prof_1);title('velocity profile in the first bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;

initial_vel_1 =

   20


r11 =

    5
r12 =

   10
```


velocity profile in the first bend of the duct
velocity (cm/sec)
radius (cm)
30
28
26
24
22
20
18
16
14
5
6
7
8
9
10
60
50
40
30
20
10

(Figure 14)

- **Second portion:**

The M-file matlab program is:

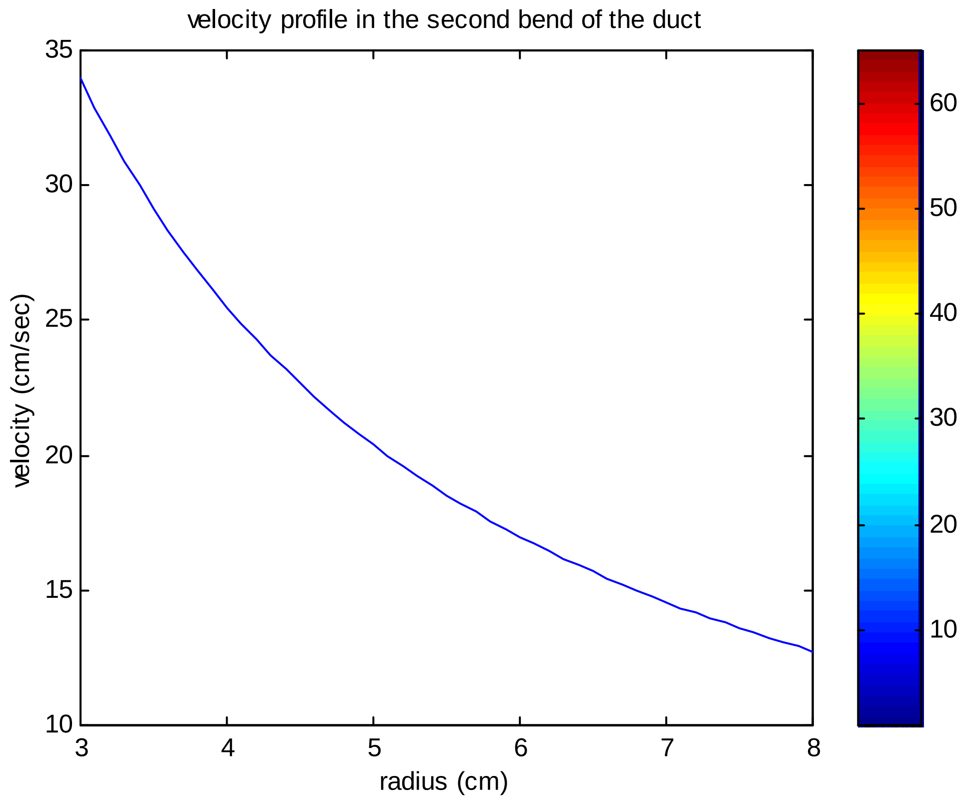
```
initial_vel_2=20

%this is for the second bend(r21=3 cm and r22=8 cm)
r21=3
r22=8
r2=r21:0.1:r22;
velocity_prof_2=initial_vel_2.*(r22-
r21)./(r2*log(r22./r21));
plot(r2,velocity_prof_2);title('velocity profile in the
second bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» initial_vel_2=20

%this is for the second bend(r21=3 cm and r22=8 cm)
r21=3
r22=8
r2=r21:0.1:r22;
velocity_prof_2=initial_vel_2.*(r22-r21)./(r2*log(r22./r21));
plot(r2,velocity_prof_2);title('velocity profile in the second bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;


initial_vel_2 =

   20


r21 =

    3
```

r22 =

8



(Figure 15)

- **Third portion:**

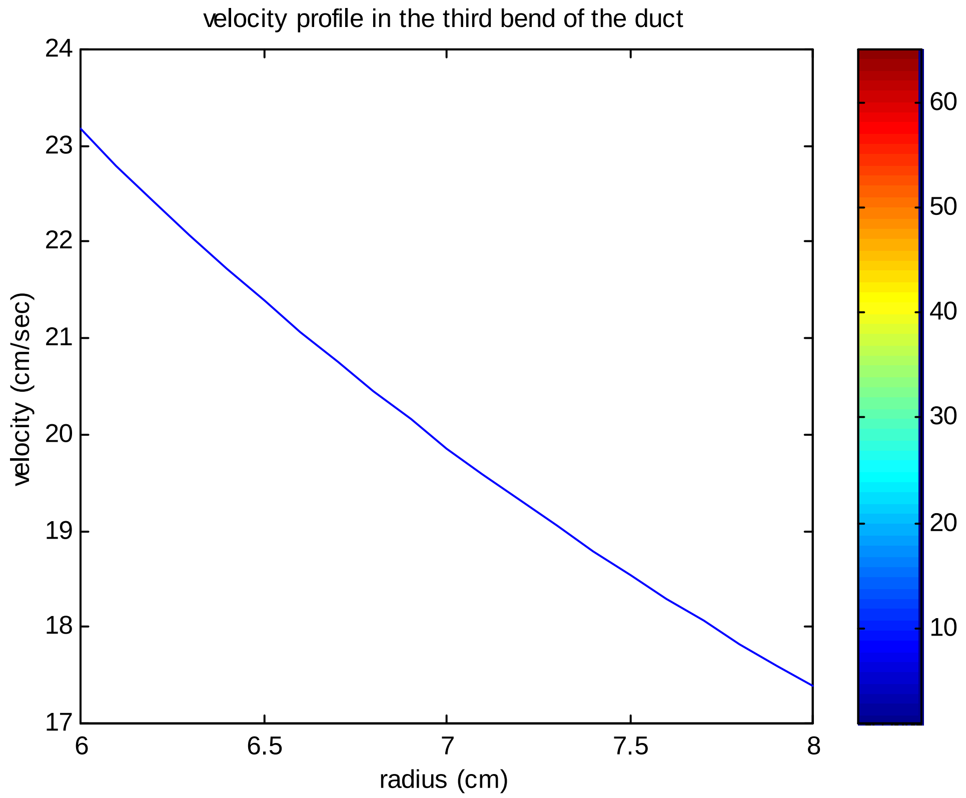
The M-file matlab program is:

```
initial_vel_3=20


%this is for the third bend(r31=6 cm and r32=8 cm)
r31=6
r32=8
r3=r31:0.1:r32;
velocity_prof_3=initial_vel_3.*(r32-
r31)./(r3*log(r32./r31));
plot(r3,velocity_prof_3);title('velocity profile in the
third bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» initial_vel_3=20


%this is for the third bend(r31=6 cm and r32=8 cm)
r31=6
r32=8
r3=r31:0.1:r32;
velocity_prof_3=initial_vel_3.*(r32-r31)./(r3*log(r32./r31));
plot(r3,velocity_prof_3);title('velocity profile in the third bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;


initial_vel_3 =

   20
```

```
r31 =

     6


r32 =

     8
```



(Figure 16)

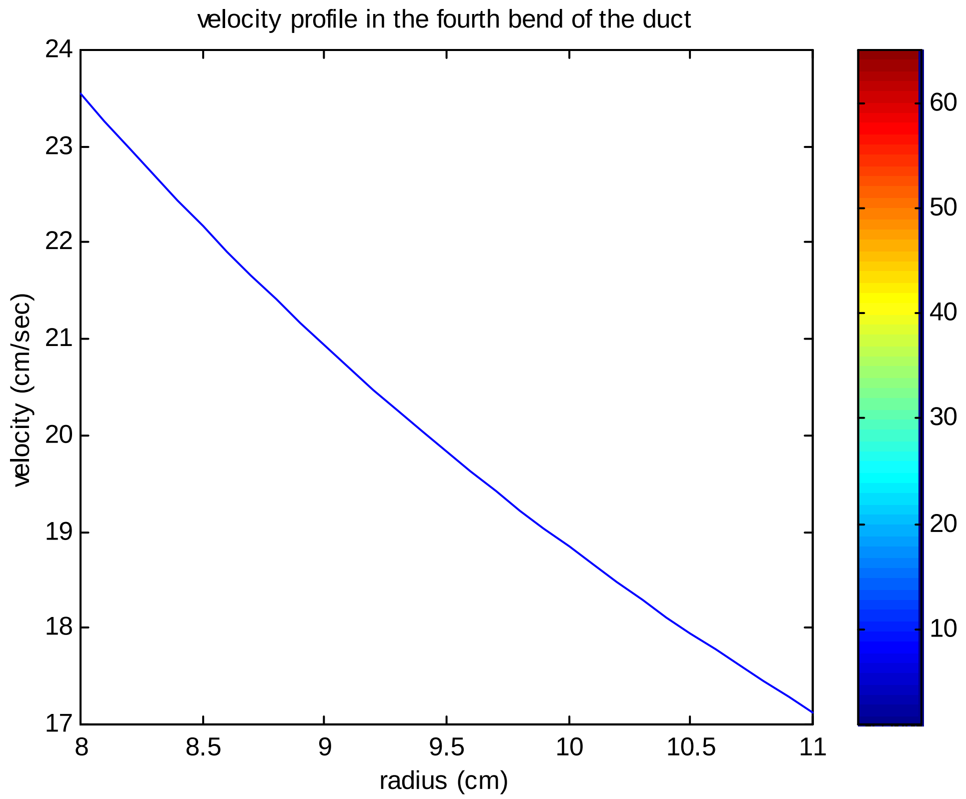
- **Fourth portion:**

The M-file matlab program is:

```
%this is for the fourth bend(r41=8 cm and r42=11 cm)

r41=8
r42=11
r4=r41:0.1:r42;
velocity_prof_4=initial_vel_3.*(r42-
r41)./(r4*log(r42./r41));
plot(r4,velocity_prof_4);title('velocity profile in the
fourth bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» %this is for the fourth bend(r41=8 cm and r42=11 cm)

r41=8
r42=11
r4=r41:0.1:r42;
velocity_prof_4=initial_vel_3.*(r42-r41)./(r4*log(r42./r41));
plot(r4,velocity_prof_4);title('velocity profile in the fourth bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;


r41 =

     8


r42 =

    11
```


velocity profile in the fourth bend of the duct
velocity (cm/sec)
radius (cm)
24
23
22
21
20
19
18
17
8
8.5
9
9.5
10
10.5
11
60
50
40
30
20
10

(Figure 17)

- **All portions in same plot:**

The M-file matlab program is:

```
%this is for the first bend(r11=5 cm and r12=10 cm)
initial_vel_1=20
r11=5
r12=10
r1=r11:0.1:r12;
velocity_prof_1=initial_vel_1.*(r12-
r11)./(r1*log(r12./r11));
subplot(411);plot(r1,velocity_prof_1);title('velocity
profile in the first bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
%
initial_vel_2=20


%this is for the second bend(r21=3 cm and r22=8 cm)

r21=3
r22=8
r2=r21:0.1:r22;
velocity_prof_2=initial_vel_2.*(r22-
r21)./(r2*log(r22./r21));
subplot(412);plot(r2,velocity_prof_2);title('velocity
profile in the second bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
%
initial_vel_3=20


%this is for the third bend(r31=6 cm and r32=8 cm)

r31=6
r32=8
r3=r31:0.1:r32;
velocity_prof_3=initial_vel_3.*(r32-
r31)./(r3*log(r32./r31));
subplot(413);plot(r3,velocity_prof_3);title('velocity
profile in the third bend of the duct')
xlabel('radius (cm)')
```

```
ylabel('velocity (cm/sec)')
colorbar;
%
initial_vel_3=20

%this is for the fourth bend(r41=8 cm and r42=11 cm)

r41=8
r42=11
r4=r41:0.1:r42;
velocity_prof_4=initial_vel_3.*(r42-
r41)./(r4*log(r42./r41));
subplot(414);plot(r4,velocity_prof_4);title('velocity
profile in the fourth bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

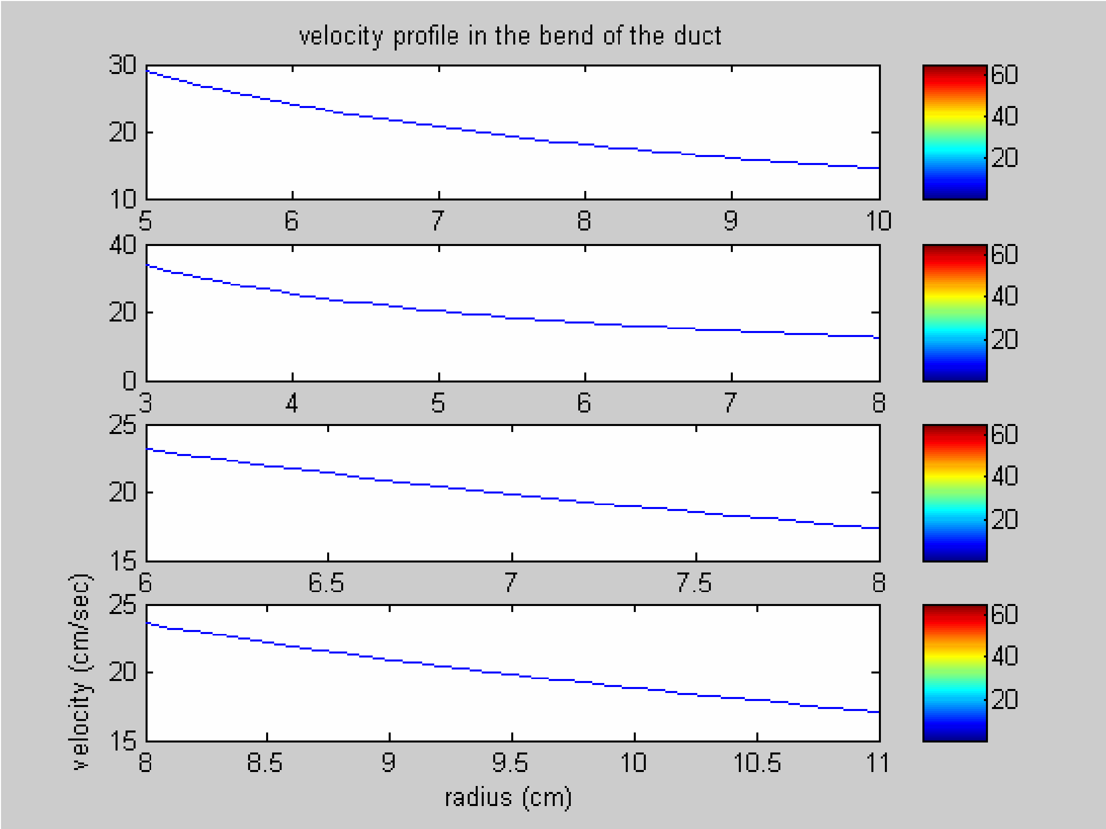
The output is:

```
» %this is for the first bend(r11=5 cm and r12=10 cm)
initial_vel_1=20
r11=5
r12=10
r1=r11:0.1:r12;
velocity_prof_1=initial_vel_1.*(r12-r11)./(r1*log(r12./r11));
subplot(411);plot(r1,velocity_prof_1);title('velocity profile in the first
bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
%
initial_vel_2=20


%this is for the second bend(r21=3 cm and r22=8 cm)

r21=3
r22=8
r2=r21:0.1:r22;
velocity_prof_2=initial_vel_2.*(r22-r21)./(r2*log(r22./r21));
subplot(412);plot(r2,velocity_prof_2);title('velocity profile in the second
bend of the duct')
xlabel('radius (cm)')
```

```
ylabel('velocity (cm/sec)')
colorbar;
%
initial_vel_3=20


%this is for the third bend(r31=6 cm and r32=8 cm)

r31=6
r32=8
r3=r31:0.1:r32;
velocity_prof_3=initial_vel_3.*(r32-r31)./(r3*log(r32./r31));
subplot(413);plot(r3,velocity_prof_3);title('velocity profile in the third
bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
%
initial_vel_3=20

%this is for the fourth bend(r41=8 cm and r42=11 cm)

r41=8
r42=11
r4=r41:0.1:r42;
velocity_prof_4=initial_vel_3.*(r42-r41)./(r4*log(r42./r41));
subplot(414);plot(r4,velocity_prof_4);title('velocity profile in the fourth
bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;

initial_vel_1 =

    20


r11 =

     5
```

```
r12 =

    10


initial_vel_2 =

    20


r21 =

     3


r22 =

     8


initial_vel_3 =

    20


r31 =

     6


r32 =

     8


initial_vel_3 =

    20


r41 =
```

```
    8


r42 =

   11
```



(Figure 18)

- According to the equations that were presented in the theory part, the **pressure coefficient** (**Cp**) is calculated by Excel program.

	U (cm/s)	R1	R2	r	u	Cp
First portion	20	5	10	7.5	19.23593	0.95191
Second portion	20	3	8	5.5	18.53719	0.953657
Third portion	20	6	8	7	19.8632	0.950342
Fourth portion	20	6	11	8.5	19.40936	0.951477

Table (5)

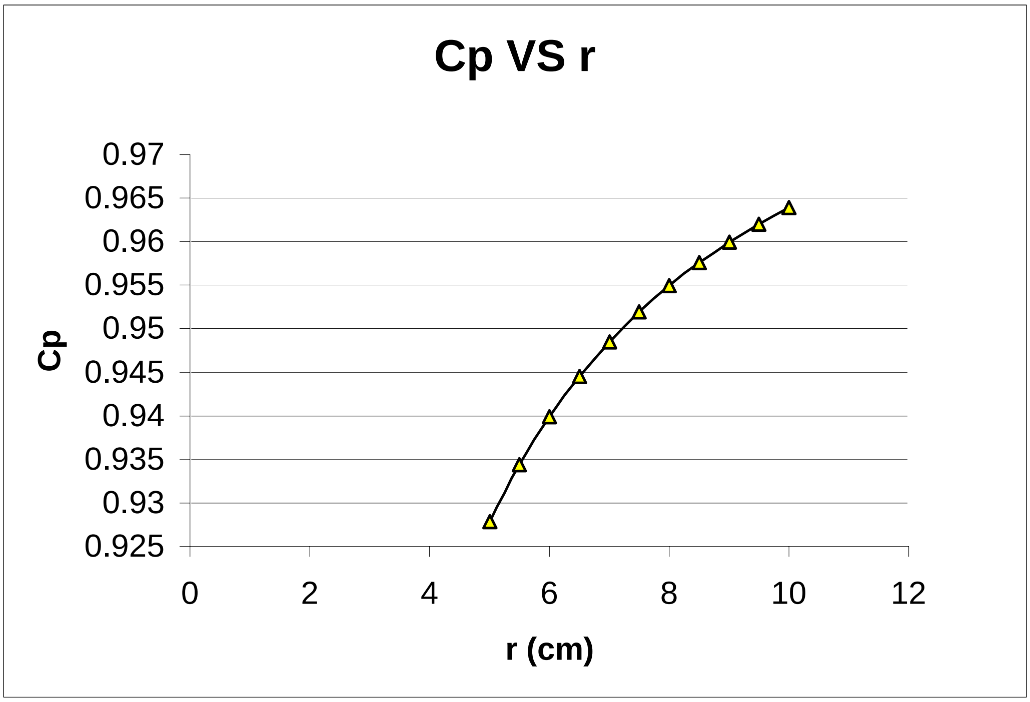
To draw the profile of the **pressure coefficient** (**Cp**), we use Excel program to do the following analysis:

We investigate the **pressure coefficient** against the radius of bend by taking increment of 0.5 cm.

First portion:

r	u/U	u/U^2	Cp
5	1.442695	0.072135	0.927865
5.5	1.311541	0.065577	0.934423
6	1.202246	0.060112	0.939888
6.5	1.109765	0.055488	0.944512
7	1.030496	0.051525	0.948475
7.5	0.961797	0.04809	0.95191
8	0.901684	0.045084	0.954916
8.5	0.848644	0.042432	0.957568
9	0.801497	0.040075	0.959925
9.5	0.759313	0.037966	0.962034
10	0.721348	0.036067	0.963933

(Table 6)

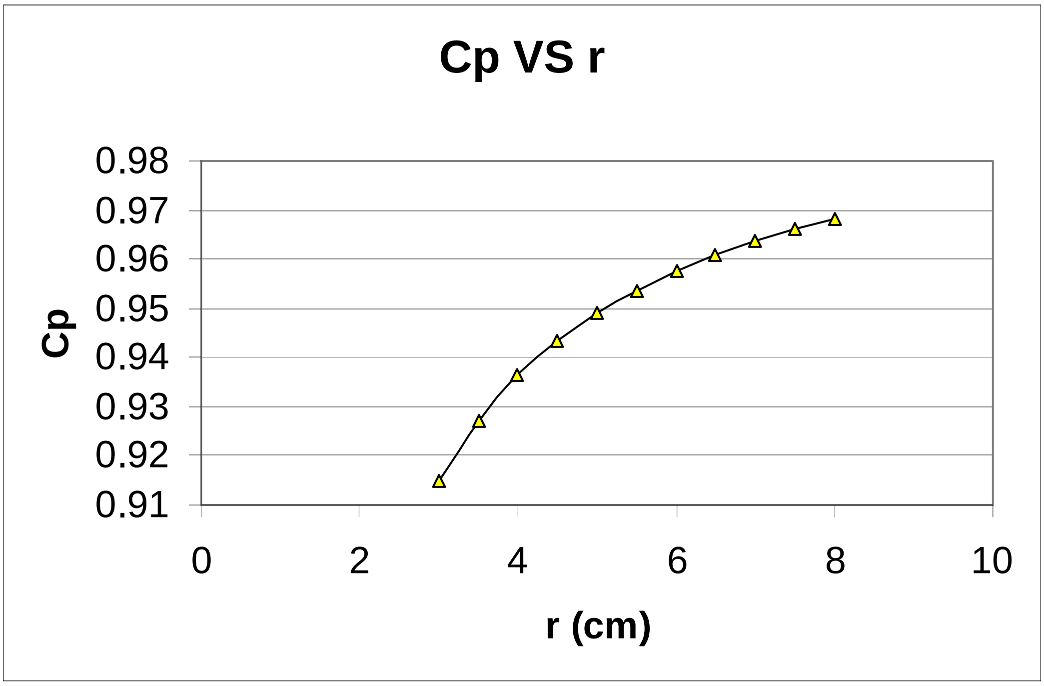


(Figure 19)

Second portion:

r	u/U	u/U^2	Cp
3	1.699242	0.084962	0.915038
3.5	1.456493	0.072825	0.927175
4	1.274432	0.063722	0.936278
4.5	1.132828	0.056641	0.943359
5	1.019545	0.050977	0.949023
5.5	0.926859	0.046343	0.953657
6	0.849621	0.042481	0.957519
6.5	0.784266	0.039213	0.960787
7	0.728247	0.036412	0.963588
7.5	0.679697	0.033985	0.966015
8	0.637216	0.031861	0.968139

(Table 7)

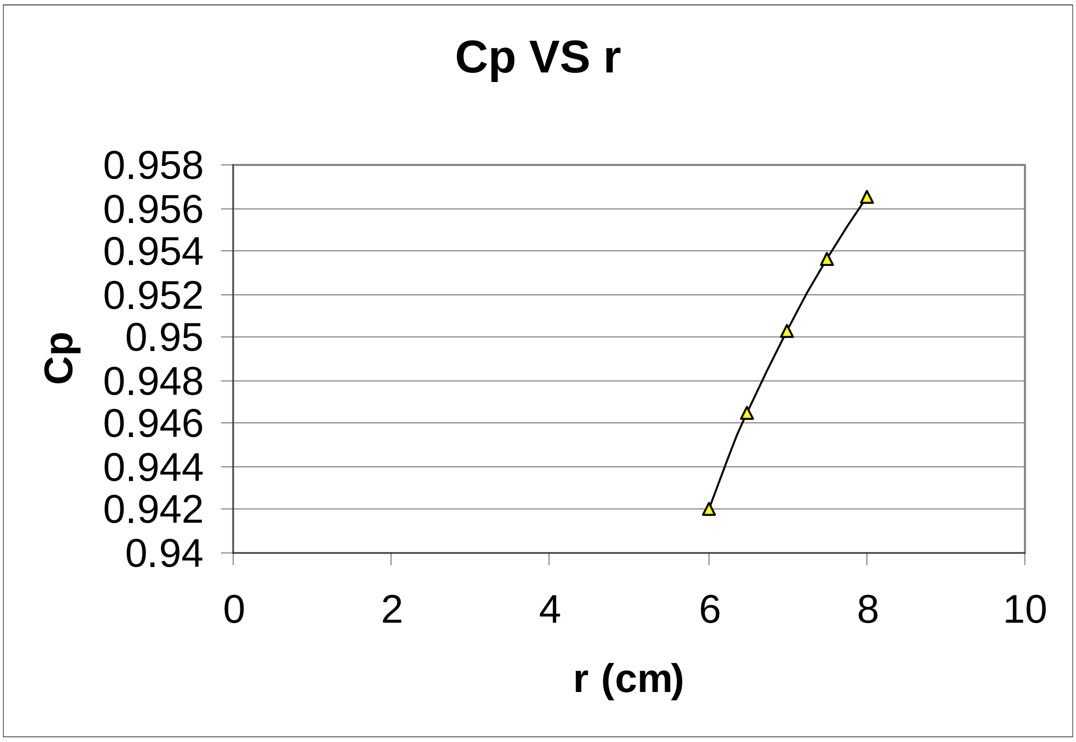


(Figure 20)

Third portion:

r	u/U	u/U^2	Cp
6	1.158686	0.057934	0.942066
6.5	1.069557	0.053478	0.946522
7	0.99316	0.049658	0.950342
7.5	0.926949	0.046347	0.953653
8	0.869015	0.043451	0.956549

(Table 8)

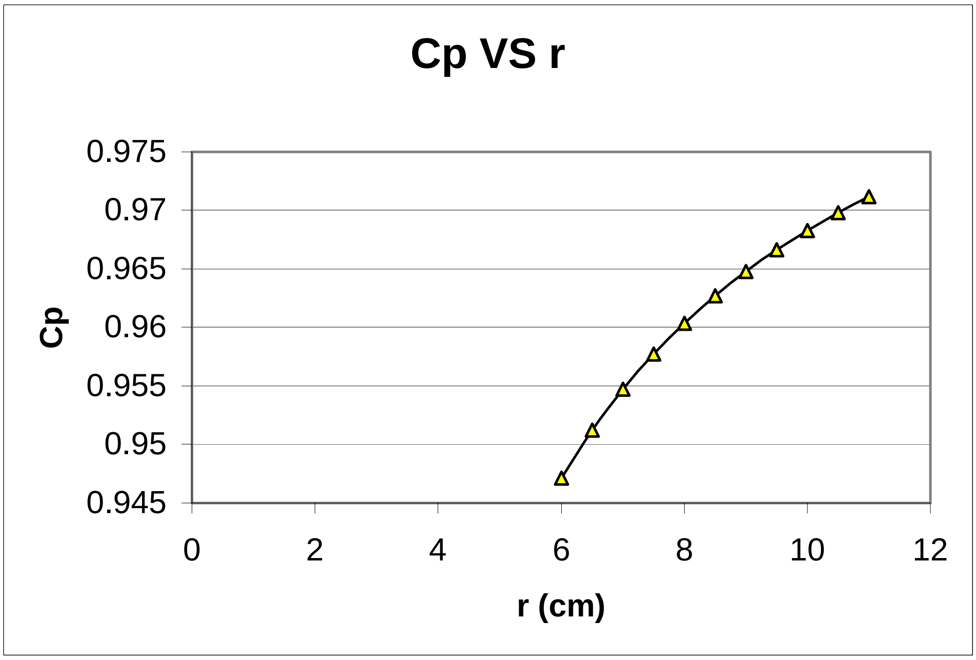


(Figure 21)

Fourth portion:

r	u/U	u/U^2	Cp
6	1.056916	0.052846	0.947154
6.5	0.975615	0.048781	0.951219
7	0.905928	0.045296	0.954704
7.5	0.845533	0.042277	0.957723
8	0.792687	0.039634	0.960366
8.5	0.746058	0.037303	0.962697
9	0.704611	0.035231	0.964769
9.5	0.667526	0.033376	0.966624
10	0.63415	0.031707	0.968293
10.5	0.603952	0.030198	0.969802
11	0.5765	0.028825	0.971175

(Table 9)



(Figure 22)

CASE 2:

In this case, we will consider **separators with parabolic curvature**.

Now we will concentrate on this type of separators, which is actually our particle separator that we already studied theoretically and generated its preliminary drawings.

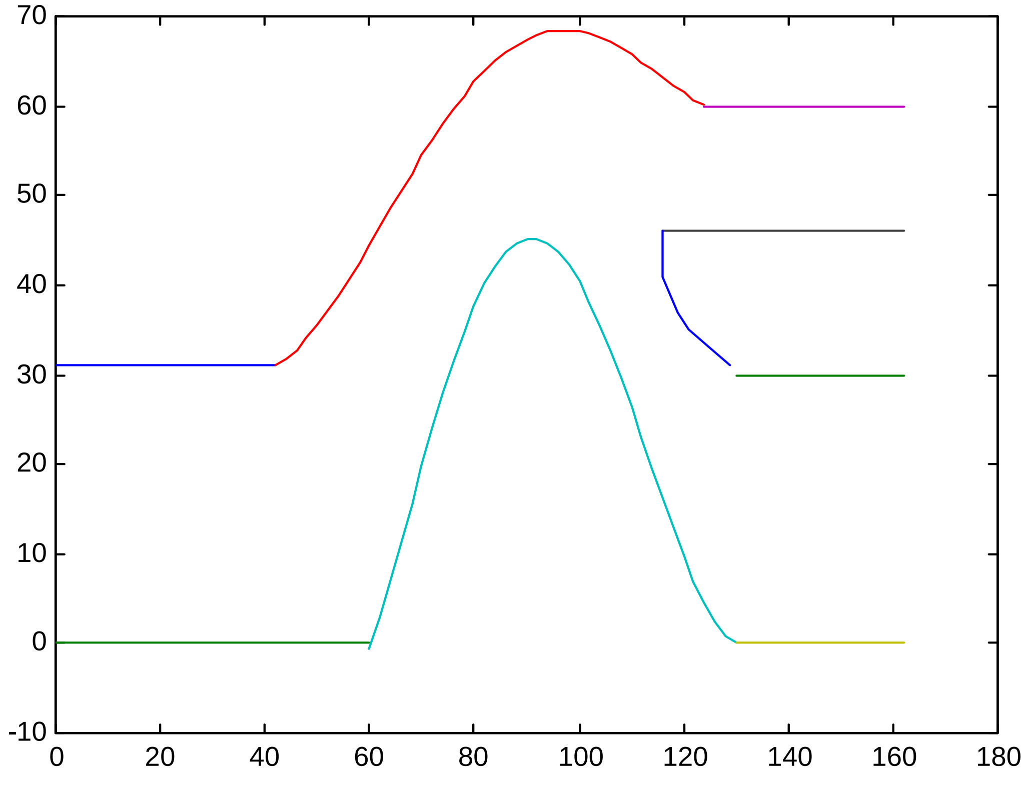
The front view drawing is needed to be graphed to study the flow inside its duct. Because of that, it was drawn by writing the following matlab program:

The M-file matlab program is:

```
x=0:2:42;
y2=[31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
31 31 31 31];
x2=0:2:60;
y1=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0];
x1=[43 51 61 65 67 71 77 81 85 91 95 97 103 105 107 109
111 115 119 121 123 125 131 137];
y3=[31 37 45 50 51 55 61 63 65 68 69 69 68 67 67 66 65 63
62 61 60 60 60 60];
y4=[0 0 0 5 10 19 33 41 45 49 47 45 39 35 29 23 19 15 11
9 8 7 3 0];
p=polyfit(x1,y3,5);
f=polyfit(x1,y4,5);


x77=42:2:124;
x88=60:2:130;
y7=polyval(p,x77);
y8=polyval(f,x88);
x3=124:2:162;
y5=[60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60 60
60 60];
x4=130:2:162;
```

```
y6=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
x11=116:2:162;
y11=[46 46 46 46 46 46 46 46 46 46 46 46 46 46 46 46 46
46 46 46 46 46 46 46];
x12=[116 116 119 121 125 129];
y12=[46 41 37 35 33 31];
z=polyfit(x12,y12,4);
x22=116:2:130;
y22=polyval(z,x22);
x33=130:2:162;
y33=[30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30];
plot(x,y2,x2,y1,x77,y7,x88,y8,x3,y5,x4,y6,x11,y11,x22,y22
,x33,y33);
plot(x,y2,x2,y1,x77,y7,x88,y8,x3,y5,x4,y6,x11,y11,x12,y12
,x33,y33);
axis('equal'),
x12=[116 116 119 121 125 129];
y12=[46 41 37 35 33 31];
plot(x,y2,x2,y1,x77,y7,x88,y8,x3,y5,x4,y6,x11,y11,x12,y12
,x33,y33);
```

The output is:



(Figure 23)

Method of solution:

➔Potential flow by finite difference:

We define all points inside the duct and the boundary conditions to find the flow rate profile as follows:

All values of the flow and boundary conditions are written in matlab program in this form:

```
% boundaryi conditions:
%delta x=5mm
%delta y=5mm
%inlet=6cm
%exit1=6cm
%exit2=3cm
%U-inlet=18 cm/s

% inlet conditions:
for j=2:13;
i=1;
p(j,i)=9*(j-1);
end


% exit conditions:
%exit # 1 (6 cm)
for j=2:13;
i=65;

p(j,i)=6*(j-1);
end



%exit # 2 (3 cm)
for j=19:25;
i=65;

p(j,i)=72+6*(j-19);
end
```

```
%lower wall boundary conditions:

        %a)straight sections:

for i=2:25;
j=1;

p(j,i)=0;
end

for i=55:64;
j=1;

p(j,i)=0;
end









for i=2:18;
  j=13;
  p(j,i)=108;
end


    for i=48:64;
  j=25;
  p(j,i)=108;
end

%that in hafh
for i=47:64;
  j=19;
  p(j,i)=72;
end

for i=50:64;
  j=13;
  p(j,i)=72;
end


          % b)boundary conditions at curved sections:
```

```
%lower curve

p(2,26)=0;p(3,27)=0;p(4,27)=0;
p(5,28)=0;
p(6,28)=0;
p(7,28)=0;
p(8,29)=0;
p(9,29)=0;
p(20,30)=0;


p(11,30)=0;p(12,31)=0;
p(13,31)=0;
p(14,31)=0;
p(15,32)=0;
p(16,33)=0;
p(17,33)=0;
p(18,34)=0;
p(19,35)=0;


p(20,36)=0;p(20,37)=0;
p(20,38)=0;
p(19,39)=0;
p(18,40)=0;
p(17,41)=0;
p(16,42)=0;
p(15,43)=0;
p(14,43)=0;


p(13,43)=0;
p(12,43)=0;
p(11,44)=0;
p(10,44)=0;
p(9,44)=0;
p(8,45)=0;
p(7,46)=0;
p(6,47)=0;
p(5,48)=0;

p(4,49)=0;
p(3,50)=0;
p(2,51)=0;
p(2,52)=0;
p(2,53)=0;
p(1,54)=0;


%upper curve:
```

```
p(14,19)=108;p(15,20)=108;p(16,21)=108;p(17,22)=108;p(17,23)=108;p(18,24)=108;
p(19,25)=108;
p(20,26)=108;p(21,27)=108;p(22,28)=108;p(23,29)=108;p(24,30)=108;p(25,31)=108;
p(25,32)=108;
p(26,33)=108;p(26,34)=108;p(27,35)=108;p(27,36)=108;p(28,37)=108;p(28,38)=108;
p(28,39)=108;
p(28,40)=108;p(28,41)=108;p(28,42)=108;p(27,43)=108;p(27,44)=108;p(26,45)=108;
p(26,46)=108;
p(26,47)=108;



%middle curve:

p(18,47)=72;p(17,47)=72;p(16,47)=72;p(15,48)=72;p(14,49)=72;
```

Now we want to find the stream function inside the duct to investigate different properties in further step later on, but at this stage only we consider the stream function in our analysis.

The M-file program is:

```
% boundaryi conditions:
%delta x=5mm
%delta y=5mm
%inlet=6cm
%exit1=6cm
%exit2=3cm
%U-inlet=18 cm/s

% inlet conditions:
for j=2:13;
i=1;
p(j,i)=9*(j-1);
end


% exit conditions:
%exit # 1 (6 cm)
for j=2:13;
i=65;

p(j,i)=6*(j-1);
```

```
end



%exit # 2 (3 cm)
for j=19:25;
i=65;

p(j,i)=72+6*(j-19);
end



%lower wall boundary conditions:

            %a)straight sections:

for i=2:25;
j=1;

p(j,i)=0;
end

for i=55:64;
j=1;

p(j,i)=0;
end









for i=2:18;
  j=13;
  p(j,i)=108;
end


    for i=48:64;
  j=25;
  p(j,i)=108;
end

%that in hafh
for i=47:64;
```

```
  j=19;
  p(j,i)=72;
end

for i=50:64;
  j=13;
  p(j,i)=72;
end


        % b)boundary conditions at curved sections:


%lower curve

p(2,26)=0;p(3,27)=0;p(4,27)=0;
p(5,28)=0;
p(6,28)=0;
p(7,28)=0;
p(8,29)=0;
p(9,29)=0;
p(20,30)=0;


p(11,30)=0;p(12,31)=0;
p(13,31)=0;
p(14,31)=0;
p(15,32)=0;
p(16,33)=0;
p(17,33)=0;
p(18,34)=0;
p(19,35)=0;


p(20,36)=0;p(20,37)=0;
p(20,38)=0;
p(19,39)=0;
p(18,40)=0;
p(17,41)=0;
p(16,42)=0;
p(15,43)=0;
p(14,43)=0;


p(13,43)=0;
p(12,43)=0;
p(11,44)=0;
p(10,44)=0;
p(9,44)=0;
p(8,45)=0;
```

```
p(7,46)=0;
p(6,47)=0;
p(5,48)=0;

p(4,49)=0;
p(3,50)=0;
p(2,51)=0;
p(2,52)=0;
p(2,53)=0;
p(1,54)=0;


%upper curve:

p(14,19)=108;p(15,20)=108;p(16,21)=108;p(17,22)=108;p(17,23)=108;p(18,24)=108;
p(19,25)=108;
p(20,26)=108;p(21,27)=108;p(22,28)=108;p(23,29)=108;p(24,30)=108;p(25,31)=108;
p(25,32)=108;
p(26,33)=108;p(26,34)=108;p(27,35)=108;p(27,36)=108;p(28,37)=108;p(28,38)=108;
p(28,39)=108;
p(28,40)=108;p(28,41)=108;p(28,42)=108;p(27,43)=108;p(27,44)=108;p(26,45)=108;
p(26,46)=108;
p(26,47)=108;



%middle curve:

p(18,47)=72;p(17,47)=72;p(16,47)=72;p(15,48)=72;p(14,49)=72;


    % the iteration formula is:
    %p(i,j)=p(i,j)+0.425*(p(i,j+1)+p(i,j-1)+p(i+1,j)+p(i-1,j)-4*p(i,j))



% solution:

for i=2:18;
  for j=2:12;
    p(j,i)=54

  end
end




i=19;
for j=2:13;
```

```
  p(j,i)=54

end



i=20;
for j=2:14;
  p(j,i)=54

end



i=21;
for j=2:15;
  p(j,i)=54

end



i=22;
for j=2:16;
  p(j,i)=54

end



i=23;

for j=2:16;
  p(j,i)=54

end


i=24;

for j=2:17;
  p(j,i)=54

end



i=25;

for j=2:18;
```

```
  p(j,i)=54

end



i=26;

for j=3:19;
  p(j,i)=54

end


i=27;

for j=5:20;
  p(j,i)=54

end


i=28;

for j=8:21;
  p(j,i)=54

end


i=29;

for j=10:22;
  p(j,i)=54

end


i=30;

for j=12:23;
  p(j,i)=54

end



i=31;

for j=15:24;
```

```
  p(j,i)=54

end


i=32;

for j=16:24;
  p(j,i)=54

end


i=33;

for j=18:25;
  p(j,i)=54

end


i=34;

for j=19:25;
  p(j,i)=54

end


i=35;

for j=20:26;
  p(j,i)=54

end


i=36;

for j=21:26;
  p(j,i)=54

end


i=37;

for j=21:27;
  p(j,i)=54
```

```
end


i=38;

for j=21:27;

   p(j,i)=54

end



i=39;

for j=20:27;
   p(j,i)=54

end


i=40;

for j=19:27;
   p(j,i)=54

end


i=41;

for j=18:27;
   p(j,i)=54

end


i=42;

for j=17:27;
   p(j,i)=54

end


i=43;

for j=16:26;
   p(j,i)=54
```

```
end


i=44;

for j=12:26;
  p(j,i)=54

end


i=45;

for j=9:25;
  p(j,i)=54

end


i=46;

for j=8:25;
  p(j,i)=54

end

% including the middle points:
i=47;

for j=7:15;
  p(j,i)=54

end


i=47;

for j=20:25;
  p(j,i)=54

end


i=48;

for j=6:14;
  p(j,i)=54

end
```

```
%*******************
 for i=48:64;
   for j=20:24;
      p(j,i)=54
   end

end




i=49;

for j=5:13;
   p(j,i)=54

end




i=50;

for j=4:12;
   p(j,i)=54

end


i=51;

for j=3:12;
   p(j,i)=54

end


i=52;

for j=3:12;
   p(j,i)=54

end


i=53;

for j=3:12;
   p(j,i)=54
```

```
end


i=54;

for j=2:12;
   p(j,i)=54

end




for i=55:64;
   for j=2:12;
      p(j,i)=54

   end
end




%solutions formula:
for iter=1:4;
 for i=2:18;
   for j=2:12;
      p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

   end
 end



% my adding

i=19;
for j=2:13;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=20;
for j=2:14;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))
```

```
end



i=21;
for j=2:15;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=22;
for j=2:16;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=23;

for j=2:16;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=24;

for j=2:17;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=25;

for j=2:18;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=26;

for j=3:19;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))
```

```
end


i=27;

for j=5:20;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=28;

for j=8:21;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=29;

for j=10:22;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=30;

for j=12:23;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=31;

for j=15:24;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=32;

for j=16:24;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))
```

```
end


i=33;

for j=18:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=34;

for j=19:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=35;

for j=20:26;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=36;

for j=21:26;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=37;

for j=21:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=38;

for j=21:27;

   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end
```

```
i=39;

for j=20:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=40;

for j=19:27;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=41;

for j=18:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=42;

for j=17:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=43;

for j=16:26;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=44;

for j=12:26;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end
```

```
i=45;

for j=9:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=46;

for j=8:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end

% including the middle points:
i=47;

for j=7:15;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=47;

for j=20:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=48;

for j=6:14;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end

%*******************
 for i=48:64;
   for j=20:24;
      p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))
   end

end
```

```
i=49;

for j=5:13;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end



i=50;

for j=4:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=51;

for j=3:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=52;

for j=3:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=53;

for j=3:12;
p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=54;

for j=2:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end
```

```
for i=55:64;
  for j=2:12;
    p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

  end
end



end
```

To show part of the program output:

Columns 43 through 56

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	4.5753	6.3281
0	0	0	0	0	0	0	0	4.1021	6.3166	8.1610	10.7176	12.4563
0	0	0	0	0	0	0	5.0054	9.0564	11.8933	14.1861	16.3434	18.0312
0	0	0	0	0	0	5.4692	9.9473	13.8684	16.8731	19.2362	21.2571	22.9125
0	0	0	0	0	5.7079	10.5852	14.6306	18.2896	21.2797	23.6257	25.5483	27.1440
0	0	0	0	5.4692	10.5852	15.0437	18.7908	22.1831	25.0831	27.4044	29.2661	30.7963
0	0	0	5.0054	9.9473	14.6306	18.7908	22.3362	25.5104	28.2922	30.5851	32.4234	33.9123
0	0	4.1021	9.0564	13.8684	18.2896	22.1831	26.0976	30.0358	33.7032	36.8193	39.2983	41.2164
0	0	6.3166	11.8933	16.8731	21.2797	28.0194	34.6540	39.7662	43.4532	46.1121	48.0500	49.4785
0	0	8.1610	14.1861	19.2362	30.0855	39.4171	45.8616	49.9028	52.5964	54.4921	55.8579	56.8523
0	4.5753	10.7176	16.3434	30.0660	41.8575	50.3621	57.3023	60.4017	62.0817	63.1530	63.8979	64.4253
0	6.5924	12.9056	27.3174	42.0540	51.8995	61.6706	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000
0	8.0970	21.3387	38.5075	50.6596	61.8390	72.0000	0	0	0	0	0	0
0	12.9735	30.5797	46.1708	60.4473	72.0000	0	0	0	0	0	0	0
7.3877	20.7466	37.8607	53.4187	72.0000	0	0	0	0	0	0	0	0
13.6789	27.6722	42.8490	56.3663	72.0000	0	0	0	0	0	0	0	0
19.4781	33.3108	44.7430	57.9087	72.0000	0	0	0	0	0	0	0	0
24.8797	36.2314	46.3587	57.7414	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000
29.2410	38.3000	46.3064	54.7058	62.3520	65.4029	66.8745	67.6979	68.1938	68.5016	68.6978	68.8281	68.9186
32.8037	40.0687	46.0921	52.2239	57.5216	62.5532	67.5139	71.9436	75.4789	78.0517	79.7980	80.9257	81.6307
36.2587	41.8358	48.1210	55.5113	69.7198	79.8461	84.7499	86.6579	87.4020	87.8353	88.2110	88.5530	88.8396
42.3256	56.0008	67.1207	84.6068	86.6316	88.4275	90.8132	92.8575	94.1919	94.9555	95.3898	95.6609	95.8496
81.5876	83.7649	92.0390	89.5453	95.3094	98.7629	100.0889	100.8511	101.4059	101.7943	102.0414	102.1914	102.2851
81.9687	91.0947	97.2580	100.6316	102.4656	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000
97.0692	101.4144	108.0000	108.0000	108.0000	0	0	0	0	0	0	0	0
108.0000	108.0000	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

CONCLUSION:

We considered two cases in our calculations:

Case 1: a **crude separator with 2 x 90 bend**

Case 2: **separators with parabolic curvature**.

By performing the analysis in the first case, it was found that this configuration has much pressure losses, which we don't want, and it was not what we looking for of perfect design.

In the second case, great results were found in calculating the stream function, which gives good result indication of the flow rate in the duct. Actually, the duct shape in the second case was the best one of different shapes we already studied by trying different parabolic functions and after large survey,

So we can conclude that this shape is the best to give us the best performance and less pressure losses.

For further study, we will concentrate in this shape and continue all studies according to it.

REFERENCES

1. http://public.cranfield.ac.uk/me/me063a/Jason.htm

2. http://www.fbodaily.com/cbd/archive/2000/06(June)/15-jun-2000/also1009.htm

3. http://www.ifrc.co.uk/TeaBagPaper/ teabagpaper.html

4. Journal of aircraft /AIAA/journal of fluid mecaniches

5. J.O. Hinze, “turbulance,” 2d ed., McGraw-Hill, New York, 1975.

6. H. Schlichting, “boundary layer theory,” 7th ed., McGraw-Hill, New York, 1979.

7. American concrete pipe assoc.Vienna, Va., 1980.

8. Fried, Irwan and I.E Idelchik. Flow resistyance: a design guide for engineers. Hemisphere Puplishing Corp., New York, 1989.

9.Chow, Ven Te. Open Channel Hydraulics. McGraw-Hill, New York, 1959.

10. Streeter, V L. (ed) *Handbook of fluid mechanics*. McGraw-Hill, New York, 1961

11. Streeter, V L., and E. B. Wyle. *Fluid mechanics*, 7th ed. McGraw-Hill, New York, 1979

12. Whire, F. M. Viscous Fluid Flow. McGraw-Hill, New York, (1991).

13. Yen, B. C. (ed) Channel Flow Resistance: Centennial of Manning’s Formula. Water Resourses Puplications, Littlton, CO, 1992.

14. Crowe. R. Engineering Fluid Mechanics, 7 th .1996.

15. Frank M. White. Fluid Mechanics, 2 th . McGraw-Hill,1986.

APPENDICES

Appendix A: (matlab programs)

➢ **For cross section (perpendicular to the flow):**

The M-file matlab program is:

```
%first set the truncation order of the expansion
N=10
resolution=20

%next, set the physical properties
G=-1;% pressure gradient
mu=1;%viscousity
Lx=1;%duct lengh in x direction
Ly=1;%duct lengh in y direction

%next, make  a 2D grid
x_vect=linspace(0,Lx,resolution);
y_vect=linspace(0,Ly,resolution);
[X,Y]=meshgrid(x_vect,y_vect);

%now we calculate the values of the velocity at each
point
VZ=0*X;

%sum over the orders of the expansion
var1=-16*G/mu/pi^4
for i=0:N
    m=2*i+1;
    for j=0:N
        n=2*i+1;
        factor1=var1/m/n/(m^2/Lx^2+n^2/Ly^2);
        %for every poni in the grid
        for p=1:size(X,1)
             for q=1:size(X,2)
                 x=X(p,q);
                 sin_mx=sin(m*pi*X(p,q)/Lx);
                 y=Y(p,q);
                 sin_ny=sin(m*pi*Y(p,q)/Ly);
                 VZ(p,q)=VZ(p,q)+...
                     factor1*sin_mx*sin_ny;
             end
         end
```

```
    end
end
figure;
contour(X,Y,VZ,20)
colorbar;
title('laminar pressure-driven duct flow')
xlabel('x');
ylabel('y');
```

The output of the previous program is:

```
» %first set the truncation order of the expansion
N=10
resolution=20

%next, set the physical properties
G=-1;% pressure gradient
mu=1;%viscousity
Lx=1;%duct lengh in x direction
Ly=1;%duct lengh in y direction

%next, make  a 2D grid
x_vect=linspace(0,Lx,resolution);
y_vect=linspace(0,Ly,resolution);
[X,Y]=meshgrid(x_vect,y_vect);

%now we calculate the values of the velocity at each point
VZ=0*X;

%sum over the orders of the expansion
var1=-16*G/mu/pi^4
for i=0:N
   m=2*i+1;
   for j=0:N
      n=2*i+1;
      factor1=var1/m/n/(m^2/Lx^2+n^2/Ly^2);
      %for every poni in the grid
      for p=1:size(X,1)
          for q=1:size(X,2)
             x=X(p,q);
             sin_mx=sin(m*pi*X(p,q)/Lx);
             y=Y(p,q);
             sin_ny=sin(m*pi*Y(p,q)/Ly);
             VZ(p,q)=VZ(p,q)+...
```

```
                     factor1*sin_mx*sin_ny;
             end
         end
     end
 end
 figure;
 contour(X,Y,VZ,20)
 colorbar;
 title('laminar pressure-driven duct flow')
 xlabel('x');
 ylabel('y');


N =

    10


resolution =

    20


var1 =

    0.1643
```

Now taking each part of the duct.

- **First portion:**

The M-file matlab program is:

```
%this is for the first bend(r11=5 cm and r12=10 cm)
initial_vel_1=20
r11=5
r12=10
r1=r11:0.1:r12;
velocity_prof_1=initial_vel_1.*(r12-
r11)./(r1*log(r12./r11));
plot(r1,velocity_prof_1);title('velocity profile in the
first bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» %this is for the first bend(r11=5 cm and r12=10 cm)
initial_vel_1=20
r11=5
r12=10
r1=r11:0.1:r12;
velocity_prof_1=initial_vel_1.*(r12-r11)./(r1*log(r12./r11));
plot(r1,velocity_prof_1);title('velocity profile in the first bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;

initial_vel_1 =

   20


r11 =
```

```
    5
r12 =

   10
```

- **Second portion:**

The M-file matlab program is:

```
initial_vel_2=20

%this is for the second bend(r21=3 cm and r22=8 cm)
r21=3
r22=8
r2=r21:0.1:r22;
velocity_prof_2=initial_vel_2.*(r22-
r21)./(r2*log(r22./r21));
plot(r2,velocity_prof_2);title('velocity profile in the
second bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» initial_vel_2=20

%this is for the second bend(r21=3 cm and r22=8 cm)
r21=3
r22=8
r2=r21:0.1:r22;
velocity_prof_2=initial_vel_2.*(r22-r21)./(r2*log(r22./r21));
plot(r2,velocity_prof_2);title('velocity profile in the second bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;


initial_vel_2 =
```

```
   20


r21 =

    3


r22 =

    8
```

➢ **Third portion:**

The M-file matlab program is:

```
initial_vel_3=20


%this is for the third bend(r31=6 cm and r32=8 cm)
r31=6
r32=8
r3=r31:0.1:r32;
velocity_prof_3=initial_vel_3.*(r32-
r31)./(r3*log(r32./r31));
plot(r3,velocity_prof_3);title('velocity profile in the
third bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» initial_vel_3=20


%this is for the third bend(r31=6 cm and r32=8 cm)
r31=6
r32=8
r3=r31:0.1:r32;
velocity_prof_3=initial_vel_3.*(r32-r31)./(r3*log(r32./r31));
```

```
plot(r3,velocity_prof_3);title('velocity profile in the third bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;


initial_vel_3 =

    20


r31 =

     6


r32 =

     8
```

- **Fourth portion:**

The M-file matlab program is:

```
%this is for the fourth bend(r41=8 cm and r42=11 cm)

r41=8
r42=11
r4=r41:0.1:r42;
velocity_prof_4=initial_vel_3.*(r42-r41)./(r4*log(r42./r41));
plot(r4,velocity_prof_4);title('velocity profile in the fourth bend of the duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;
```

The output is:

```
» %this is for the fourth bend(r41=8 cm and r42=11 cm)
```

```
r41=8
r42=11
r4=r41:0.1:r42;
velocity_prof_4=initial_vel_3.*(r42-r41)./(r4*log(r42./r41));
plot(r4,velocity_prof_4);title('velocity profile in the fourth bend of the
duct')
xlabel('radius (cm)')
ylabel('velocity (cm/sec)')
colorbar;


r41 =

     8


r42 =

    11
```

The M-file program for **stream function (case 2) inside duct:**

```
% boundaryi conditions:
%delta x=5mm
%delta y=5mm
%inlet=6cm
%exit1=6cm
%exit2=3cm
%U-inlet=18 cm/s

% inlet conditions:
for j=2:13;
i=1;
p(j,i)=9*(j-1);
end


% exit conditions:
%exit # 1 (6 cm)
for j=2:13;
i=65;
```

```
p(j,i)=6*(j-1);
end



%exit # 2 (3 cm)
for j=19:25;
i=65;

p(j,i)=72+6*(j-19);
end



%lower wall boundary conditions:

            %a)straight sections:

for i=2:25;
j=1;

p(j,i)=0;
end

for i=55:64;
j=1;

p(j,i)=0;
end








for i=2:18;
  j=13;
  p(j,i)=108;
end


    for i=48:64;
  j=25;
  p(j,i)=108;
end
```

```
%that in hafh
for i=47:64;
  j=19;
  p(j,i)=72;
end

for i=50:64;
  j=13;
  p(j,i)=72;
end


                % b)boundary conditions at curved
sections:


%lower curve

p(2,26)=0;p(3,27)=0;p(4,27)=0;
p(5,28)=0;
p(6,28)=0;
p(7,28)=0;
p(8,29)=0;
p(9,29)=0;
p(20,30)=0;


p(11,30)=0;p(12,31)=0;
p(13,31)=0;
p(14,31)=0;
p(15,32)=0;
p(16,33)=0;
p(17,33)=0;
p(18,34)=0;
p(19,35)=0;


p(20,36)=0;p(20,37)=0;
p(20,38)=0;
p(19,39)=0;
p(18,40)=0;
p(17,41)=0;
p(16,42)=0;
p(15,43)=0;
p(14,43)=0;


p(13,43)=0;
p(12,43)=0;
p(11,44)=0;
```

```
p(10,44)=0;
p(9,44)=0;
p(8,45)=0;
p(7,46)=0;
p(6,47)=0;
p(5,48)=0;

p(4,49)=0;
p(3,50)=0;
p(2,51)=0;
p(2,52)=0;
p(2,53)=0;
p(1,54)=0;


%upper curve:

p(14,19)=108;p(15,20)=108;p(16,21)=108;p(17,22)=108;p(17,
23)=108;p(18,24)=108;p(19,25)=108;
p(20,26)=108;p(21,27)=108;p(22,28)=108;p(23,29)=108;p(24,
30)=108;p(25,31)=108;p(25,32)=108;
p(26,33)=108;p(26,34)=108;p(27,35)=108;p(27,36)=108;p(28,
37)=108;p(28,38)=108;p(28,39)=108;
p(28,40)=108;p(28,41)=108;p(28,42)=108;p(27,43)=108;p(27,
44)=108;p(26,45)=108;p(26,46)=108;
p(26,47)=108;



%middle curve:

p(18,47)=72;p(17,47)=72;p(16,47)=72;p(15,48)=72;p(14,49)=
72;


      % the iteration formula is:
      %p(i,j)=p(i,j)+0.425*(p(i,j+1)+p(i,j-
1)+p(i+1,j)+p(i-1,j)-4*p(i,j))



% solution:

for i=2:18;
   for j=2:12;
      p(j,i)=54

   end
end
```

```
i=19;
for j=2:13;
   p(j,i)=54

end



i=20;
for j=2:14;
   p(j,i)=54

end



i=21;
for j=2:15;
   p(j,i)=54

end



i=22;
for j=2:16;
   p(j,i)=54

end



i=23;

for j=2:16;
   p(j,i)=54

end


i=24;

for j=2:17;
   p(j,i)=54

end
```

```
i=25;

for j=2:18;
   p(j,i)=54

end



i=26;

for j=3:19;
   p(j,i)=54

end


i=27;

for j=5:20;
   p(j,i)=54

end


i=28;

for j=8:21;
   p(j,i)=54

end


i=29;

for j=10:22;
   p(j,i)=54

end


i=30;

for j=12:23;
   p(j,i)=54

end
```

```
i=31;

for j=15:24;
   p(j,i)=54

end


i=32;

for j=16:24;
   p(j,i)=54

end


i=33;

for j=18:25;
   p(j,i)=54

end


i=34;

for j=19:25;
   p(j,i)=54

end


i=35;

for j=20:26;
   p(j,i)=54

end


i=36;

for j=21:26;
   p(j,i)=54

end
```

```
i=37;

for j=21:27;
   p(j,i)=54

end


i=38;

for j=21:27;

   p(j,i)=54

end



i=39;

for j=20:27;
   p(j,i)=54

end


i=40;

for j=19:27;
   p(j,i)=54

end


i=41;

for j=18:27;
   p(j,i)=54

end


i=42;

for j=17:27;
   p(j,i)=54

end
```

```
i=43;

for j=16:26;
   p(j,i)=54

end


i=44;

for j=12:26;
   p(j,i)=54

end


i=45;

for j=9:25;
   p(j,i)=54

end


i=46;

for j=8:25;
   p(j,i)=54

end

% including the middle points:
i=47;

for j=7:15;
   p(j,i)=54

end


i=47;

for j=20:25;
   p(j,i)=54

end


i=48;
```

```
for j=6:14;
   p(j,i)=54

end

%******************
 for i=48:64;
   for j=20:24;
      p(j,i)=54
   end

end




i=49;

for j=5:13;
   p(j,i)=54

end




i=50;

for j=4:12;
   p(j,i)=54

end


i=51;

for j=3:12;
   p(j,i)=54

end


i=52;

for j=3:12;
   p(j,i)=54

end
```

```
i=53;

for j=3:12;
   p(j,i)=54

end


i=54;

for j=2:12;
   p(j,i)=54

end





for i=55:64;
   for j=2:12;
      p(j,i)=54

   end
end





%solutions formula:
for iter=1:4;
 for i=2:18;
   for j=2:12;
      p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-
1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

   end
 end



% my adding

i=19;
for j=2:13;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end
```

```
i=20;
for j=2:14;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end



i=21;
for j=2:15;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end



i=22;
for j=2:16;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end



i=23;

for j=2:16;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=24;

for j=2:17;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end



i=25;
```

```
for j=2:18;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end



i=26;

for j=3:19;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=27;

for j=5:20;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=28;

for j=8:21;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=29;

for j=10:22;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=30;

for j=12:23;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))
```

```
end



i=31;

for j=15:24;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=32;

for j=16:24;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=33;

for j=18:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=34;

for j=19:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=35;

for j=20:26;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end
```

```
i=36;

for j=21:26;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=37;

for j=21:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=38;

for j=21:27;

   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end



i=39;

for j=20:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=40;

for j=19:27;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=41;

for j=18:27;
```

```
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=42;

for j=17:27;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=43;

for j=16:26;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=44;

for j=12:26;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=45;

for j=9:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=46;

for j=8:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end
```

```
% including the middle points:
i=47;

for j=7:15;
  p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=47;

for j=20:25;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end


i=48;

for j=6:14;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end

%******************
 for i=48:64;
   for j=20:24;
      p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))
   end

end




i=49;

for j=5:13;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

end
```

```
i=50;

for j=4:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=51;

for j=3:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=52;

for j=3:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end


i=53;

for j=3:12;
p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-1)-
4*p(j,i))

end


i=54;

for j=2:12;
   p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-1,i)+p(j,i+1)+p(j,i-
1)-4*p(j,i))

end





for i=55:64;
```

```
   for j=2:12;
      p(j,i)=p(j,i)+0.425*(p(j+1,i)+p(j-
1,i)+p(j,i+1)+p(j,i-1)-4*p(j,i))

   end
end



end
```

The output of the program is:

```
   94.7155   91.8885   81.0953   72.8117   65.1679   55.9775   45.7719   40.5085   24.1498   18.2898
  99.0143   90.7697   92.8144   84.1946   74.4640   64.2463   50.8410   47.7807   35.1429   20.4103
 108.0000  103.1793   94.2042   94.6125   85.1620   74.0382   59.0843   53.5793   44.7658   26.4027
        0  108.0000  108.0000   95.6097   94.8943   84.5355   69.4188   60.5638   50.9676   40.2003
        0         0         0  108.0000   95.4274   94.1304   80.7343   69.3677   57.5793   50.5119
        0         0         0         0  108.0000   94.8913   91.3219   79.7201   66.1475   58.5864
        0         0         0         0         0  108.0000   93.1716   90.0785   76.6940   66.9255
        0         0         0         0         0         0  108.0000   92.6391   87.8571   76.9384
        0         0         0         0         0         0         0  108.0000   91.2757   87.7592
        0         0         0         0         0         0         0         0  108.0000   95.0182
        0         0         0         0         0         0         0         0         0  108.0000
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0

 Columns 31 through 40

        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
  11.8650         0         0         0         0         0         0         0         0         0
  23.2551    7.6442         0         0         0         0         0         0         0         0
  24.7143   18.6934         0         0         0         0         0         0         0         0
  36.2541   21.5105   16.1967         0         0         0         0         0         0         0
  47.3898   33.4980   20.0792   15.9399         0         0         0         0         0    6.4028
  56.9640   45.2131   32.1589   23.1843   14.3818         0         0         0    7.6747   13.1232
  66.3596   57.3557   48.7715   36.2288   29.3959   20.7907   16.7324   15.5230   17.7718   20.6763
  82.1548   67.7899   58.5606   48.2124   42.8044   37.5537   33.1367   30.1633   29.1518   29.4047
  83.2463   82.1610   78.6835   62.7694   52.4614   48.8478   46.7219   44.2902   41.8976   40.0135
  99.4474   91.2960   82.7240   81.3506   75.0662   62.6675   55.6643   54.1410   55.0754   56.3859
 108.0000  108.0000  100.4722   93.1032   82.2567   81.5924   77.3810   71.9554   68.4554   67.3832
        0         0  108.0000  108.0000  100.0727   92.9112   88.8083   86.8554   85.2076   90.8779
        0         0         0         0  108.0000  108.0000  100.3329   97.6486   98.5640   98.1599
        0         0         0         0         0         0  108.0000  108.0000  108.0000  108.0000

 Columns 41 through 50

        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0         0    5.0054
        0         0         0         0         0         0         0         0    5.4692    9.9473
        0         0         0         0         0         0         0    5.7079   10.5852   14.6306
        0         0         0         0         0         0    5.4692   10.5852   15.0437   18.7908
        0         0         0         0         0    5.0054    9.9473   14.6306   18.7908   22.3362
        0         0         0         0    4.1021    9.0564   13.8684   18.2896   22.1831   26.0976
        0         0         0         0    6.3166   11.8933   16.8731   21.2797   28.0194   34.6540
        0         0         0         0    8.1610   14.1861   19.2362   30.0855   39.4171   45.8616
        0         0         0    4.5753   10.7176   16.3434   30.0660   41.8575   50.3621   57.3023
        0         0         0    6.5924   12.9056   27.3174   42.0540   51.8995   61.6706   72.0000
        0         0         0    8.0970   21.3387   38.5075   50.6596   61.8390   72.0000         0
        0         0         0   12.9735   30.5797   46.1708   60.4473   72.0000         0         0
        0         0    7.3877   20.7466   37.8607   53.4187   72.0000         0         0         0
        0    6.1134   13.6789   27.6722   42.8490   56.3663   72.0000         0         0         0
```

```
    6.0666   11.6364   19.4781   33.3108   44.7430   57.9087   72.0000         0         0         0
   11.9377   16.8715   24.8797   36.2314   46.3587   57.7414   72.0000   72.0000   72.0000   72.0000
   17.7366   21.8735   29.2410   38.3000   46.3064   54.7058   62.3520   65.4029   66.8745   67.6979
   23.7386   26.7948   32.8037   40.0687   46.0921   52.2239   57.5216   62.5532   67.5139   71.9436
   30.4849   32.0843   36.2587   41.8358   48.1210   55.5113   69.7198   79.8461   84.7499   86.6579
   38.9121   38.6165   42.3256   56.0008   67.1207   84.6068   86.6316   88.4275   90.8132   92.8575
   57.3577   65.0507   81.5876   83.7649   92.0390   89.5453   95.3094   98.7629  100.0889  100.8511
   78.4700   84.2493   81.9687   91.0947   97.2580  100.6316  102.4656  108.0000  108.0000  108.0000
   87.1709   92.4036   97.0692  101.4144  108.0000  108.0000  108.0000         0         0         0
  100.8150  101.8670  108.0000  108.0000         0         0         0         0         0         0
  108.0000  108.0000         0         0         0         0         0         0         0         0

  Columns 51 through 60

         0         0         0         0         0         0         0         0         0         0
         0         0         0    4.5753    6.3281    7.1716    7.6545    7.9558    8.1504    8.2790
    4.1021    6.3166    8.1610   10.7176   12.4563   13.5688   14.3111   14.8262   15.1876   15.4400
    9.0564   11.8933   14.1861   16.3434   18.0312   19.2463   20.1154   20.7509   21.2221   21.5701
   13.8684   16.8731   19.2362   21.2571   22.9125   24.1851   25.1333   25.8413   26.3777   26.7868
   18.2896   21.2797   23.6257   25.5483   27.1440   28.4274   29.4225   30.1806   30.7596   31.2061
   22.1831   25.0831   27.4044   29.2661   30.7963   32.0531   33.0606   33.8485   34.4576   34.9293
   25.5104   28.2922   30.5851   32.4234   33.9123   35.1340   36.1306   36.9284   37.5560   38.0457
   30.0358   33.7032   36.8193   39.2983   41.2164   42.6990   43.8516   44.7475   45.4403   45.9736
   39.7662   43.4532   46.1121   48.0500   49.4785   50.5479   51.3607   51.9820   52.4545   52.8101
   49.9028   52.5964   54.4921   55.8579   56.8523   57.5880   58.1448   58.5742   58.9070   59.1631
   60.4017   62.0817   63.1530   63.8979   64.4253   64.7980   65.0665   65.2692   65.4298   65.5599
   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000   72.0000
   68.1938   68.5016   68.6978   68.8281   68.9186   68.9838   69.0316   69.0664   69.0913   69.1087
   75.4789   78.0517   79.7980   80.9257   81.6307   82.0641   82.3295   82.4929   82.5945   82.6584
   87.4020   87.8353   88.2110   88.5530   88.8396   89.0603   89.2206   89.3329   89.4104   89.4636
   94.1919   94.9555   95.3898   95.6609   95.8496   95.9879   96.0890   96.1610   96.2110   96.2451
  101.4059  101.7943  102.0414  102.1914  102.2851  102.3484  102.3943  102.4284  102.4534  102.4712
  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0

  Columns 61 through 65

         0         0         0         0         0
    6.8006    5.4445    6.2898    5.9321    6.0000
   11.5834   11.7547   11.6882   11.8034   12.0000
   15.4913   17.9316   16.8728   17.4828   18.0000
   19.2421   23.3357   22.1355   23.0375   24.0000
   23.0912   27.9554   27.3917   28.5542   30.0000
   27.0264   32.0456   32.5075   34.0579   36.0000
   30.9409   35.8581   37.4283   39.5343   42.0000
   40.0815   44.9132   47.5358   50.3340   48.0000
   48.2119   52.9701   56.4480   53.2798   54.0000
   56.0865   61.0370   59.1046   59.8991   60.0000
   64.0743   65.8846   65.9821   65.9758   66.0000
   72.0000   72.0000   72.0000   72.0000   72.0000
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
   72.0000   72.0000   72.0000   72.0000   72.0000
   69.9035   72.0117   73.9051   76.1150   78.0000
   85.0088   88.6193   92.6609   96.5667   84.0000
   93.7531   98.5451  104.6279   89.5865   90.0000
  102.6054  108.6017   94.8991   96.8264   96.0000
  110.8972  101.0980  102.9980  102.5252  102.0000
  108.0000  108.0000  108.0000  108.0000  108.0000
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0


p =

  Columns 1 through 10

         0         0         0         0         0         0         0         0         0         0
    9.0000    9.0390    8.9664    8.8672    8.7707    8.6891    8.6281    8.5881    8.5659    8.5565
   18.0000   17.7999   17.4563   17.1082   16.8074   16.5642   16.3767   16.2397   16.1462   16.0875
   27.0000   26.3119   25.5293   24.7965   24.1790   23.6880   23.3107   23.0285   22.8243   22.6825
   36.0000   34.6081   33.2368   32.0070   30.9762   30.1542   29.5203   29.0429   28.6904   28.4362
   45.0000   42.7291   40.6329   38.8064   37.2831   36.0600   35.1079   34.3844   33.8445   33.4482
   54.0000   50.7138   47.7752   45.2604   43.1747   41.4926   40.1708   39.1554   38.3896   37.8210
   63.0000   58.5937   54.7182   51.4334   48.7208   46.5292   44.7948   43.4491   42.4234   41.6535
   72.0000   68.1545   66.2637   65.3233   64.6306   63.8948   63.0784   62.2425   61.4576   60.7703
   81.0000   78.9071   76.6260   74.7245   73.5614   73.0302   72.8559   72.8072   72.7558   72.6575
   90.0000   88.3168   87.6139   87.0863   86.4593   85.8274   85.3166   84.9687   84.7582   84.6368
   99.0000   98.3708   97.8507   97.5348   97.3374   97.1660   96.9935   96.8315   96.6965   96.5950
  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000  108.0000
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
```

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Columns 11 through 20

0	0	0	0	0	0	0	0	0	0
8.5549	8.5573	8.5612	8.5650	8.5683	8.5708	8.5726	8.5738	8.5746	8.5751
16.0543	16.0382	16.0325	16.0323	16.0346	16.0376	16.0404	16.0426	16.0443	16.0455
22.5890	22.5311	22.4979	22.4808	22.4734	22.4714	22.4719	22.4735	22.4752	22.4767
28.2579	28.1374	28.0594	28.0113	27.9835	27.9686	27.9616	27.9589	27.9585	27.9591
33.1626	32.9612	32.8229	32.7309	32.6719	32.6357	32.6145	32.6030	32.5972	32.5947
37.4048	37.1046	36.8920	36.7445	36.6447	36.5790	36.5372	36.5116	36.4965	36.4881
41.0833	40.6664	40.3655	40.1515	40.0019	39.8995	39.8309	39.7862	39.7579	39.7405
60.1998	59.7451	59.3943	59.1308	58.9376	58.7990	58.7019	58.6353	56.8291	53.8051
72.5144	72.3463	72.1744	72.0142	71.8750	71.7602	71.6698	66.3159	63.6077	61.7469
84.5614	84.5039	84.4503	84.3961	84.3418	84.2898	78.9573	81.0881	74.3994	69.1049
96.5240	96.4759	96.4428	96.4183	96.3983	94.6189	95.6887	89.3567	89.5237	81.1540
108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	94.1696	93.2059
0	0	0	0	0	0	0	0	108.0000	95.0775
0	0	0	0	0	0	0	0	0	108.0000
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Columns 21 through 30

0	0	0	0	0	0	0	0	0	0
8.5754	6.8138	9.4607	1.8961	3.5240	0	0	0	0	0
16.0463	13.0518	15.7900	5.6684	10.6125	1.7767	0	0	0	0
22.4779	19.2966	18.6450	16.3366	8.2362	6.1658	0	0	0	0
27.9600	25.2560	21.7637	22.9627	10.4501	7.7808	5.1479	0	0	0
32.5939	30.5822	26.0079	25.0723	15.3039	13.7688	4.9856	0	0	0
36.4837	35.1137	30.7904	25.3532	24.6351	12.2949	8.0891	0	0	0
39.7302	38.8523	35.3898	27.4116	29.2164	14.9566	15.0332	3.5822	0	0
50.6045	47.3589	42.8457	33.3900	30.9852	25.2832	13.5728	8.3430	0	0
59.3362	56.0584	51.3708	41.9640	34.9779	32.6325	17.1930	16.3592	4.2826	0
65.5946	62.4713	58.4365	50.4260	41.3568	36.4270	28.9003	15.9452	9.8217	0
74.0698	69.0336	64.6254	57.8596	48.7157	40.7143	36.9925	20.3394	13.2513	7.4587
84.7104	77.1822	73.0535	67.8000	59.4263	49.3558	42.8848	28.4772	22.2402	9.4970
94.7155	91.8885	81.0953	72.8117	65.1679	55.9775	45.7719	40.5085	24.1498	18.2898
99.0143	90.7697	92.8144	84.1946	74.4640	64.2463	50.8410	47.7807	35.1429	20.4103
108.0000	103.1793	94.2042	94.6125	85.1620	74.0382	59.0843	53.5793	44.7658	26.4027
0	108.0000	108.0000	95.6097	94.8943	84.5355	69.4188	60.5638	50.9676	40.2003
0	0	0	108.0000	95.4274	94.1304	80.7343	69.3677	57.5793	50.5119
0	0	0	0	108.0000	94.8913	91.3219	79.7201	66.1475	58.5864
0	0	0	0	0	108.0000	93.1716	90.0785	76.6940	66.9255
0	0	0	0	0	0	108.0000	92.6391	87.8571	76.9384
0	0	0	0	0	0	0	108.0000	91.2757	87.7592
0	0	0	0	0	0	0	0	108.0000	95.0182
0	0	0	0	0	0	0	0	0	108.0000
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Columns 31 through 40

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
11.8650	0	0	0	0	0	0	0	0	0
23.2551	7.6442	0	0	0	0	0	0	0	0
24.7143	18.6934	0	0	0	0	0	0	0	0
36.2541	21.5105	16.1967	0	0	0	0	0	0	0
47.3898	33.4980	20.0792	15.9399	0	0	0	0	0	6.4028
56.9640	45.2131	32.1589	23.1843	14.3818	0	0	0	7.6747	13.1232
66.3596	57.3557	48.7715	36.2288	29.3959	20.7907	16.7324	15.5230	17.7718	20.6763
82.1548	67.7899	58.5606	48.2124	42.8044	37.5537	33.1367	30.1633	29.1518	29.4047
83.2463	82.1610	78.6835	62.7694	52.4614	48.8478	46.7219	44.2902	41.8976	40.0135
99.4474	91.2960	82.7240	81.3506	75.0662	62.6675	55.6643	54.1410	55.0754	56.3859
108.0000	108.0000	100.4722	93.1032	82.2567	81.5924	77.3810	71.9554	68.4554	67.3832
0	0	108.0000	108.0000	100.0727	92.9112	88.8083	86.8554	85.2076	90.8779
0	0	0	0	108.0000	108.0000	100.3329	97.6486	98.5640	98.1599
0	0	0	0	0	0	108.0000	108.0000	108.0000	108.0000

Columns 41 through 50

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	5.0054
0	0	0	0	0	0	0	0	5.4692	9.9473
0	0	0	0	0	0	0	5.7079	10.5852	14.6306
0	0	0	0	0	0	5.4692	10.5852	15.0437	18.7908
0	0	0	0	0	5.0054	9.9473	14.6306	18.7908	22.3362
0	0	0	0	4.1021	9.0564	13.8684	18.2896	22.1831	26.0976
0	0	0	0	6.3166	11.8933	16.8731	21.2797	28.0194	34.6540
0	0	0	0	8.1610	14.1861	19.2362	30.0855	39.4171	45.8616
0	0	0	4.5753	10.7176	16.3434	30.0660	41.8575	50.3621	57.3023
0	0	0	6.5924	12.9056	27.3174	42.0540	51.8995	61.6706	72.0000
0	0	0	8.0970	21.3387	38.5075	50.6596	61.8390	72.0000	0
0	0	0	12.9735	30.5797	46.1708	60.4473	72.0000	0	0
0	0	7.3877	20.7466	37.8607	53.4187	72.0000	0	0	0
0	6.1134	13.6789	27.6722	42.8490	56.3663	72.0000	0	0	0
6.0666	11.6364	19.4781	33.3108	44.7430	57.9087	72.0000	0	0	0
11.9377	16.8715	24.8797	36.2314	46.3587	57.7414	72.0000	72.0000	72.0000	72.0000
17.7366	21.8735	29.2410	38.3000	46.3064	54.7058	62.3520	65.4029	66.8745	67.6979
23.7386	26.7948	32.8037	40.0687	46.0921	52.2239	57.5216	62.5532	67.5139	71.9436
30.4849	32.0843	36.2587	41.8358	48.1210	55.5113	69.7198	79.8461	84.7499	86.6579
38.9121	38.6165	42.3256	56.0008	67.1207	84.6068	86.6316	88.4275	90.8132	92.8575
57.3577	65.0507	81.5876	83.7649	92.0390	89.5453	95.3094	98.7629	100.0889	100.8511
78.4700	84.2493	81.9687	91.0947	97.2580	100.6316	102.4656	108.0000	108.0000	108.0000
87.1709	92.4036	97.0692	101.4144	108.0000	108.0000	108.0000	0	0	0
100.8150	101.8670	108.0000	108.0000	0	0	0	0	0	0
108.0000	108.0000	0	0	0	0	0	0	0	0

Columns 51 through 60

0	0	0	0	0	0	0	0	0	0
0	0	0	4.5753	6.3281	7.1716	7.6545	7.9558	8.1504	8.2790
4.1021	6.3166	8.1610	10.7176	12.4563	13.5688	14.3111	14.8262	15.1876	15.4400
9.0564	11.8933	14.1861	16.3434	18.0312	19.2463	20.1154	20.7509	21.2221	21.5701
13.8684	16.8731	19.2362	21.2571	22.9125	24.1851	25.1333	25.8413	26.3777	26.7868
18.2896	21.2797	23.6257	25.5483	27.1440	28.4274	29.4225	30.1806	30.7596	31.2061
22.1831	25.0831	27.4044	29.2661	30.7963	32.0531	33.0606	33.8485	34.4576	34.9293
25.5104	28.2922	30.5851	32.4234	33.9123	35.1340	36.1306	36.9284	37.5560	38.0457
30.0358	33.7032	36.8193	39.2983	41.2164	42.6990	43.8516	44.7475	45.4403	45.9736
39.7662	43.4532	46.1121	48.0500	49.4785	50.5479	51.3607	51.9820	52.4545	52.8101
49.9028	52.5964	54.4921	55.8579	56.8523	57.5880	58.1448	58.5742	58.9070	59.1631
60.4017	62.0817	63.1530	63.8979	64.4253	64.7980	65.0665	65.2692	65.4298	65.5599
72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000	72.0000
68.1938	68.5016	68.6978	68.8281	68.9186	68.9838	69.0316	69.0664	69.0913	69.1087
75.4789	78.0517	79.7980	80.9257	81.6307	82.0641	82.3295	82.4929	82.5945	82.6584
87.4020	87.8353	88.2110	88.5530	88.8396	89.0603	89.2206	89.3329	89.4104	89.4636
94.1919	94.9555	95.3898	95.6609	95.8496	95.9879	96.0890	96.1610	96.2110	96.2451
101.4059	101.7943	102.0414	102.1914	102.2851	102.3484	102.3943	102.4284	102.4534	102.4712
108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000	108.0000
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Columns 61 through 65

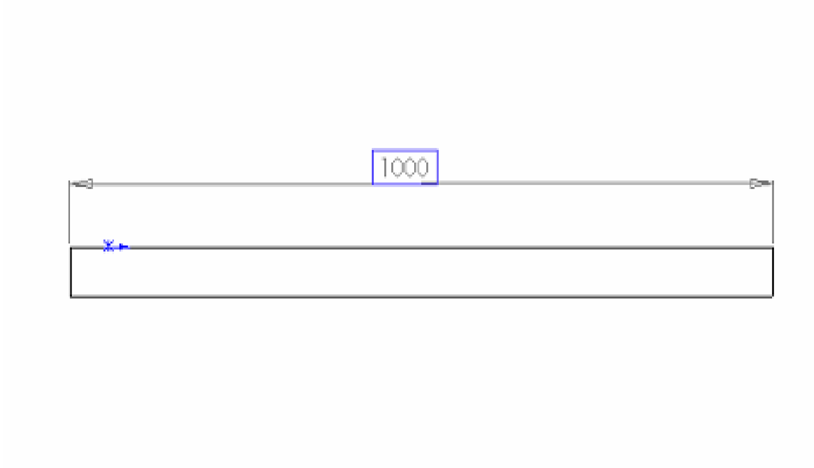
0	0	0	0	0
6.8006	5.4445	6.2898	5.9321	6.0000
11.5834	11.7547	11.6882	11.8034	12.0000
15.4913	17.9316	16.8728	17.4828	18.0000
19.2421	23.3357	22.1355	23.0375	24.0000
23.0912	27.9554	27.3917	28.5542	30.0000
27.0264	32.0456	32.5075	34.0579	36.0000
30.9409	35.8581	37.4283	39.5343	42.0000
40.0815	44.9132	47.5358	50.3340	48.0000
48.2119	52.9701	56.4480	53.2798	54.0000
56.0865	61.0370	59.1046	59.8991	60.0000
64.0743	65.8846	65.9821	65.9665	66.0000
72.0000	72.0000	72.0000	72.0000	72.0000
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
72.0000	72.0000	72.0000	72.0000	72.0000
69.9035	72.0117	73.9051	76.1150	78.0000
85.0088	88.6193	92.6609	96.5667	84.0000
93.7531	98.5451	104.6279	89.5865	90.0000
102.6054	108.6017	94.8991	96.8264	96.0000
110.8972	101.0980	102.9980	102.5252	102.0000
108.0000	108.0000	108.0000	108.0000	108.0000
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Appendix B: (drawings):

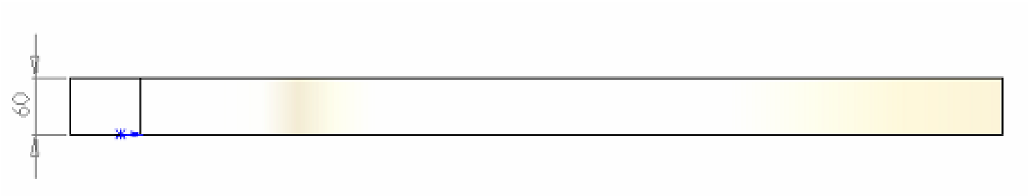
Part 1:

Front

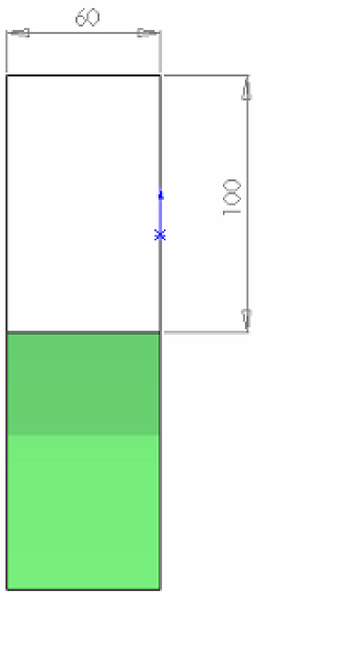
Top



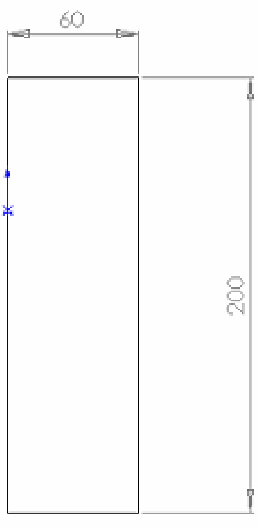
Bottom



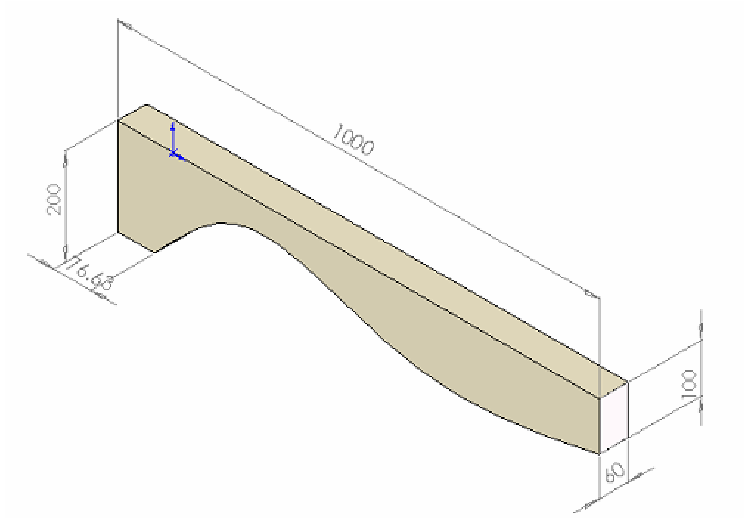
Right veiw:



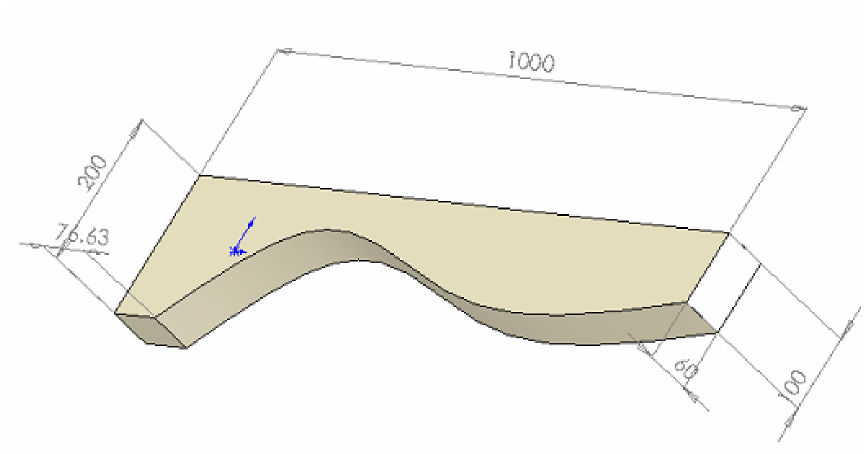
Left:



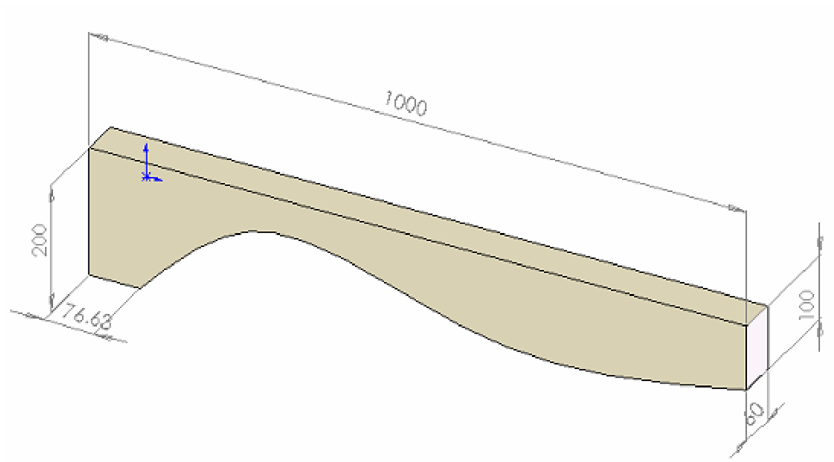
Isometric:



Trimetric:

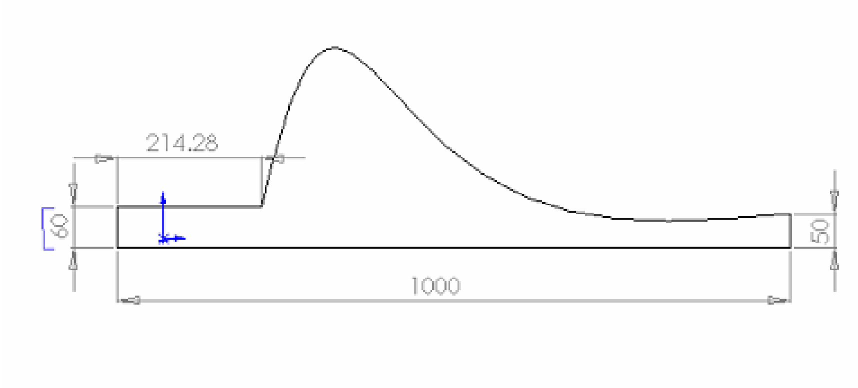


Diametric:



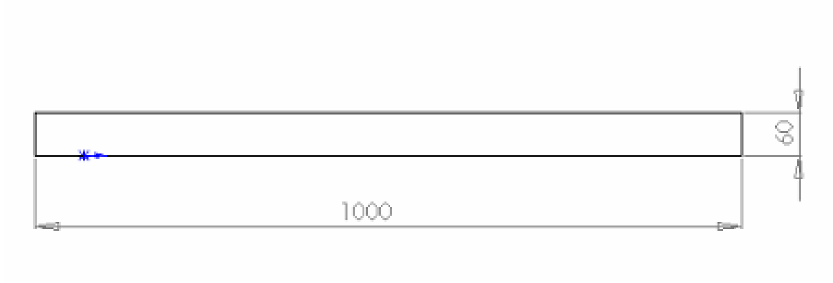
Part 2:

Front:

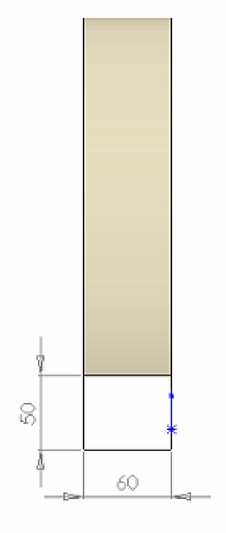


Top:

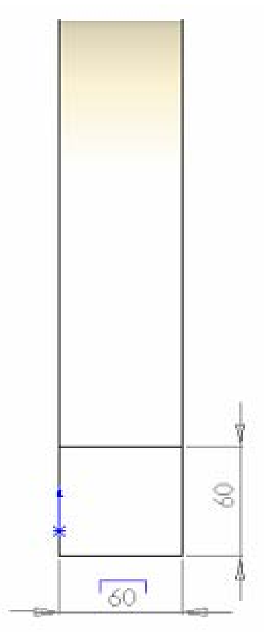
Bottom:



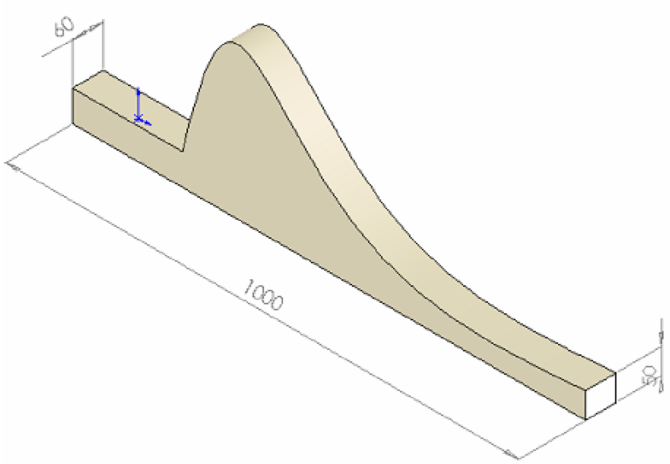
Right:



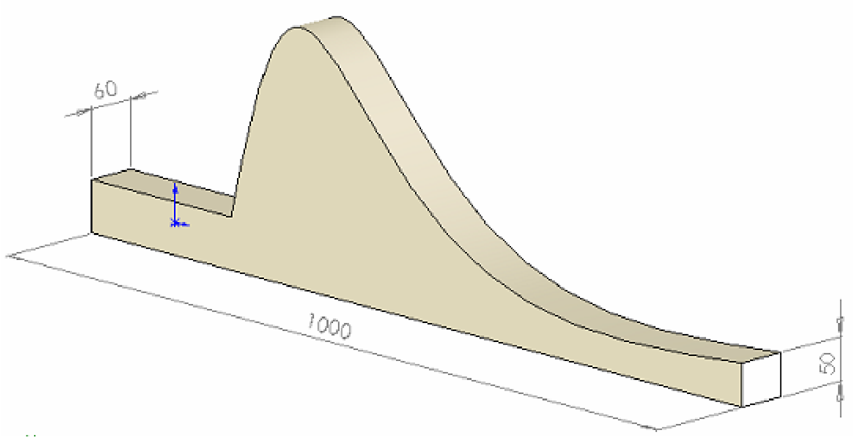
Left:



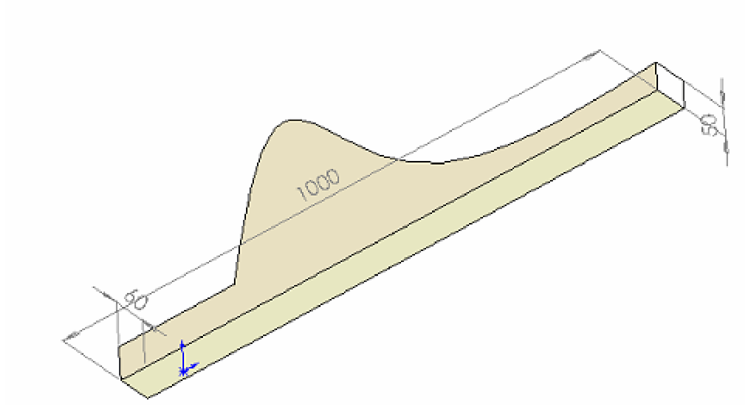
Isometric:



Diametric:

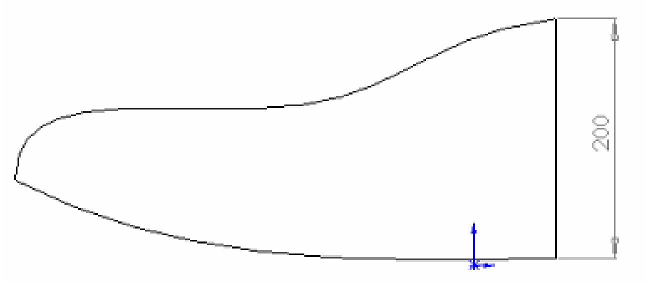


Trimetric:

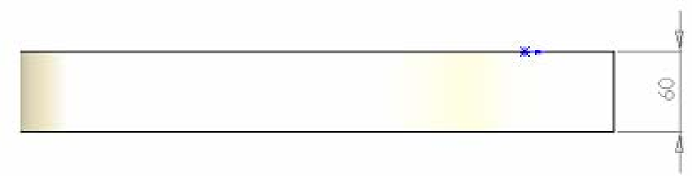


Part 3:

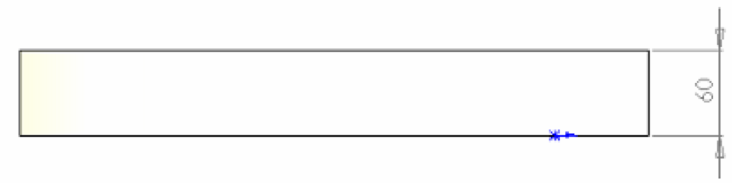
Front:



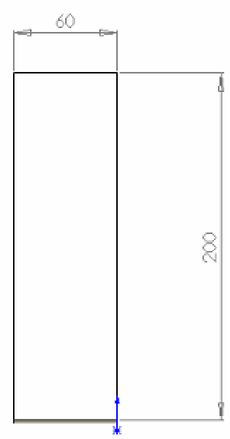
Top:



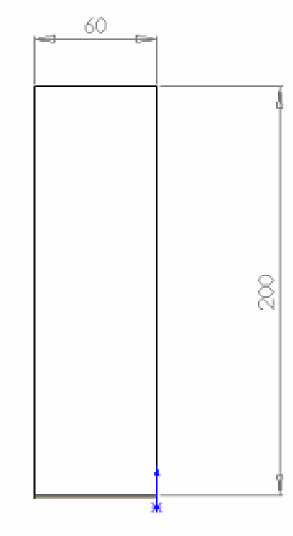
Bottom:



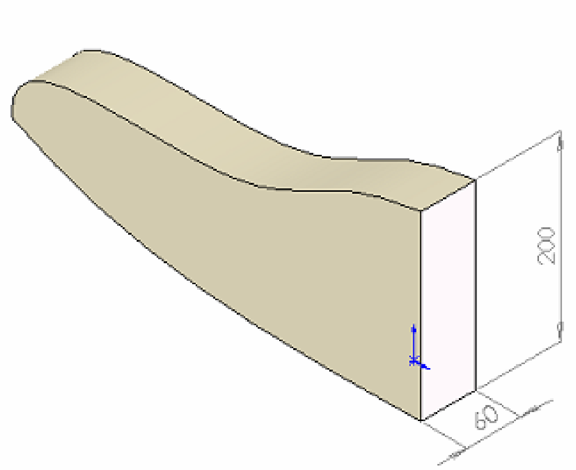
Right:



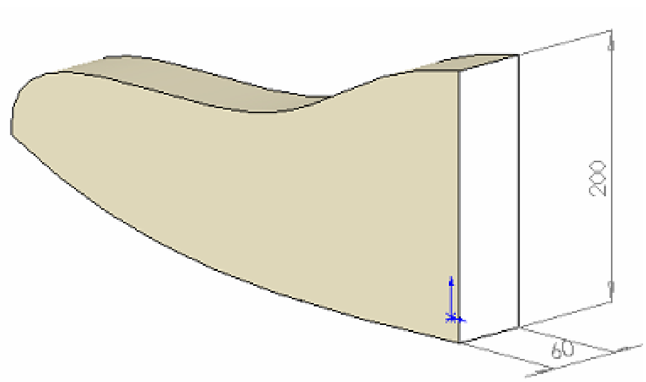
Left:



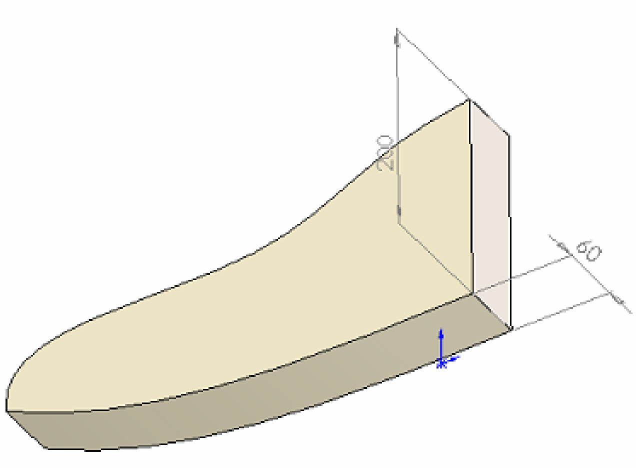
Isometric:



Diametric:

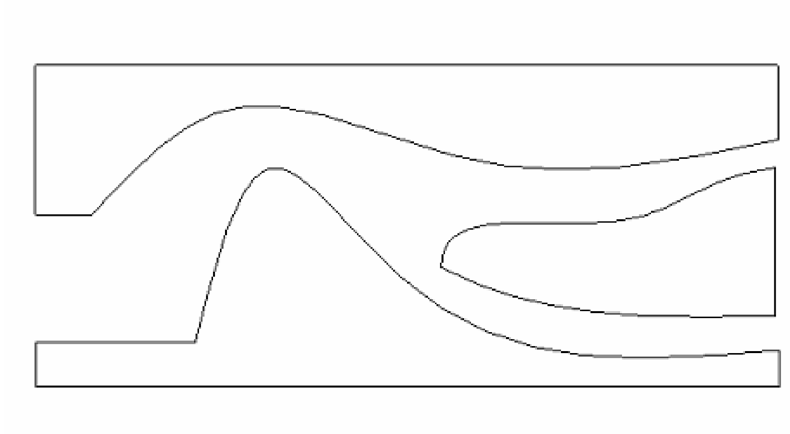


Trimetric:

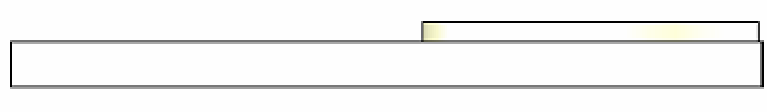


Assembly drawing:

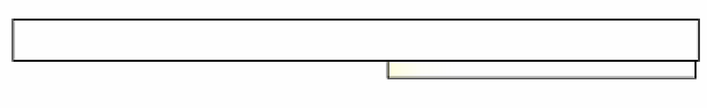
Front:



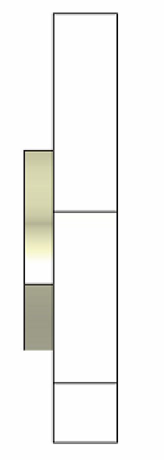
Top:



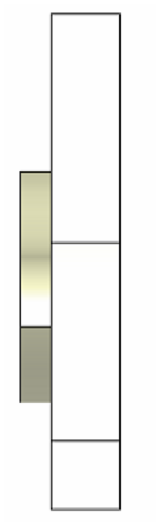
Bottom:



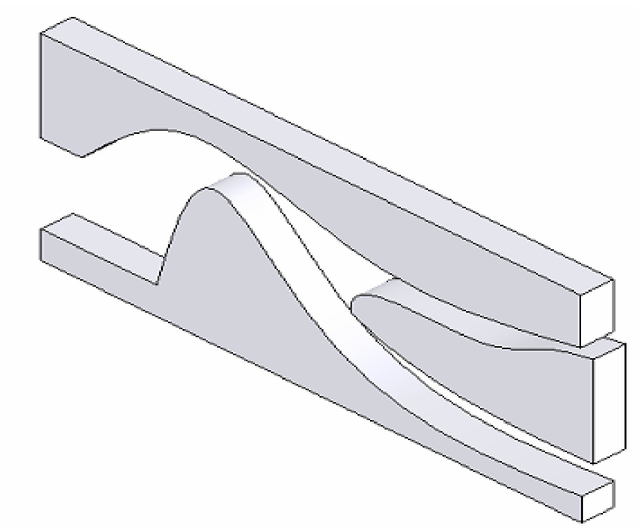
Right:



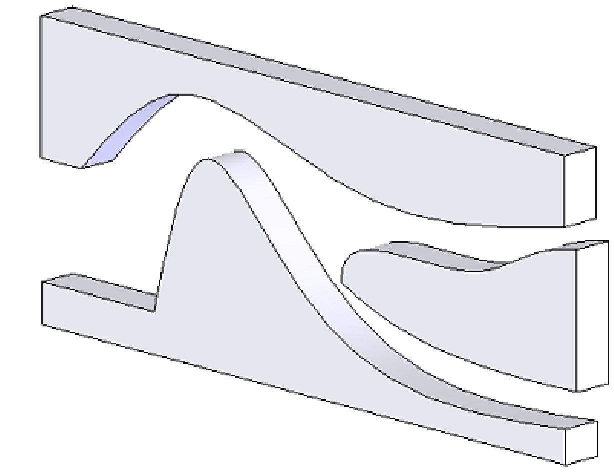
Left:



Isometric:



Diametric:



Trimetric: