2

This is to certify that the

thesis entitled

GRID GENERATION OF VOLUTES USING VISUAL BASIC

presented by

Joscelyn Wayne Pereira

has been accepted towards fulfillment of the requirements for

MS degree in Engineering

Major professor

Date December 1st, 1998

O-7639

GRID GENERATION OF VOLUTES USING VISUAL BASIC

By

Joscelyn Wayne Pereira

A THESIS

Submitted to
Michigan State University
in partial fulfillment of requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1998

ABSTRACT

GRID GENERATION OF VOLUTES USING VISUAL BASIC

By

Joscelyn Wayne Pereira

The design of most volutes, for centrifugal compressors and radial turbines are based on a linear A/R (cross sectional area by centroid radius) distribution together with some empirical experience of the designer. Depending on the particular configuration, the losses in volutes especially at off-design points can be significantly high leading to poor stage efficiency. It is strongly believed that volute performance improvements are possible if an aerodynamic analysis aids the design process. The aerodynamic analysis could be one-dimensional, two-dimensional or three-dimensional in a form of computational fluid dynamics (CFD). To analyze a volute aerodynamically, a geometric model of the volute is required. This geometric model should allow the various volute parameters to be systematically defined and to be easily varied.

This thesis has developed the necessary mathematical model to define the geometry of the volute. The model with its procedures has been programmed in Visual Basic with the objective of serving as an aid for volute design and performance analysis.

For Mama

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation to several persons who have helped me make this possible. Firstly to Dr. Abraham Engeda, my mentor and friend who has helped me in pursuing a field I was interested in. His technical guidance, support and encouragement, throughout the course of this work were invaluable and very much appreciated. To Dr Ranjan Mukherjee and Dr. John McGrath, the committee members for their advice, suggestions and interest in this work, and for their help and support; and to Mr. Craig Gunn at MSU for his help in preparing this thesis.

I would also like to specially thank Mr. Geoffrey Bruce and Mr. Jonathan McGuire from Schwitzer Inc. for their guidance, support and criticism which has helped me gain a vast amount of knowledge and experience. I also express my gratitude to Schwitzer Inc. for their support of this project.

A special thank you also for Mr. Sandeep Rao for his numerous criticism, assistance and friendship in painstakingly going through this thesis. Also to all the other friendships and great people that have made my stay at Michigan State an enjoyable one.

Last, but certainly not the least, to all my family for their support, help and love, and especially to mama, who still has her one hand behind me, pushing me all the way.

Friday November 13, 1998 *Joscelyn Wayne Pereira*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	area
b	width or distance along a quasi orthogonal
c	absolute velocity
m	mass flow
P	pressure
r	radius
R1	radius of volutes
R2	radius of volutes
R3	corner radius
S	width of volute

Greek

φ	angle along the volute
ρ	density
μ	mass ratio
ω	rotational speed

Subscripts

2	impeller exit
3	diffuser inlet
4	diffuser exit
5	volute inner surface
6	volute outer surface
c	effective compressor mass flow
CR	centroid radius
max	maximum
o	recirculating mass flow
th	throat
u	tangential component of velocity
z	recirculating flow

Chapter 1

INTRODUCTION

1.1 Introduction

A volute is used to either distribute or to collect fluid from a rotor. Volute casings are well established in turbocharger applications. They are cost effective to manufacture and have an aerodynamic advantage of adapting to a wide operating range. However, the shape of such casings takes the role of vanes in guiding the flow into the rotor, and performance penalties are often paid for incorrect design. However, there is very little documented data on internal flow measurements in volutes. Because of their complex geometry and form, detailed flow patterns have not been obtained. Therefore, the effects of geometrical parameters on performance have not been clarified adequately.

In recent times, turbochargers and turbocharging technologies have progressed significantly. There has been a great deal of research into improvement of compressor and turbine aerodynamic performance: new bearings, new materials, variable geometry systems and new control systems. This has all contributed to efficient turbocharged engine performance.

In the interests of size, cost and response it is usual for automotive turbochargers to be small, high specific speed units. The compressor impeller exit flow has high kinetic energy, typically fifty percent of the total pressure at the impeller exit will be dynamic pressure and fifty percent static pressure. The diesel engine cannot utilize this level of kinetic energy and if good stage performance is to be achieved, then it must be recovered and converted to static pressure before it reaches the engine valves. This recovery must therefore be attempted in the diffuser after the impeller and in the volute.

The heart of the compressor is the impeller, where all the energy transfer takes place; it would be the likely suspect for any shortcomings that could occur. Surprisingly, the designer has had more success with the impeller than with the diffuser or volute systems. The design of an appropriate diffusing and volute system to slow down the fluid efficiently has been for a long time and still is one of the main difficulties in centrifugal compressor design. The strong demand, by turbocharger users for turbochargers with good range and efficiency, is forcing the turbocharger designer to review the aerodynamics. Two fruitful sources for potential improvement in turbocharger performance are in:

- Proper design of the diffuser and volute, and
- Good understanding of the factors affecting the stable operating range and pressure recovery of diffusers and volutes.

Taking the centrifugal compressor as an example, its basic elements are defined as a stationary casing containing a rotating impeller followed by a radial diffuser as illustrated in Figure 1.1. The fluid is drawn in through the inlet casing into the eye of the impeller parallel to the axis of rotation. In a radial compressor for use in gas turbines or turbochargers this axial portion is usually referred to as the inducer. In order to increase the angular momentum, the impeller whirls the fluid outwards and turns it in the direction perpendicular to the rotation axis. As a result, the energy level is increased and both higher pressure and velocity are achieved. The purpose of the following diffuser is to convert the kinetic energy of the fluid into additional pressure energy. Following the

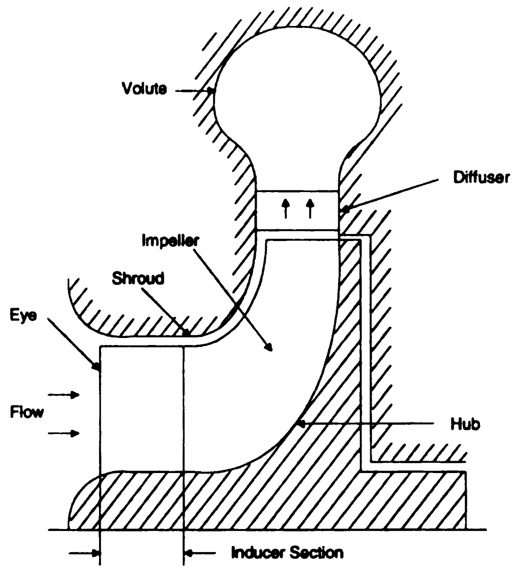


Figure 1.1: Cross Section of Centrifugal Compressor

diffuser is a scroll or volute whose function is to collect the flow from the diffuser and deliver it to the outlet pipe. It is possible to gain a further deceleration and thereby additional pressures rise in this part of the compressor.

The flow in turbine volutes is of a complex nature, due to the high degree of turning and curvature, as well as due to the existence of a recirculation region around what is commonly known as 'the tongue', where the flow renters the volute and high mixing occurs. The function of the volute will depend upon whether the turbine has inlet nozzle vanes. If it has, then the volute must simply deliver a uniform gas flow ahead of the rotor; if it has not, it must also guide the flow ahead of the rotor.

Much of the early work on volute design employed one-dimension, inviscid, incompressible analysis. This analysis assumed a constant angular momentum, which is called free vortex flow. Clearly these analysis were not realistic, as they do not take into account the effects of the geometry, viscosity and compressibility.

1.2 The Need for a Design Tool.

It is seen from the preceding section that improvement in turbocharger component efficiency is an important goal for all turbomachinery designs.

Most of the research work has been concentrated on rotors since they were considered to have the main effect on turbomachinery efficiency. Recent research has shown the need for new design techniques for the volutes to improve overall performance.

Turbochargers are some of the smallest machinery that fall under the general category of turbo machines. In these machines a change in a percentage point in the efficiency is of great importance to the designer. One of the main areas where the efficiency could be improved by efficient and sufficient design, are the volutes for compressors and turbines.

At present the most common way of designing of volutes empirically, follows the testing of the volute. The aerodynamic engineer specifies the critical area and supplies this to the drafting engineer. The draftsman uses the critical area and the linear distribution of the area (or A/R), generates the cross section shapes, and wraps the volute around the rotor. The volute casing is then manufactured and is tested. Overall stage performance across the machine (including the volute) is measured.

Ideally, the test of the volute would like to be done only after a simulation has proved a design. But this would be possible only if a suitable design tool were available that would simulate a volute model, which could then be analyzed. The design tool should then be capable of manipulating the geometry of the volute if an unfavorable result was obtained. This iteration process is completed once an optimized design is achieved.

Thus, the design process would be completed and proven analytically with the analysis of the volute model on a computer before a test could be performed to verify the expected results, thereby saving the designer, and hence industry valuable time and resources.

1.3 Objectives and Structure of this Thesis.

The objective of this thesis is to build a universal design tool applicable to a wide variety of volute configurations that would be more powerful and more useful to practicing design engineers. A design tool that could achieve this would meet the following requirements.

(1) Easy to understand and use.

(2) Capabile of manipulating the geometry.

(3) Able to analyze the design with the help of necessary packages

(4) Able to export the volute to another package, for e.g. Drafting, Manufacturing etc.

A thoughtfully prepared approach is introduced in this thesis; concentrating on building the design platform, being able to manipulate the geometry and exporting the volute to other packages. The design tool is built in Visual Basic, a powerful interactive programming language available for personal computers. Visual Basic allows for easy graphical modeling of the volute cross sections through the use of Bezier polynomials. It is capable of supporting well designed systems with boundless capabilities; and because it is a Windows based system it is extremely user friendly.

This thesis is divided into six chapters;

Chapter 2, "Volutes", presents a general overview of volutes, their types and the general design procedure.

Chapter 3, "Mathematical Elements", contains the main mathematical elements used to generate the volute cross sections. This chapter gives detailed explanation of the curves, the design of the different cross sections, and the various formulas required to calculate these cross sections.

Chapter 4, "Volute Design – The Program", is written as a manual for the program. This provides the help to the terminology used describing different terms. It provides a detailed description of each screen, their menus and functions.

Chapter 5, "Geometry Verification", attempts to verify the geometry external to Visual Basic.

Chapter 6, "Conclusions and Future Work", throws some light on the future course of this project.

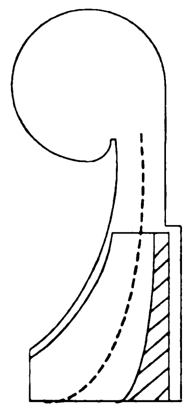
The appendices contain information important to the code. They contain a list of all subroutines available from the code, as well as some sample files that the program uses.

Chapter 2

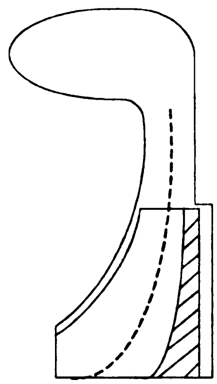
VOLUTES

2.1 Introduction

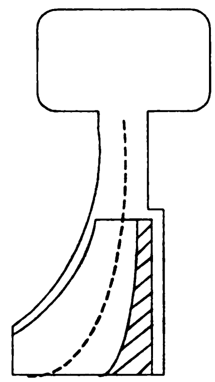
Volutes are mainly classified based on their shape. The most common shapes are circular. Overhang volutes are those volutes where the centroid radius is lower than the inner volute radius. Figure 2.1 illustrates a few common volutes.



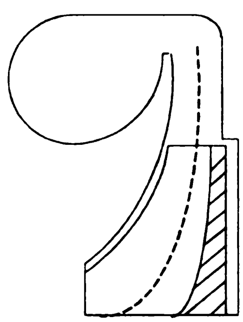
(a) Circular Volute



(b) Elliptical Volute



(c) Rectangular Volute



(d) Overhang Volute

Figure 2.1: Types of Volutes

Past work has focused on the design of the rotor, while the impact of the volute configuration on the efficiency of a turbo-machine was often neglected. Recent research work has shown the importance of volute design techniques in further improvement of the turbo machine performance.

The main object for designing the volute is to provide a uniform distribution of flow parameters along the circumference at the outlet of the volute so that the blades of the rotor or the vanes of the stator do not experience fluctuating forces. Reduction or elimination of such forces would significantly increase fatigue life.

Until recently, analytical solutions of the flow field in the volute were limited to one or two-dimensional flow assumptions. Some one and two-dimensional analyses on the volute cross section planes were performed with various through-flow, velocity profiles. Although these investigations provided some insight about the features of the flow in the volute, they do not provide a solution for a real three-dimensional flow. A true three-dimensional analysis is needed since the through-flow velocity profile depends on both the cross-sectional configuration of the volute and its location along the circumference.

At the present time, volute designs are still based on one-dimensional flow calculations. The inlet velocity distribution has been assumed to be uniform across the volute. However, such assumptions are unrealistic; a variation in the inlet velocity distribution exists and depends mainly on the volute effects.

To begin a design, the flow within the volute is treated as one dimensional and free of vortex. From this, the flow area and its centroid at any azimuth angle can be calculated. The exact shape of the cross section of the volute is, however, left undecided.

A designer must use personal experience in choosing the cross section. One problem associated with such methods is that because the flow equations are only satisfied at the centroids of the flow passage and not at the wall. The flow at the wall is likely to have different conditions and velocity vectors from those at the centroid.

2.2 Design Approach Review

In the design of a compressor volute, the main problem is to provide a constant pressure along the circumference so that a point on the impeller does not experience a fluctuating force. A simple procedure is to design for constant velocity at volute inlet by continuity, but for completeness it is necessary to account for frictional and density effects. The calculation of the circumferential variation of the cross-sectional area of the volute then becomes elaborate. A full analysis is given by Brown and Bradshow [3]. This calculation procedure furnishes the cross-sectional area but gives no guidance as to the shape. Brown and Bradshaw [3] also investigated four typical volute forms and showed that for these four types exactly the same compressor performance was obtained. On the other hand, Eckert [4] showed that certain volute geometries are more efficient than others.

Stiefel [5] studied the optimization of the impeller, vaneless diffuser and volute. He found that with a vaneless diffuser the optimum volute operation was achieved when the volute was ten to fifteen per cent smaller than that which would be designed through the frictionless flow assumption. By reducing the size of the volute by thirty percent, he transformed an unstable performance characteristic to a stable one up to a pressure ratio of *6.3*. This was done by changing the design point of the volute from a pressure ratio of *3.8 : 1* to one of *6 : 1*.

For a radial turbine volute Chapple, Flynn and Mully [6] developed a performance prediction approach and performed the designs based on it. The large number of parameters influencing the performance of a centrifugal compressor volute prohibit systematic experimental investigation because of the time and cost involved in the manufacturing and testing of the complex three-dimensional geometry.

2.3 Simple Theoretical Compressor Volute Design

Figure 2.2 shows typical volute geometry.

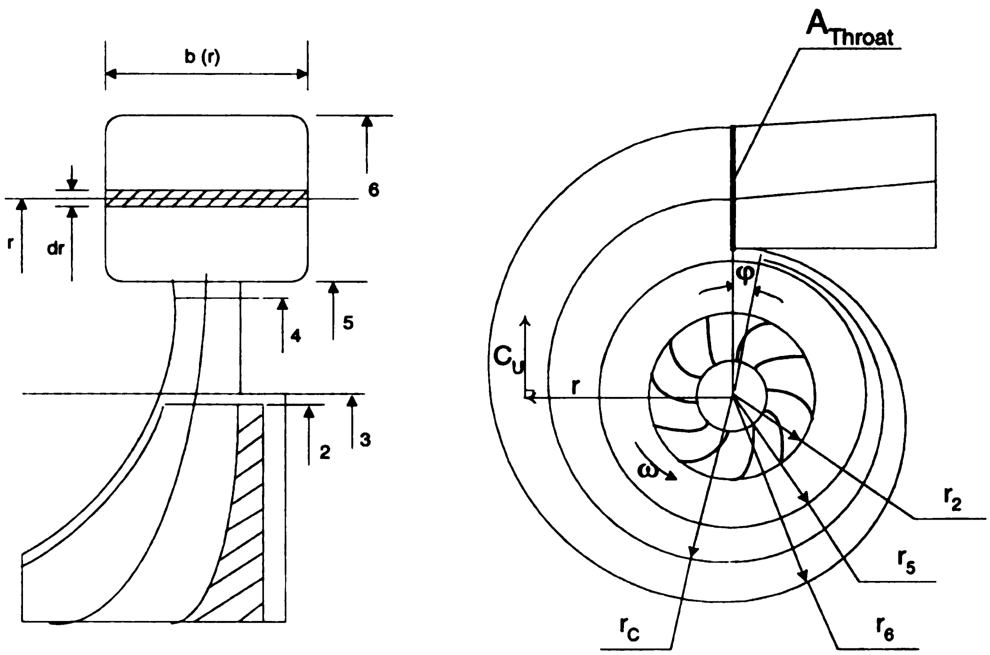


Figure 2.2: Basic Volute Geometry

Using simplified theory, volute flow aims to collect the flow uniformly along the circumference from $\varphi = 0$ to $\varphi = \varphi_{max}$ (usually $\varphi_{max} = 360^0 - 8^0 = 352^0$).

The mass flow distribution can be described as

$$m(\varphi) = \frac{m_c}{\varphi_{\max}} \varphi + m_o \tag{2.1}$$

Where m_c is the effective compressor flow and m_o is the recirculating flow through the tongue gap. Rearranging equation (2.1)

$$\frac{m(\varphi)}{m_c} = \frac{\varphi}{\varphi_{\max}} + \frac{m_o}{m_c} \tag{2.2}$$

gives a definition for the recirculating mass flow and for the mass flow ratio as

The recirculating mass flow and

$$\frac{m_o}{m_c} = \mu_z$$

The mass ratio

$$\frac{m(\varphi)}{m_c} = \mu(\varphi)$$

Thus

$$\mu(\varphi) = \frac{\varphi}{\mu_{\max}} + \mu_z \tag{2.3}$$

To determine the optimum path for the outer radius r_6 as a function of the angle φ, the conservation of angular momentum is applied as

$$r\, c_u = \frac{D_5}{2} c_{u5} = Const \tag{2.4}$$

Which gives the magnitude of the velocity as dependent on the radius. Applying continuity of mass through a cross-sectional area dA gives

$$dm = dA\ \rho\ c_u(r) \tag{2.5}$$

$$dm = b(r)\, dr\, \rho\, \frac{D_5}{2} \frac{c_{u5}}{r} \tag{2.6}$$

The total mass flow through $A(\varphi)$ is

$$m(\varphi) = \frac{D_5}{2} c_{u5} \int_{r=\frac{D_5}{2}}^{r_6} \rho\, b(r) \frac{dr}{r} \tag{2.7}$$

A first simple solution is to assume that density in the cross-section $A(\varphi)$ is constant, which gives

$$\frac{m(\varphi)}{m_c} = \frac{D_5}{2} c_{u5} \frac{\rho(\varphi)}{m_c} \int_{r=\frac{D_5}{2}}^{r_6} b(r) \frac{dr}{r} = \mu(\varphi) \tag{2.8}$$

Using equation (2.3), one of the basic equations of volutes can be obtained as

$$\int_{r=\frac{D_5}{2}}^{r_6} \frac{b(r)}{D_5} \frac{dr}{r} = \frac{\dfrac{\varphi}{\varphi_{max}} + \mu_z}{\dfrac{D^2{}_5}{2} c_{u5} \dfrac{\rho(\varphi)}{m_c}} \tag{2.9}$$

The solution to equation (2.9) describes the distribution of the outer radius r_6 as a function of the angle φ.

A very simple solution is usually to assume constant density circumferentially

$$\rho(\varphi) = const = \rho_5 = \rho_{th} \tag{2.10}$$

And a rectangular cross-sectional area

$$b(r) = const = b_D \tag{2.11}$$

So that equation (2.9) simplifies to

$$\int_{r=\frac{D_5}{2}}^{r_6} \frac{dr}{r} = \frac{D_5}{b_D} \frac{\dfrac{\varphi}{\varphi_{max}} + \mu_z}{\dfrac{D_5^2}{2} \dfrac{c_{u5}}{m_c} \rho_{th}} \tag{2.12}$$

And can be solved to give the solution

$$\ln r\Big|_{r=\frac{D_5}{2}}^{r_a} = \ln\frac{2.r_6}{D_5} = \frac{1}{b_D}\frac{\frac{\varphi}{\varphi_{\max}}+\mu_z}{\frac{D_5}{2}\frac{c_{u5}}{m_c}\rho_{th}} \tag{2.13}$$

$$\frac{r_6}{D_5} = \frac{1}{2}e^{\left(\frac{\frac{\varphi}{\varphi_{\max}}+\mu_z}{b_D.\frac{D_5}{2}.\frac{c_{u5}}{m_c}.\rho_{th}}\right)} \tag{2.14}$$

The second basic equation for the volute is based on the equation of motion for an adiabatic, inviscid, and incompressible flow.

$$\frac{p}{\rho}+\frac{c^2}{2} = const \tag{2.15}$$

$$\frac{p_5}{\rho}+\frac{c_5^2}{2} = \frac{p}{\rho}+\left(\frac{c_5}{2}\frac{r_5}{r}\right)^2 \tag{2.16}$$

$$\frac{p-p_5}{\rho} = \frac{c_5^2}{2}\left(1-\frac{r_5}{r}\right) \tag{2.17}$$

where p, c, r and ρ are pressure, velocity, radius and density respectively.

2.4 Volute Performance

Detailed published information on the performance of volutes is very limited. Japikse [7] presented an incompressible flow model for a turbocharger volute. He established the volute losses through three modeling assumptions as:

(a) The kinetic energy associated with the meridional component of velocity at volute inlet is totally lost.

(b) When the flow through the volute decelerates, the pressure loss is assumed to be equivalent to that in a sudden expansion mixing loss.

(c) If the flow accelerates through the volute, no pressure loss is assumed.

Weber and Koronowski [8] developed a meanline performance prediction for volutes. Eckert [4] probably made one of the first attempts to account for friction and secondary losses in volutes. Furthermore, Iverson et al [9], Kurokawa [10] and Badie et al [11] also attempted flow prediction in volutes. A reliable prediction method will be of great help in determining the influence of the different design parameters on the volute flow and losses. However, be it a simple or complex analysis tool, the starting point is a simple flexible geometry model of the volute cross section.

2.5 Secondary Flow in Volutes

Secondary flows in volutes are understood very little. A large amount of losses in these components is suspected to be associated with secondary flows.

The flow inside a compressor volute is highly three-dimensional with swirling flow. A swirling velocity component has an important influence on the cross-wise and circumferential variation of the static pressure and velocity distribution. The volute flow is built up of layers of non-uniform total pressure and temperature in addition to the high shear forces at the center of the volute, which results in a rotational flow. One and two-dimensional methods are therefore of limited interest, and are unable to provide a reliable prediction of the circumferential pressure distortion and performance of three-dimensional volutes.

In turbine volutes, although the accelerating flow suppresses the boundary layer growth, a strong secondary flow is generated by centrifugal forces in a circulating flow field and influences the volute internal flow characteristics as well as nozzle exit ones. The flow is skewed axially and behaves in a complex three-dimensional character. This three dimensional nature is closely related to the turbine performance when a wide axial width nozzle is used with a high specific speed turbine. It is necessary to take the three-dimensional nature into account in a design to match the nozzle flow with the turbine rotor. Again, this requires a simple flexible geometry model of the volute cross section.

Chapter 3

MATHEMATICAL ELEMENTS

3.1 Bezier Curves

3.1.1 Explicit, Implicit and Parametric Functions.

In this section a brief summary is given of some of the elementary geometry concepts that are important for an appreciation of the capabilities of Bezier Curves. A more detailed description is given in reference [2], which is a good introduction to computational geometry, in particular, Bezier Curves.

The simplest way to define a plane curve is to use the explicit form $y = f(x)$ where $f(x)$ is a prescribed function of x, enabling us to tabulate and plot the function in a familiar way.

The explicit form is satisfactory when the function is a single value and the curve has no vertical tangents. However, this precludes many curves of practical importance such as circles, ellipses and other conic sections. The general implicit form of a curve is the equation

$$f(x,y) = 0 \tag{3.1}$$

where $f(x,y)$ is a prescribed function of x and y. It can easily be determined whether or not a point (x,y) lies on the curve, but the points on the curve cannot be easily calculated unless the equation can be reduced to an explicit equation for x or y. For example, the equation

$$x^2 + y^2 - r^2 = 0 \tag{3.2}$$

is the implicit function for the circle with a radius of r. The value of y is not described directly as a function of x. If we require an explicit equation, the circle must be divided into two segments with $y=+\sqrt{(r^2-x^2)}$ for the upper half and $y=-\sqrt{(r^2-x^2)}$ for the lower half.

An alternative way of describing lines and curves that treats the coordinates x and y symmetrically is the parametric form. The coordinates x and y are expressed as functions of an auxiliary parameter u, so that $x=f(u)$ and $y=g(u)$. For example the circle $x^2+y^2=r^2$ can be express parametrically by the equations

$$\begin{aligned} x &= r\times Cos(u) \\ y &= r\times Sin(u) \end{aligned} \tag{3.3}$$

where u takes values in the range of $0 \le u \le 2\pi$. Although we normally need to describe the range of the parameter u, this can be of an advantage if we want to describe only a segment of the curve. The parametric equation enables us to plot points on the curve by evaluation $x(u)$ and $y(u)$ for successive values of u.

Because in a design process one needs to determine tangents, normals, curvatures etc., a parameterization is needed that makes differentiation easy. Polynomial functions of the parameters are an obvious choice. The n^{th} order polynomial parametric equation is

$$r(u)=\sum_{n=0}^{N} u^n a_n \tag{3.4}$$

Polynomials of high degree can describe complex curves, but they require a large number of coefficients whose physical significance is difficult to grasp. Thus, they are an inappropriate tool for the designer. Moreover, the use of high degree polynomials may introduce unwanted oscillations in the curve.

The use of quadratic and cubic (second and third order) polynomial parametric functions and the physical meaning of their vector coefficients will now be illustrated.

The segment of quadratic parametric curves and surfaces are described by an equation of the form.

$$r(u) = a_0 + a_1 u + a_2 u^2 \tag{3.5}$$

It can be seen that the three-vector a_0, a_1 and a_2 are required to define the segment of a quadratic curve. In general n vectors are needed to describe a curve of degree $(n-1)$. It is usual to assign parameter values of $u=0$ and $u=1$ to the two ends of the segment, with $0 \le u \le 1$ in between. The simplest tool to determine the vector coefficients a_0, a_1 and a_2 is to specify the values of r, ${dr}/{du}$ and ${d_2 r}/{du_2}$ at the beginning of the segment.

Thus

$$\begin{aligned} a_o &= r(0) \\ a_0 + a_1 + a_2 &= r(1) \\ a_1 &= dr/du(0) \end{aligned} \tag{3.6}$$

Solving for a_0, a_1 and a_2 we get

$$\begin{aligned} a_0 &= r(0) \\ a_1 &= ({dr}/{du})(0) \\ a_2 &= r(1) - r(0) - ({dr}/{du})(0) \end{aligned} \tag{3.7}$$

By direct substitution in equation (3.6), we can obtain r in terms of $r(0)$, $r(1)$ and ${dr}/{du}(0)$. Thus

$$r = r(u) = r(0)(1-u^2) + r(1)u^2 + {dr}/{du}(0)(u-u^2) \tag{3.8}$$

Alternately, we may write the above equation in the form $r = [U][C][S]$ where U, C and S denotes the product of the three matrices as given below.

$$r(u) = \begin{bmatrix} 1 & u & u^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ \frac{dr}{du}(0) \end{bmatrix} \tag{3.9}$$

Cubic parametric equations for the definitions of curves, for e.g., surfaces in aircraft design can be described by the equation of the form

$$r = r(u) = a_0 + a_1 u + a_2 u^2 + a_3 u^3 \tag{3.10}$$

Following a similar procedure as done above for the parametric quadratic equation we can write equation (3.10) in terms of the boundary conditions $r(0)$, $r(1)$, $\frac{dr}{du}(0)$ and $\frac{dr}{du}(1)$ as follows:

$$r(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} r(0) \\ r(1) \\ \frac{dr}{du}(0) \\ \frac{dr}{du}(1) \end{bmatrix} \tag{3.11}$$

3.1.2 Bezier Curves

The vector coefficients of the parametric curves described by the equation (3.4) can be related to the position of the end points of the curve and to derivatives at these end points with respect to the parameter u. However, the derivatives with respect to the parameter u do not have an obvious meaning in the terms of curve geometry concepts such as slope and radius of curvature. Moreover, the relationship in terms of derivatives with respect to the parameter u becomes complex for higher order polynomial curve due to the many cross couplings as can be seen from the non diagonal elements in the coefficient matrix in equation (3.11).

Bezier [2] has recombined the terms of the polynomial parameterization in a way that makes the physical meaning of the vector coefficients more apparent. This is of course most important if we wish to design curves rather than fit them. In Bezier form we write equation (3.5) as follows:

$$r = r(u) = (1-u)^2 r_0 + 2u(1-u)r_1 + u^2 r_2 \tag{3.12}$$

where again $0 \le u \le 1$ for any given segment. It can be seen that this simple rearrangement of the quadratic polynomial form of the equation (3.4) with

$$\begin{aligned} a_0 &= r_o \\ a_1 &= 2(r_1 - r_0) \\ a_2 &= r_0 - 2r_1 + r_2 \end{aligned} \tag{3.13}$$

The important consequence of this rearrangement is that

$$\begin{aligned} r(0) &= r_0 \\ r(1) &= r_2 \\ dr/du\,(0) &= 2(r_1 - r_0) \\ dr/du\,(1) &= 2(r_2 - r_1) \end{aligned} \tag{3.14}$$

Thus, the curve described by Bezier form passes through the points r_0, r_1 and r_2, has a tangent at r_0 in the direction from r_0 to r_1 and has a tangent at r_2 in the direction from r_1 to r_2. The straight lines P_0P_1 and P_1P_2 form a figure called the characteristic polygon

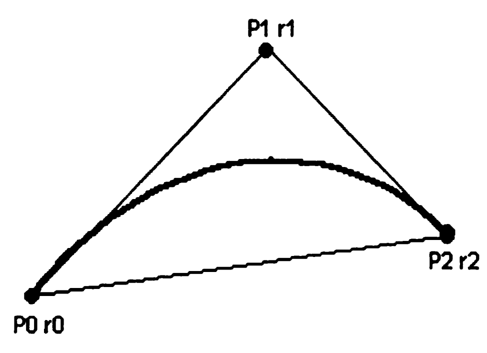


Figure 3.1 Bezier Curve

of the curve. In order to design a quadratic curve we choose the points P_0 and P_2 through which we want the curve to pass and then place P_1 so that we get the desired tangents at P_0 and P_2.

Similarly equation (3.10) can be written as

$$r(u) = (1-u)^3 r_0 + 3(1-u)^2 u r_1 + 3(1-u) u^2 r_2 + u^3 r_3 \tag{3.15}$$

where

$$\begin{aligned} a_0 &= r_0 \\ a_1 &= 3(r_1 - r_0) \\ a_2 &= 3(r_2 - 2r_1 + r_0) \\ a_3 &= r_3 - 3r_2 + 3r_1 - r_0 \end{aligned} \tag{3.16}$$

Thus the description of an n^{th} order Bezier polynomial equation could be written as

$$r(u) = \sum_{n=0}^{N} \frac{N!}{(N-n)!n!} u^n (1-u)^{N-n} r_n \tag{3.17}$$

3.1.3 Properties of Bezier Curves:

(1) The basis functions are real.

(2) The degree of the polynomial defining the curve segment is one less than the number of defining polygon points.

(3) The curve generally follows the shape of the defining polygon.

(4) The first and last points on the curve are coincident with the first and last points of the defining polygon.

(5) The tangent vectors at the ends of the curve have the same direction as the first and last polygon spans, respectively.

(6) The curve is contained within the convex hull of the defining polygon, i.e. within the largest convex polygon obtainable with the defining polygon vertices.

(7) The curve exhibits the variation diminishing property. Basically this means that the curve does not oscillate about any straight line more often than the defining polygon.

3.1.4 Changing the order of Bezier Polynomials.

When working interactively, it is often found that a particular curve segment is not sufficiently powerful (that is, does not have sufficient degrees of freedom) to adopt a desired shape. There are two possible ways to resolve this difficulty; the segment may be split into two or more segments, retaining initially the same shape or a higher order curve segment, again of the same shape may be substituted. Curve splitting is simple mathematically and may be advantageous where it is desired to use only curves of up to a certain order. Increasing the order of a curve does not change the shape of the Bezier curve. The following easily proved procedure increases the order from N to $N+1$.

$$r*\left(\frac{n}{N+1}\right)=\frac{1}{(N+1)}\left[ir*\left(\frac{n-1}{N}\right)+(N+1-n)r*\left(\frac{n}{N}\right)\right] \tag{3.18}$$

where $0 \leq n \leq N+1$ and $r*\left(\frac{n}{N}\right)$ represents the n^{th} order Bezier polygon vector of an N^{th} degree Bezier Polynomial.

This process of changing the order of the Bezier polynomial is visually implemented by adding a control point to the Bezier curve. Thus, by adding a node to the Bezier polynomial cross section we increase the order of the Bezier Polynomial.

3.1.5 Joining Bezier Segments

Additional flexibility can be obtained by subdividing a curve into two or more Bezier curves, which combined are identical to the original curve. However, slope and curvature continuity has to be maintained when joining Bezier curves. Continuity conditions between adjacent Bezier curves are simply specified. If one Bezier curve $P(t)$ of degree n is defined by B_i vertices and an adjacent Bezier curve $Q(s)$ of degree m is defined by C_i vertices, then the first derivative of continuity at the joint between the curves is given by

$$P'(1) = Q'(0) \tag{3.19}$$

Since positional continuity is implied at the joints, $C_0 = B_n$. Thus, the tangent vector directions are the same if the three vertices B_{n-1}, $B_n = C_0$, C_1 are collinear; i.e. $B_n(C_0)$ need only lie somewhere on the line between B_{n-1} and C_1. If both direction and magnitude of the tangent vectors at the joint are to be equal, then $C_0 = B_n$ must be the midpoint of the line joining B_{n-1} and C_1.

Figure 3.2 illustrates this for $n = m = 3$, i.e. for two Bezier curves.

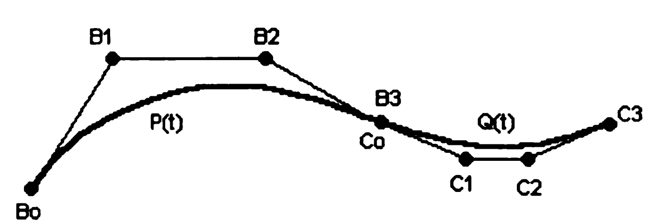


Figure 3.2: First Derivative continuity between cubic Bezier curves

3.1.6 Lines and Arcs with Bezier Curves

As a Bezier curve is defined by its defining polygon, it is possible to generate elementary geometrical shapes, such as lines and arcs with Bezier curves.

Thus, a line would require a minimum of two control points (nodes) to be defined in its entirety, as illustrated in Figure 3.3.

Figure 3.3: Line with Bezier curves

Similarly, an arc would be defined by a control polygon consisting of a minimum of three control points. This is illustrated in Figure 3.4.

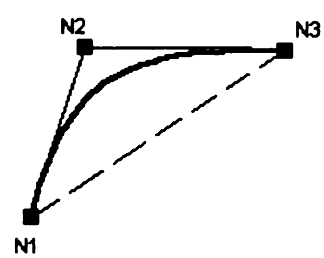


Figure 3.4: Arc with Bezier curve

In a similar way the following have been defined

Circle:

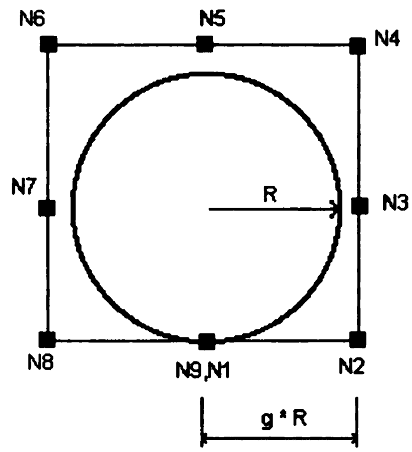


Figure 3.5: Circle with Bezier curve

Ellipse:

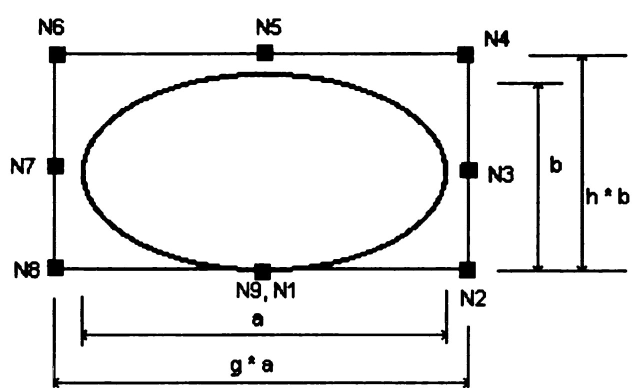


Figure 3.6: Ellipse with Bezier curve

Parabola:

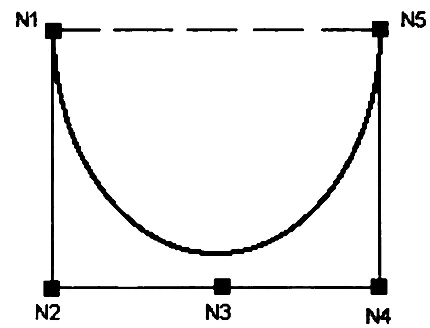


Figure 3.7: Parabola with Bezier curve

It is seen from the above figures, Figure 3.5 for a circle and Figure 3.6 for an ellipse, that when it is desired to draw a circle of radius R, it is necessary to define the control points at $g * R$, where g is a scalar, also known as the correcting factor. The 'circle' thus generates not an exact circle but a close approximation to the circular arc, the maximum deviation from the mean radius being $\pm 0.5\%$.

3.2 Volute Cross Sections

The volutes for turbines and compressors are designed using Bezier curves. Each cross section of the volute is made up of different segments comprising of arcs, straight lines, sections of a circle and sections of parabolas. Each volute comprises of a set of segments that vary only dimensionally across the volute. The different types of volute cross sections designed are as listed below.

Compressors:

(1) Circular Constant Centroid

(2) Circular Constant Inner Diameter

(3) Elliptical Constant Centroid

(4) Elliptical Constant Inner Diameter

Turbines:

(5) Open Flow Turbines

(6) Twin Flow Turbines

Each cross section is defined by certain parameters, such as the width of the volute (S), radii (R) of circles and arcs, etc. The values of these parameters vary for the volute cross sections from a section at 30^0 to a section at 360^o. However, the shape of the cross sections remains unchanged. The values of these parameters are obtained from the Area by Centroid radius distribution curve and the area at the throat [1].

Below are typical geometrical definitions that completely define the cross section of the volute.

(1) Compressor Circular Constant Centroid:

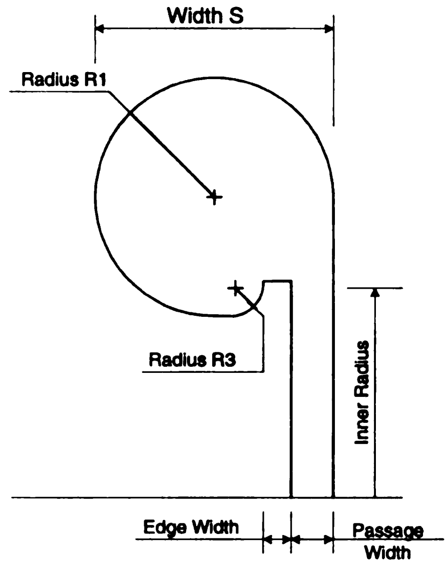


Figure 3.8: Geometry Compressor CCC -

(2) Compressor Circular Constant Inner Diameter:

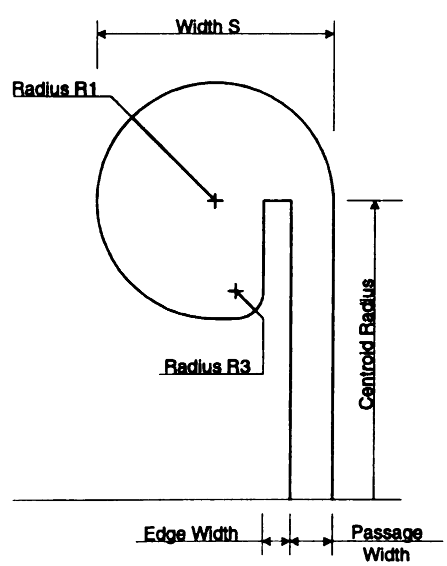


Figure 3.9: Geometry Compressor Circular Constant ID

(3) Compressor Elliptical Constant Centroid:

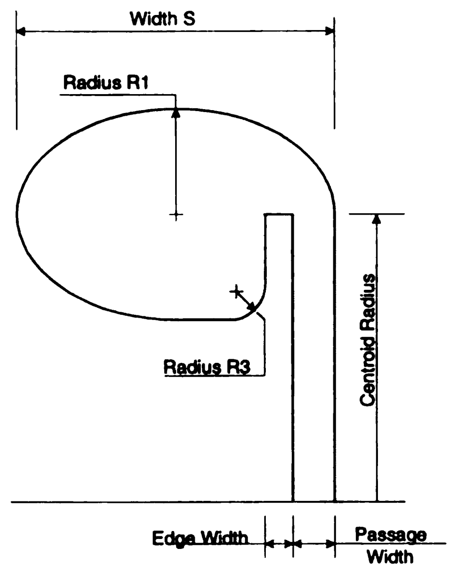


Figure 3.10: Geometry Compressor Elliptical Constant Centroid

(4) Compressor Elliptical Constant Inner Diameter:

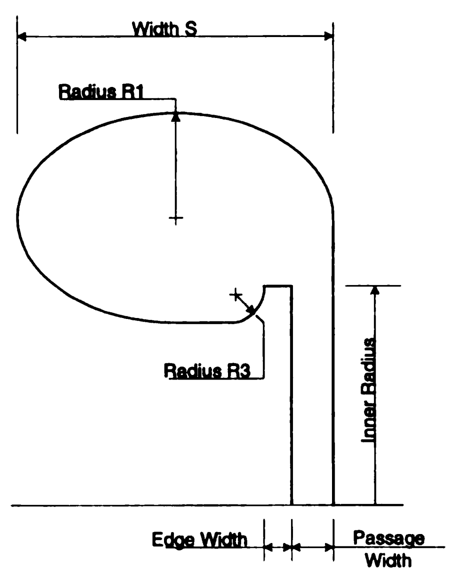


Figure 3.11: Geometry Compressor Elliptical Constant ID

(5) Turbine Open Flow:

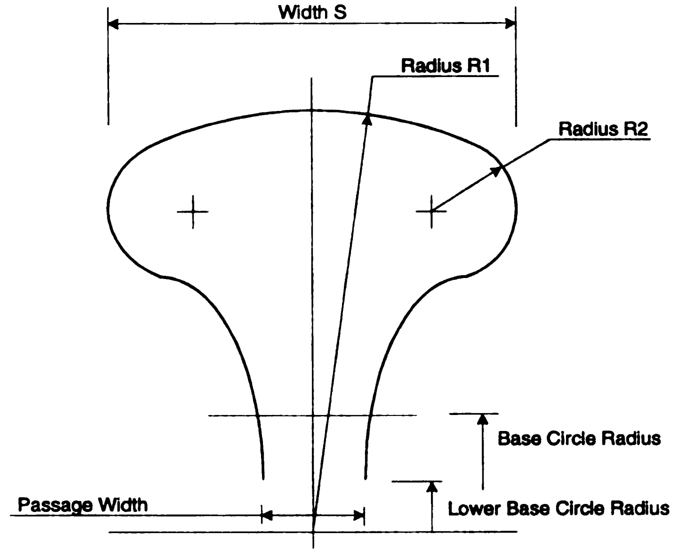


Figure 3.12: Geometry Open Flow Turbine

(6) Turbine Twin Flow:

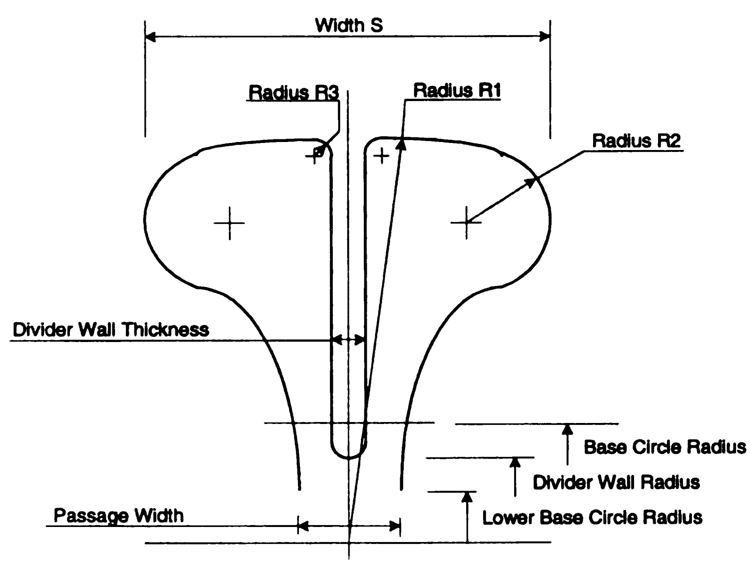


Figure 3.13: Geometry Twin Flow Turbine

3.3 Volute Cross-Sections with Bezier Curves.

The volutes for compressors and turbines are designed using Bezier curves. Each volute cross section is formed by joining different elementary components of geometry; namely lines, circles, parabolas and ellipses. Each component is called a segment. These segments are generated individually with Bezier curves (Refer Section 3.1 E) and the required volute shape is obtained by joining these segments together. The points of tangency are maintained between two adjacent segments such that all the segments taken together appear as a single curve.

The formulation of each volute cross section from its segments using Bezier Curves are as given below.

(1) Compressor Circular Constant Centroid.

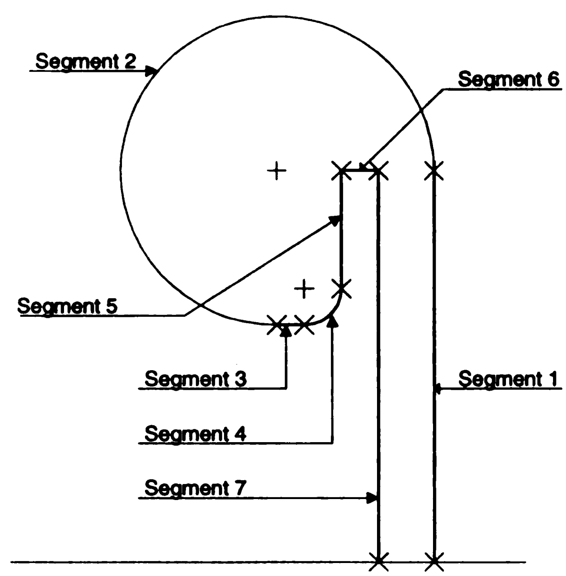


Figure 3.14: Segments, Compressor Circular Const. Centroid

The profile consists of the following segments:

Segment 1: Straight line

Segment 2: Span of 270° of a circle of radius R_1

Segment 3: Straight line

Segment 4: Span of 90° of a circle of radius R_3

Segment 5: Straight line

Segment 6: Straight line

Segment 7: Straight Line

(2) Compressor Circular Constant inner Diameter.

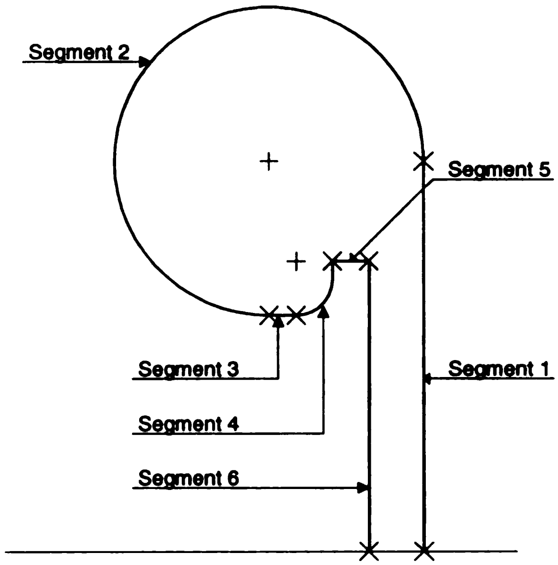


Figure 3.15: Segments, Compressor Circular Const. ID

The profile consists of the following segments:

Segment 1: Straight line

Segment 2: Span of 270° of a circle of radius R_1

Segment 3: Straight line

Segment 4: Span of 90° of a circle of radius R_3

Segment 5: Straight line

Segment 6: Straight line

(3) Compressor Elliptical Constant Centroid.

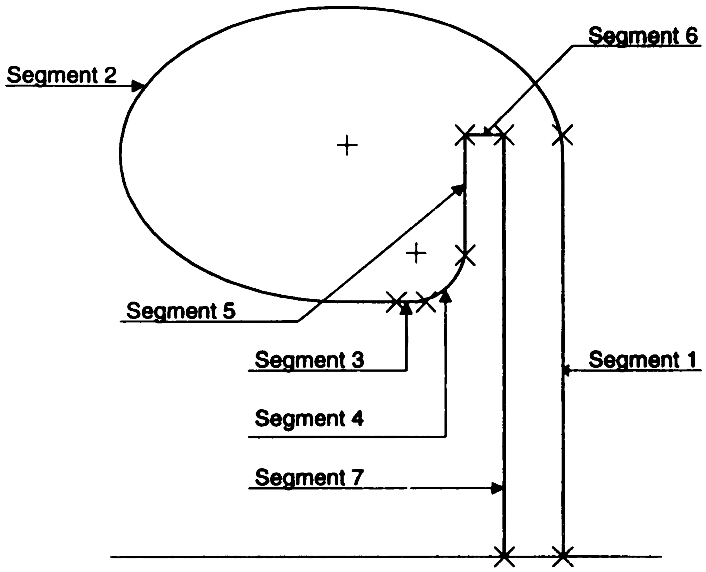


Figure 3.16: Segments, Compressor Elliptical Const. Centroid

The profile consists of the following segments:

Segment 1: Straight line

Segment 2: Span of 270° of an ellipse with width S and height $2 * R_1$

Segment 3: Straight line

Segment 4: Span of 90° of a circle of radius R_3

Segment 5: Straight line

Segment 6: Straight line

Segment 7: Straight Line

(4) Compressor Elliptical Constant Inner Diameter:

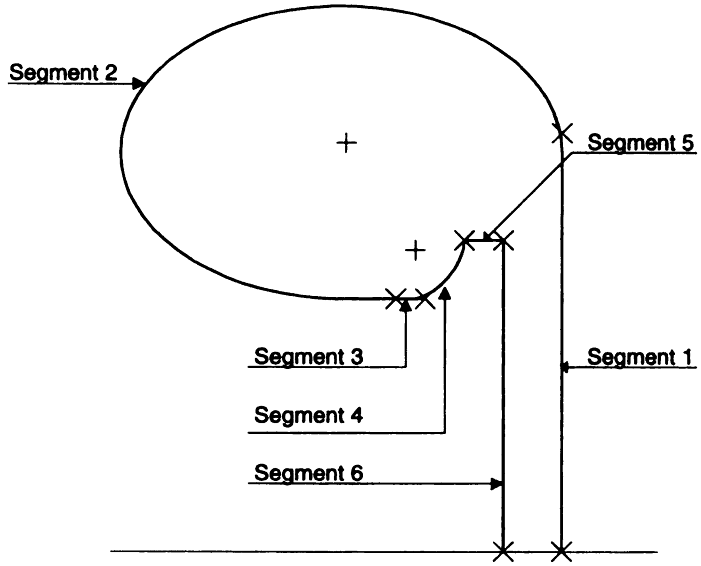


Figure 3.17: Segments, Compressor Elliptical Const. ID

The profile consists of the following segments:

Segment 1: Straight line

Segment 2: 270° of an ellipse with width S and height $2 * R_1$

Segment 3: Straight line

Segment 4: 90° of a circle of radius R_3

Segment 5: Straight line

Segment 6: Straight line

(5) Turbine Open Flow:

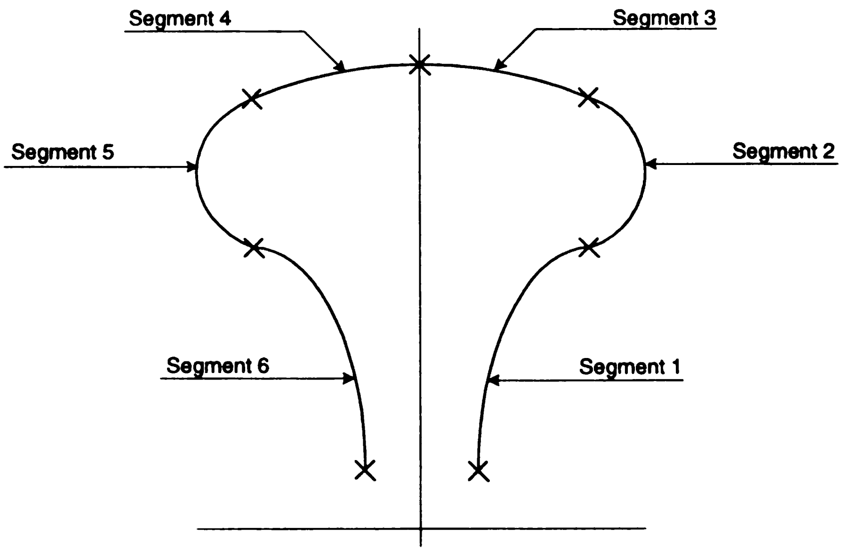


Figure 3.18: Segments, Open Flow Turbine

The profile consists of the following segments:

Segment 1: Half a parabola

Segment 2: Span of approximately 120° of a circle of Radius R_2

Segment 3: Span of approximately 30° of a circle of Radius R_1

Segment 4: Span of approximately 30° of a circle of Radius R_1

Segment 5: Span of approximately 120° of a circle of Radius R_2

Segment 6: Half of a parabola

(6) Turbine Twin Flow:

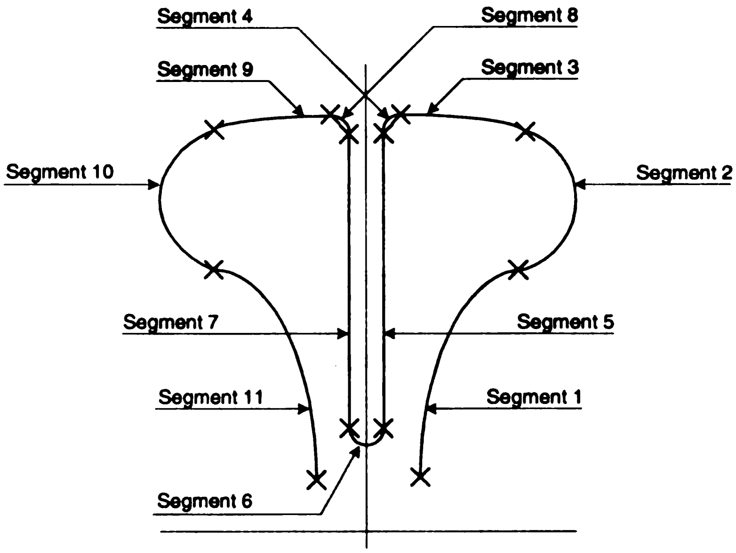


Figure 3.19: Segments, Twin Flow Turbine

The profile consists of the following segments:

Segment 1: Half a parabola

Segment 2: Span of approximately 120° of a circle of Radius R_2

Segment 3: Span of approximately 28° of a circle of Radius R_1

Segment 4: One quarter of a circle of Radius R_3

Segment 5: Straight line

Segment 6: Semi-circle of diameter equal to the divider wall thickness

Segment 7: Straight line

Segment 8: One quarter of a circle of radius R_3

Segment 9: Span of approximately 28° of a circle of Radius R_1

Segment 10: Span of approximately 120° of a circle of Radius R_2

Segment 11: Half of a parabola

3.4 Diffusers/Inlet Section.

For compressors the section through which the gas leaves the compressor is known as the diffuser. This is used to diffuse the flow, i.e., converting the kinetic head of the fluid into pressure head. Similarly, the section into which the gas enters a turbine is called the Inlet Section. This is used to accelerate the flow of the gas, which then impinges on the blades of the turbine. The three general cases (i.e., compressors, open flow turbines and twin flow turbines) are described below.

[*Note:*] The term 'diffuser' as used above should not be confused with the diffuser as in the overall turbo-machine.

3.4.1 Compressors

We shall consider the case of a compressors first. The diffuser section originates from its first cross section at 360° of the volute, which is as shown in Figure 3.21

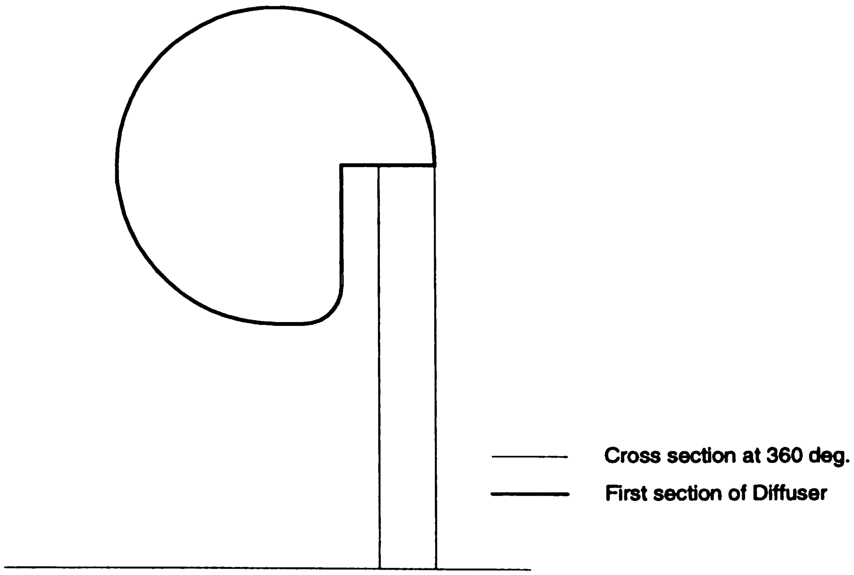


Figure 3.20: First Section of Diffuser for Compressors

The second section, i.e. the exit of the diffuser is of a circular cross section of radius R, refer Figure 3.22

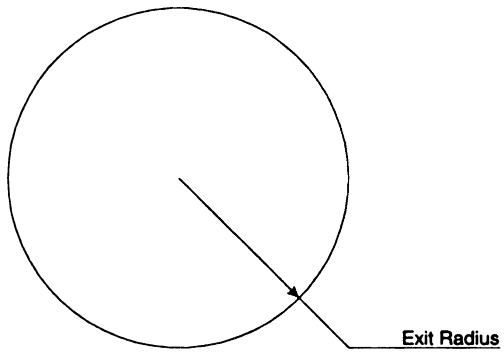


Figure 3.21: Second Section of Diffuser for Compressors

The diffuser thus transforms from a cross section shape of Figure 3.21 to a circular cross section as shown in Figure 3.22. This transformation is linear through the length of the diffuser.

3.4.2 Turbine Open Flow

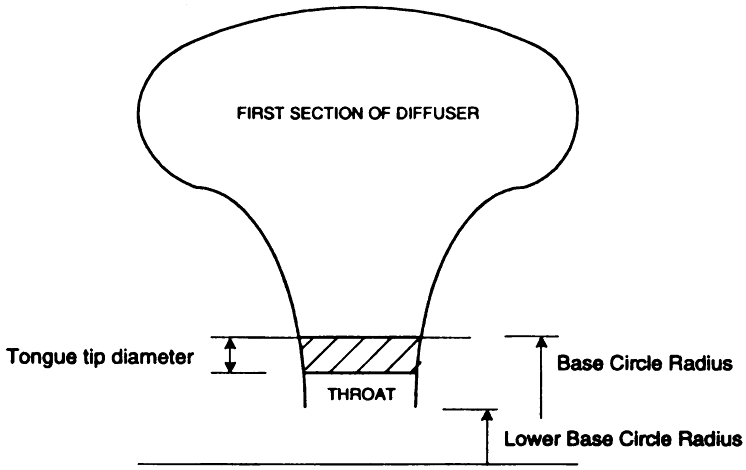


Figure 3.22: Open Flow Turbine Cross Section at Throat

For an open flow turbine the base of the inlet section (at 360°) is designed up to the base circle radius as illustrated in Figure 3.23. The first section of the Inlet Section is rectangular in shape and is dimensioned as in Figure 3.24, the inlet section is formed by varying the cross section varies linearly from the cross section at 360° (Figure 3.23) to the first section of the Inlet Section (Figure 3.24) over the length of the Inlet Section.

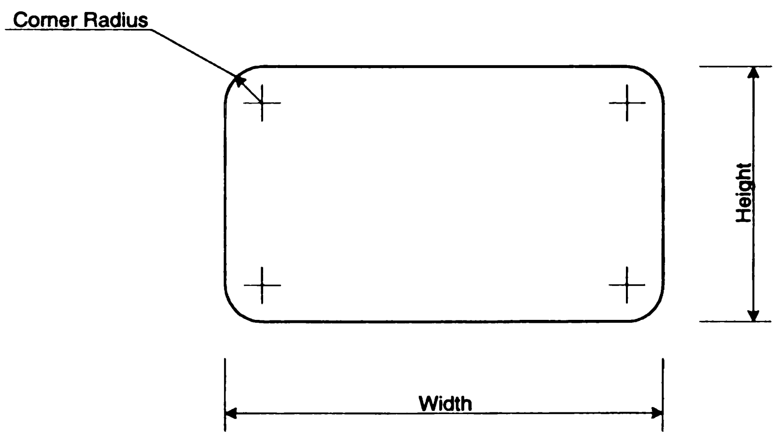


Figure 3.23: Open Flow Turbine Inlet

3.4.3 Turbine Twin Flow.

The Inlet Section for the Twin Flow turbines are slightly different. The inlet section comprises of two passages whose cross sections are mirror images of each other across the dividing plane. We shall consider just one of these two Inlet sections passages. The second section of the Inlet Section i.e. at the cross section 360^{o} of the volute, has the same shape as the cross section of the volute, with the base at the Base Circle Radius. This is as shown in Figure 3.25.

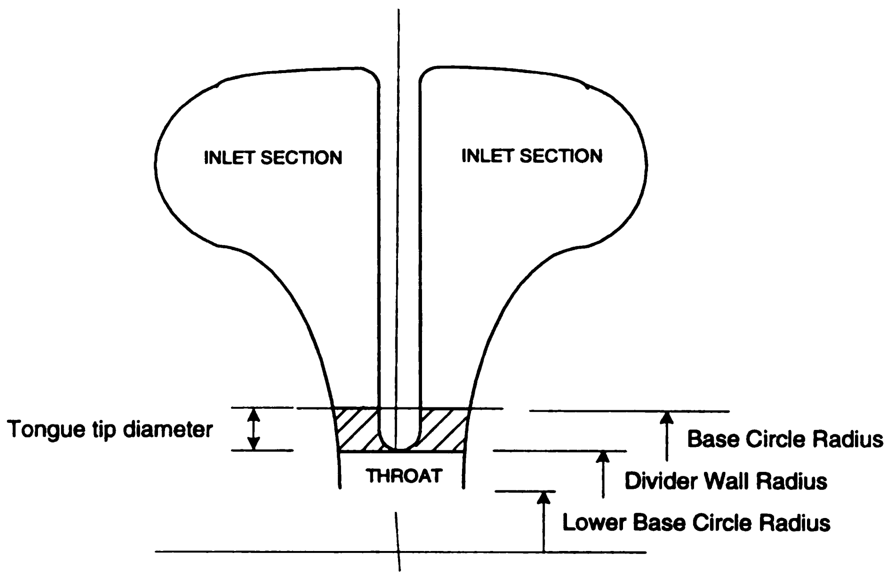


Figure 3.24: Twin Flow Turbine Cross Section at Throat

Each passage of the Inlet Section for the twin flow turbines have a rectangular cross-section at the inlet, the same as in open flow turbines. The complete cross section for the twin flow turbine at the turbine inlet is as shown in Figure 3.26.

Here as well, the cross section transforms linearly for each passage from the cross section as shown in Figure 3.25 to the section as shown in Figure 3.26 over the length of the inlet section.

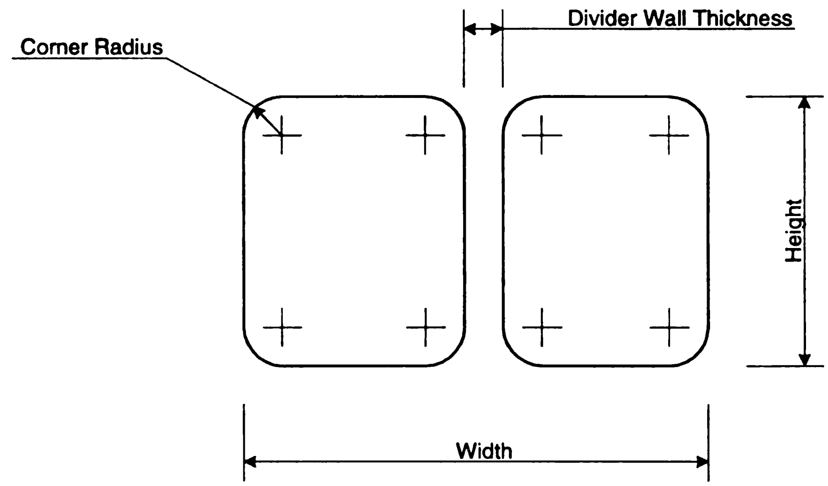


Figure 3.25: Twin Flow Turbine Inlet

3.5 Generating the Volute.

The volute is divided into three main regions

(1) The Diffuser (for compressors) or Inlet Section (for turbines)

(2) The Volute

(3) The Throat (only for turbines)

The volute spans from 0° to 360° for compressors and from 30° to 360° for turbines. At every 30° interval, a 2-D cross-section is generated using Bezier curves. The 2-D Bezier cross sections are generated by specifying the control points that are calculated from the geometrical data. Thus, by defining a cross section in 2-D and knowing its angle of location on the volute, we generate the 3-D model of the volute. The region between sections are then interpolated. The method of interpolation is such that the x and y coordinates of the node points are interpolated between sections, depending on the number of sections required (say n). Thus, the angle between two adjacent sections would be $360/n$. Using these interpolated values of the node points, a 2-D Bezier cross section is generated. Its angle of location on the volute being the third axis, a 3-D model of the volute is thus generated having n number of cross sections.

The diffuser/Inlet Section is generated in a similar way as the volute and is defined in the preceding article (i.e. Article 3.4)

The region between 0° and 30° for a turbine is known as the Throat. This is the area that wraps under the Inlet section and back into the volute. The cross section at 30° is the same as that generated in the volute. The section at 0° is obtained as explained. It is a part of the cross section at 360°. The upper surface is at a position of the base circle radius minus twice the tongue tip radius (Figure 3.23 and Figure 3.25). Thus knowing these two sections in 2-D the sections between them are linearly interpolated to give 2-D sections. Then with the angle at which each section lies, the 3-D model of the throat is created.

Thus, these three regions put together (i.e. the Volute, Inlet Section/Diffuser and Throat) define the volute in its entirety.

3.6 AREA CALCULATIONS:

The area is calculated numerically using Green's Theorem, which is defined as

$$Area = \oint_C Mdx + Ndy \tag{3.20}$$

The area calculated for each type of volute is as shown below by the shaded section.

(1) Compressor Constant Inner Diameter.

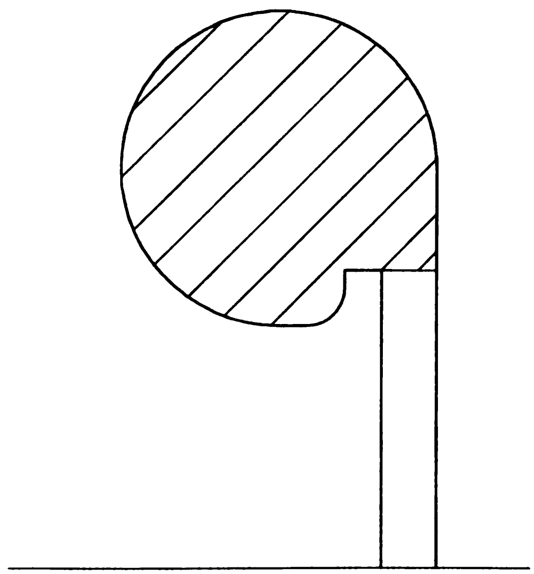


Figure 3.26: Area, Compressor Circular Constant ID

(2) Compressor Circular Constant centroid.

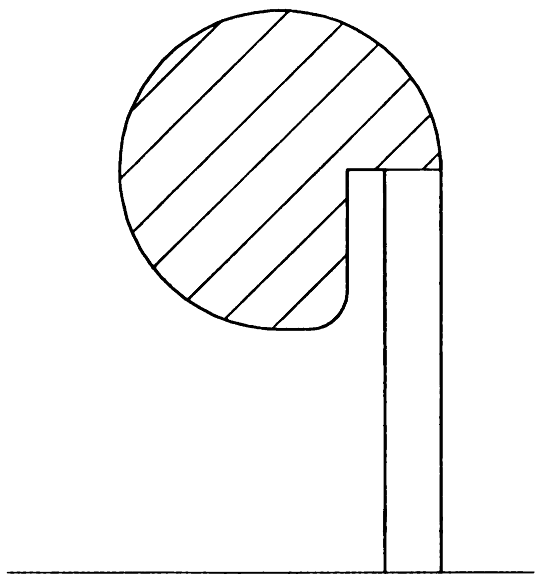


Figure 3.27: Area, Compressor Circular Constant Centroid

(3) Compressor Elliptical Constant Inner diameter.

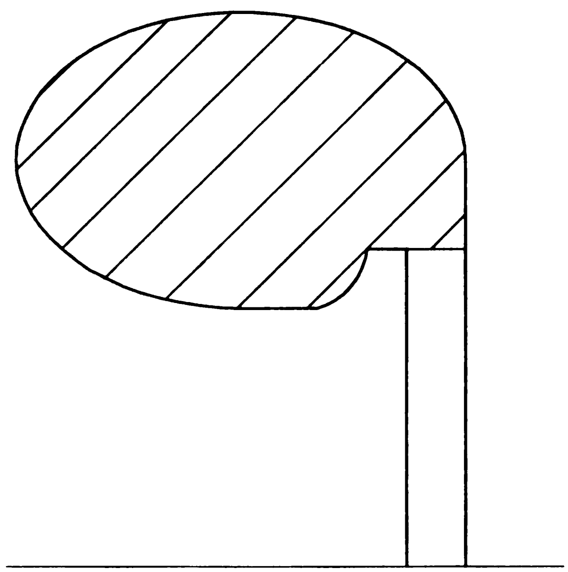


Figure 3.28: Area, Compressor Elliptical Constant ID

(4) Compressor Constant Centroid radius.

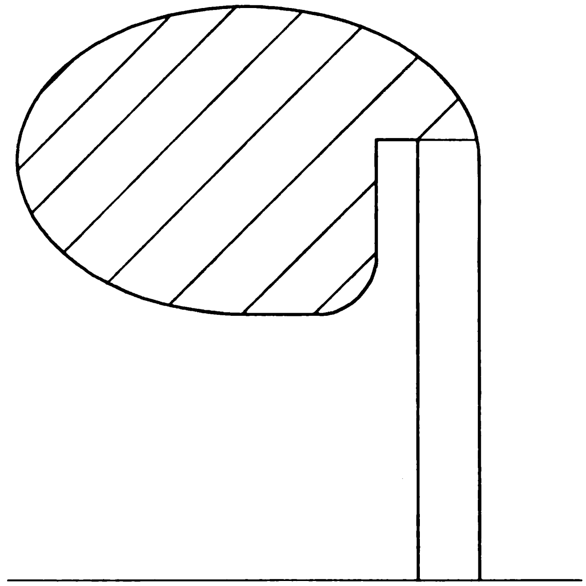


Figure 3.29: Area, Compressor Elliptical Constant Centroid

(5) Turbine Open Flow

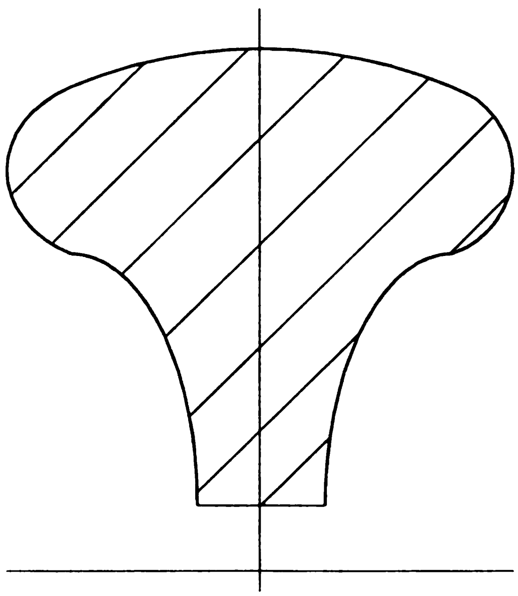


Figure 3.30: Area, Open Flow Turbine

(6) Turbine Twin Flow

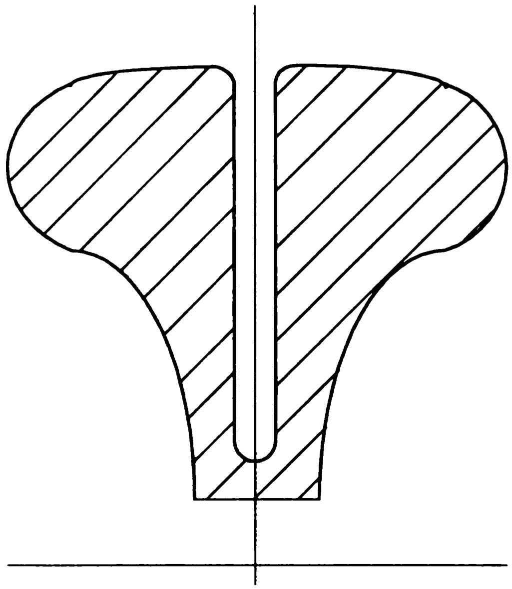


Figure 3.31: Area, Twin Flow Turbine

3.7 Area/Centroid Radius Calculation.

From the above section we calculate the area. The centroid radius is calculated numerically with the following formula. The centroid radius is about the 'x' axis.

$$y_{CR} = \int \frac{\bar{y}.dm}{M} \tag{3.21}$$

where:

M = mass

Y = Mean distance from the 'x' axis.

Thus, once the centroid radius is obtained, the ratio of area to the centroid radius is then plotted for the cross sections from 30° to 360° at 30° intervals.

Chapter 4

VOLUTE DESIGN - THE PROGRAM

4.1 Volute Design Features

(1) Drawing Bezier curves:

Each cross-section of the volute is generated using Bezier curves. As seen in the previous article 3.1, Bezier curves give us a great deal of freedom in the shapes we generate and the ease in increasing the degree of the equation. This program uses Bezier curves to generate the 2-D cross sections for the different types of volutes.

(2) Specifying Geometry:

When we start a new design, we begin with some defining parameters such as the A/R ratio, the area at the throat, base circle radius, the passage width, wheel diameters, corner radii, diffuser dimensions and overall width of the volute. The program provides a user interface to input all the geometrical and other data, processes them and generates the required volute.

(3) Changing Node Positions:

The shape of the Bezier curve is determined by its control polygon. The control polygon is made up of control points also known as node points. Thus, when it is desired to change the shape of the curve one has to change the node points. The program visually implements this by allowing the user to manipulate the node points, which is done by the user clicking the node point, dragging the cursor to the point desired, forming a new control polygon and thus generating a new Bezier curve.

(4) Adding and Deleting Nodes:

Very often, it is desired to increase the degree of freedom of a curve. This is achieved by increasing the control points in the control polygon for the Bezier curve. The program allows the user to add a node point, increasing the order of the curve. However increasing the order of the curve does not change the shape of the curve. Similarly, the user is allowed to delete a node or control point. Deleting a control point decreases the order of the curve, making a curve of order n, a curve of order $n-1$. This results in the change of shape of the curve and thus the volute.

(5) Referencing Nodes:

While modifying the shape of the volute using Bezier curves, the user views only one cross-section at a time. The user has a choice of referencing certain number of cross sections collectively while modifying a selected cross section. These options are:

(a) Reference all cross sections.

This allows the user to simultaneously modify all the sections of the volute from 30^{o} to 360^{o} while modifying any one cross-section between the two cross sections.

(b) Reference the current cross section.

This allows the user to reference only that cross-section on which one working. The remaining sections are not affected by ones changes to the working cross section.

(c) Reference between sections.

The user specifies the two outermost cross sections between which changes are to be reflected in.

(6) Hiding/Showing Nodes:

The program allows the user to toggle the display of nodes, while working on a cross-section on the screen.

(7) Area:

The program calculates the area of the displayed cross-section of the volute. The boundaries and formula for the area calculated could be viewed from section 3.6.

(8) Area Curves:

The program plots two types of area curves and displays them on a new screen, which could then be printed as output. The first curve plotted is an area distribution curve, which is the area against cross-section angle. The second curve is the plot of the ratio of the area by centroid radius against the cross-section angle. These are important curves in the design of a volute.

(9) Viewing Different Sections:

The program allows the user to switch between cross sections from 30° to 360°.

(10) Zooming:

The zoom feature allows the user to zoom into or out of a cross-section view to enable one to make minor adjustments to the shape of the volute cross section.

(11) 3-D Views:

A three-dimensional object is created using the 2-D cross-section with its angular location on the volute. Two forms of the 3-D model are available: a wire mesh model and a solid view model. There are a few features with these models. The user can control the grid size, view the model from a desired angle of projection, rotate the model with the help of scrollbars and zoom into or out of the model. The model is also divided into the Diffuser/Inlet section and Volute. Both these elements can be viewed together or individually.

(12) Dimensions:

A list of the geometrical data that define the volute cross sections is tabulated for the drafting or manufacturing of the volute.

(13) Files:

The program generates an output file of the model. This file is in a text format that can be read by other programs. The output file contains the coordinates of the model, which are specified either in rectangular, polar or in spherical coordinates

(14) Documenting a Design:

This feature allows the designer to document a design. Documenting means saving all the necessary design and geometrical data that could be retrieved and modified in the future. Thus, the designer could proceed to improve an already optimized design, without losing the data of the optimized design.

(15) Open/New/Run a Design:

This feature gives the user the option of starting a new design, in which all the parameters are initialized to begin a new design. Alternatively, it can also allow the user to open a previously documented design. *Run* is the same as documenting a design, as described in (N).

(16) Status Panel:

This status panel is displayed at the bottom of the screen. It displays online help and details about the current design project. Some of the information it displays about the current project are the type of volute, node numbers, design run number, displayed volute cross-section and the date and time.

4.2 Forms

4.2.1 Form PROFILE

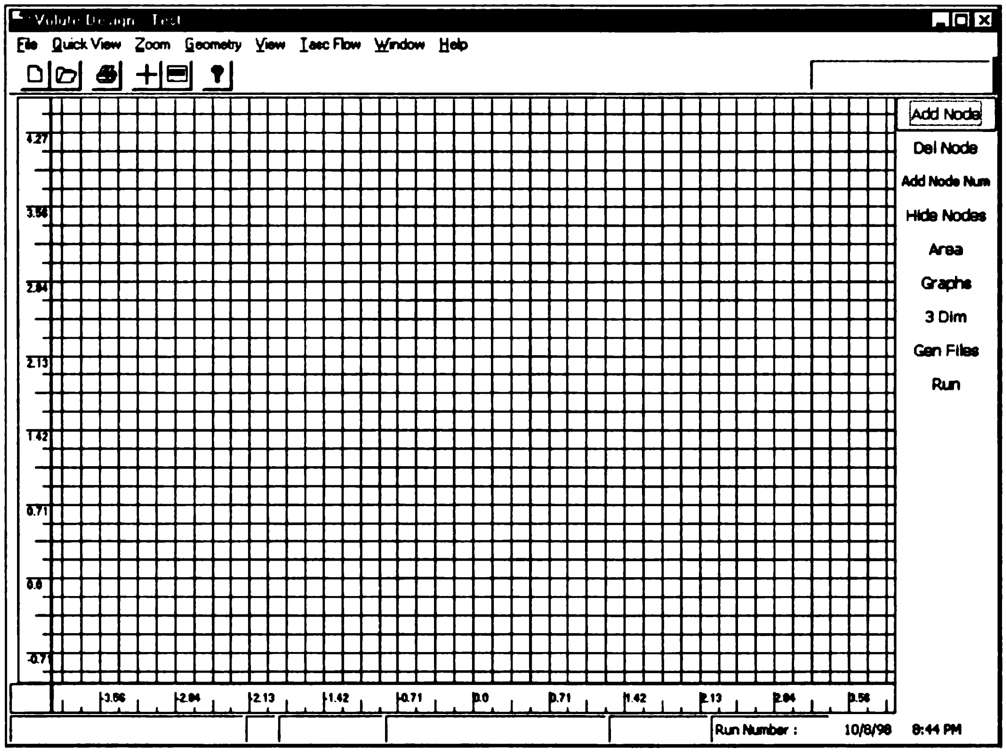


Figure 4.1: Form Profile

The form profile is the platform for the design tool *Volute Design*. It is the form that supports the volute to be designed, allows for modification of the volute cross-section and is the general interface for the various features of the program. The form consists of a working area (the grid) in which the volute cross sections are drawn. In this area the volute can be generated, modified and documented. Rulers are placed on the two axis of the grid. Above the grid lie the menu and the toolbar. On the far right of the toolbar is a display panel. This panel displays the location of the cursor while moving nodes and while zooming. It also displays some features like ‘Zooming’ or ‘Delete node’ etc. while those commands are executed. On the right of the grid lie the command buttons. Below the grid lies the status bar, which displays the help, and detailed status of a project design.

The menus command buttons and status bar are now discussed in detail.

4.2.1.1 Menus

1) *File*

K File

- New
- Open
- Save
- Redraw
- Print
- Exit

New: Starts a new design. Clears the screen and brings up the ***New Design*** form

Open: Opens an existing design. Brings up the ***Open Design*** form

Save: Saves the current design

Redraw: Clears and redraws the screen. Settings remain the same

Print: Prints the form

2) *Quick View*

Quick View is a text viewer that allows you to view the files related to the geometry and the coordinate file of the volute. Hard copies (prints) could be taken of the files.

K Quick View

- Geometry
- Output File

- Rectangular
- Circular
- Spherical

Geometry: View the geometry file

Rectangular: View the rectangular coordinate file

Circular: View the polar coordinate file

Spherical: View the spherical coordinate file

[*NOTE:*] The Output files can only be viewed after they have been generated i.e. by running the *Generate File*, as explained later. Else, these menus are disabled.

3) *Zoom*

Enlarges or shrinks the screen size.

K Zoom

- Zoom Box
- Zoom In
- Zoom Out
- Restore

Zoom Box: The zoom box can be created by clicking down on the required top left corner of the area to be zoomed and dragged till the required zoom box is drawn. Releasing the mouse zooms the screen accordingly.

Zoom In: Zooms in the picture with a fixed factor

Zoom Out: Zooms out the picture with a fixed factor

Restore: Restore restores the original scale

4) ***Geometry***

Displays the geometry form that is used to specify the geometrical and aerodynamic data for the design of a volute.

5) ***View***

K View

- Cross Section At
  - 30 ~ 360
- Dimensions

Cross Section At: $(30^{\circ} \sim 360^{\circ})$ Displays the cross-section of the volute at the desired cross-section angle. That is any cross-section angle between 30° and 360°.

Dimensions: Displays the *Dimension* form. This is the form which you can view the geometrical data (radii R_1, R_2, R_3 and width S) necessary for drafting or manufacturing the volute.

6) ***TASCFlow***

Displays the TASCFlow input form. This form is to be used as an interface to TASCFlow or any other CFD (Computational Fluid Dynamics) package.

7) ***Window***

Displays a list of all the open windows. When the desired window is clicked, it becomes the active window.

8) ***Help***

Contains the help files

4.2.1.2 Command Buttons

1) ***Add Node:***

Brings up the *Add Node* form. This form is used for adding a node (a control point in the Bezier polygon) to the existing cross-section profile.

2) ***Del Node:***

Deletes a node from the volute cross-section profile. On clicking this command, the mouse pointer changes to an 'up' arrow, which, when placed above the desired node to be deleted, asks for a conformation before deleting the node.

[*NOTE:*] For the *Add Node* and *Del Node* commands, the changes in adding or deleting a node is reflected in the all cross sections.

3) ***Add/Rem Node Num:***

This button toggles between *Add Node Num* and *Del Node Num*. It is used for displaying or removing the node numbers on the nodes.

4) ***Hide/Show Nodes:***

This button toggles between *Hide Nodes* and *Show nodes*. It is used for displaying the nodes (i.e. Show Nodes) or for removing the nodes from the screen (i.e. Hide nodes).

5) ***Area:***

Calculates and displays the area in a message box for the current volute cross-section on the screen.

6) ***Graphs:***

Displays the two graphs that are calculated by the program, namely

(a) Area Vs the Cross Section Angle, and

(b) Area/Centroid Radius Vs Cross-Section Angle.

These graphs are for the current volute being designed.

7) *3-Dim:*

This button displays the 3D view of the volute. It initially displays a splash screen where the user specifies the grid size. It then calculates the grid and plots the volute on the *Three Dimension* form. The splash form also acts as an update bar, informing the user of the current progress of execution of the operation.

8) *Gen Files:*

This command is used for generating an output file of the coordinates of the volute profile. It gives the user the choice of the grid size and a choice of the file type (i.e. rectangular coordinates, polar coordinates or spherical coordinates) for the output file.

9) *Run:*

This is used for documenting a design. When this command is executed, the program saves all the necessary data required to save the design. Thus, the same design could be opened at a later date and modifications or output files generated, for the design. [*NOTE:*] Where executing the *Run* command, the program saves only the critical data, thus if any or all output files are required, the design would have to be loaded using *Open* on the file menu and then generate the output files.

4.2.1.3 Status Bar

The status bar is designed to provide the user with information related to the design. It is divided into seven panels, each displaying information about the design project.

Panel 1: Displays the help for the command buttons

Panel 2: Displays the Node Number above that which the mouse is

Panel 3: Describes the volute types (i.e. compressors or turbines)

Panel 4: Displays the shape and geometry of the volute. That is for compressors would display either Circular or Elliptical and their geometry i.e. Constant Centroid or Constant Inner Diameter for compressors, or for a turbine whether it is an Open Flow or a Twin Flow turbine.

Panel 5: Displays the current cross-section being viewed

Panel 6: Displays the current run number of the design

Panel 7: Displays the current date and time

4.2.2 Form GEOMETRY

The *Geometry* form is the user interface that allows the user to specify the type and geometrically define the volute to be designed. The geometry data consists of the geometry for the volute as well as the diffuser/inlet section. This form is also used to set certain parameters (such as dimensioning units, type of volute, axis-offset, etc.) that are used in the design process. Overall, this form is divided into six sub screens (*General*, *Shape*, *Compressor*, *Turbine*, *Axis* and *Diffuser*) that one navigates between with the use of the *Tab* control. Each sub screen and its features shall now be described.

4.2.2.1 General

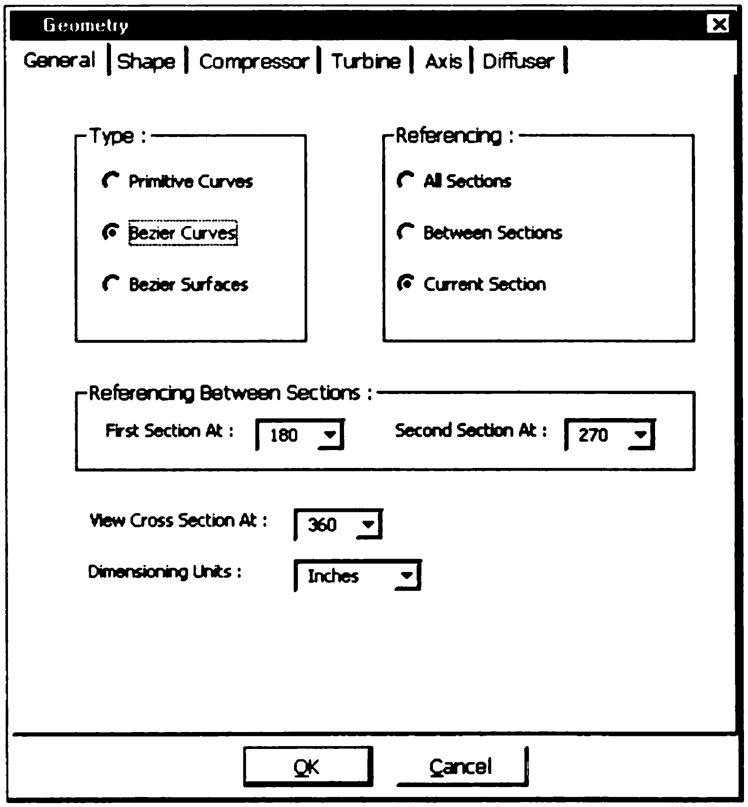


Figure 4.2: Form Geometry, Tab General

1) ***Type***

This choice specifies the type of curves that will be used to generate the volute cross-section profile. There are two choices:

(i) Generating the volute by *Primitive Curves* and

(ii) Generating the curve by *Bezier Curves*

Generating the volute cross-section profiles with *Primitive Curves* involves drawing the cross-section using solid geometry. Here the cross sections are divided into segments in the way as explained in section 3.2. These segments are then drawn by equations that describe the corresponding circles, parabolas, arcs and straight lines.

However generating the volutes with *Primitive Curves* does not provide the user with the freedom to modify the volute.

Generating the volute cross-section profiles with *Bezier Curves* involves drawing the cross-section profiles with Bezier curves. The way each cross-section is drawn is described in section 3.3 Bezier curves allow the user to repeatedly modify the cross sections to obtain an optimized design.

2) ***Referencing***

Referencing involves to what extent a modification in the current cross section, should be reflected in the remaining part of the volute.

When the choice selected is *All Sections*, the modifications made on the current cross-section will be reflected in all cross sections i.e. between cross-section 30^{o} and cross-section 360^{o}.

When the choice selected is *Between Sections*, the *Between Sections* frame gets enabled, allowing the user to select the two limiting cross sections across which modifications to one cross sections should be reflected. Thus for example, if a user specifies between sections 180^{o} and 360^{o}; then when a cross-section between 180^{o} and 360^{o} is modified, modifications are reflected in all cross sections between 180^{o} and 360^{o}.

When the choice selected is *Current Section*, modifications are only reflected in the current cross section.

3) ***View Cross Section At***

To specify which cross-section (i.e. any cross-section between 30^{o} and 360^{o}) becomes the current cross section.

4) ***Dimensioning Units***

To specify the dimension units of the design project i.e. Inches, mms or cms.

4.2.2.2 Shape

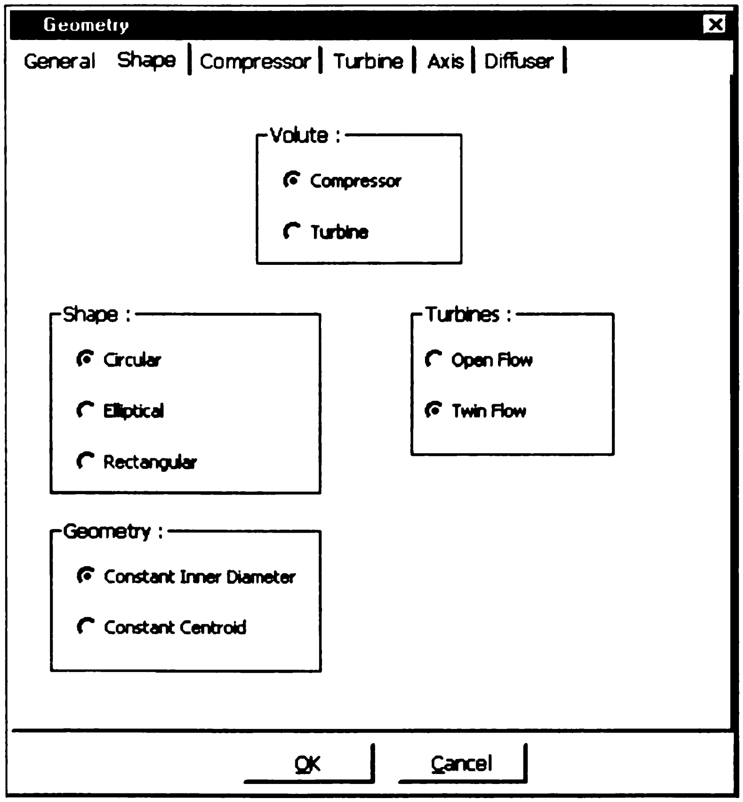


Figure 4.3: Form Geometry, Tab Shape

Shape, allows the user to specify the type of volute required to be generated. With a combination of the volute type, volute shape and volute geometry, the user can choose one of six options of a volute to be generated. These six options are as listed below:

(1) Compressor Circular Constant Inner Diameter

(2) Compressor Circular Constant Centroid

(3) Compressor Elliptical Constant Inner Diameter

(4) Compressor Elliptical Constant Centroid

(5) Turbine Open Flow

(6) Turbine Twin Flow

4.2.2.3 Compressor

This sub form is used for inputting the geometrical data for the design of a compressor.

[*NOTE:*] Please refer to the Figures 3.8-3.12 for the specific geometrical terms.

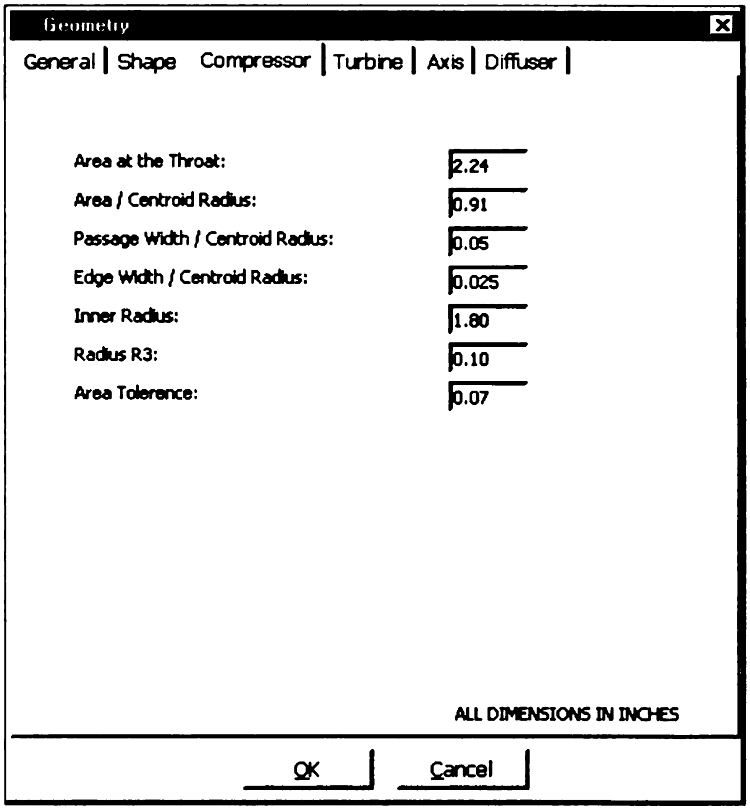


Figure 4.4: Form Geometry, Tab Compressor

Area at the Throat: Defines the critical area at the throat. That is the cross-section at 360°.

Area / Centroid Radius: Defines the ratio of the area at the throat to the centroid radius. For calculations, the centroid radius is obtained knowing this ratio and the critical area at the throat.

$$\textit{Centroid Radius} = \textit{Area at the throat} \times \textit{Area / Centroid Radius} \tag{4.1}$$

Passage Width / Centroid Radius: Defines the ratio of the passage width to the centroid radius. For calculations, the passage width is obtained as:

$$Passage\ Width = \frac{Passage\ Width}{Centroid\ Radius} \times Centroid\ Radius \tag{4.2}$$

Edge Width / Centroid Radius: Defines the ratio of the edge width to the centroid radius. For calculations the edge width is obtained as:

$$Edge\ Width = \frac{Edge\ Width}{Centroid\ Radius} \times Centroid\ Radius \tag{4.3}$$

Inner Radius: Defines the radius of the inner surface (the volute surface closest to the centerline). This radius is used only for the geometry type of *Constant Inner Diameter*.

Radius R3: Defines the corner radius of the volute.

Area Tolerance: Area tolerance is the tolerance for the area calculated by the program to the area specified.

4.2.2.4 Turbine

This sub form is used for inputting the geometrical data for the design of a turbine.

[*NOTE:*] Please refer to the figure 3.13 and figure 3.14 for the specific geometrical terms.

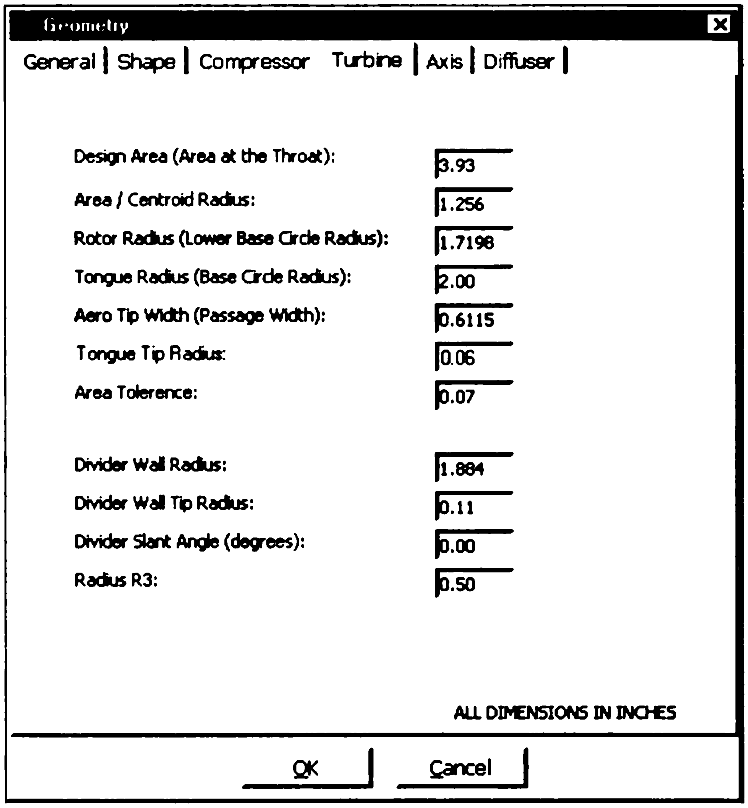


Figure 4.5: Form Geometry, Tab Turbine

Design Area (Area at the Throat): Defines the critical area at the throat. That is at the cross-section at 360°.

Area / Centroid Radius: Defines the ratio of the area at the throat to the centroid radius. For calculations, the centroid radius is obtained knowing this ratio and the critical area at the throat.

$$Centroid\ Radius = Area\ at\ the\ throat \times Area / Centroid\ Radius \tag{4.4}$$

Rotor Radius (Lower Base Circle Radius): Defines the radius of the rotor wheel. It is also known as the lower base circle radius.

Tongue Radius (Base Circle Radius): Defines the base circle radius. This is the radius of the location of the top surface of the tongue. It is defined as:

$$Base\ Circle\ Radius = Togue\ RR \times Rotor\ Radius + 2 \times Tongue\ Tip\ Radius \tag{4.5}$$

Aero Tip Width (Passage Width): Defines the aero tip passage also known as the passage width.

Tongue Tip Radius: Defines the radius of the tongue tip.

Area Tolerance: Area tolerance is the tolerance for the area calculated by the program to the area specified.

Divider Wall Radius: This is the radius of the bottom tip of the divider wall. It is also calculated as

$$Divider\ Wall\ Radius = Divider\ Wall\ RR \times Rotor\ Radius \tag{4.6}$$

(Specified only for twin flow turbines)

Divider Wall Tip Radius: Defines the radius of the divider wall tip. It is the equivalent of the divider wall thickness at its bottom edge (if the *Divider Wall Slant Angle* is not equal to zero) or the divider wall thickness, if the slant angle is zero. (Specified only for twin flow turbines)

Divider Wall Slant Angle: Defines the angle to which the walls of the divider wall are tilted from the normal. It is measured in degrees. (Specified only for twin flow turbines)

Radius R3: Defines the corner radius. (Specified only for twin flow turbines)

4.2.2.5 Axis

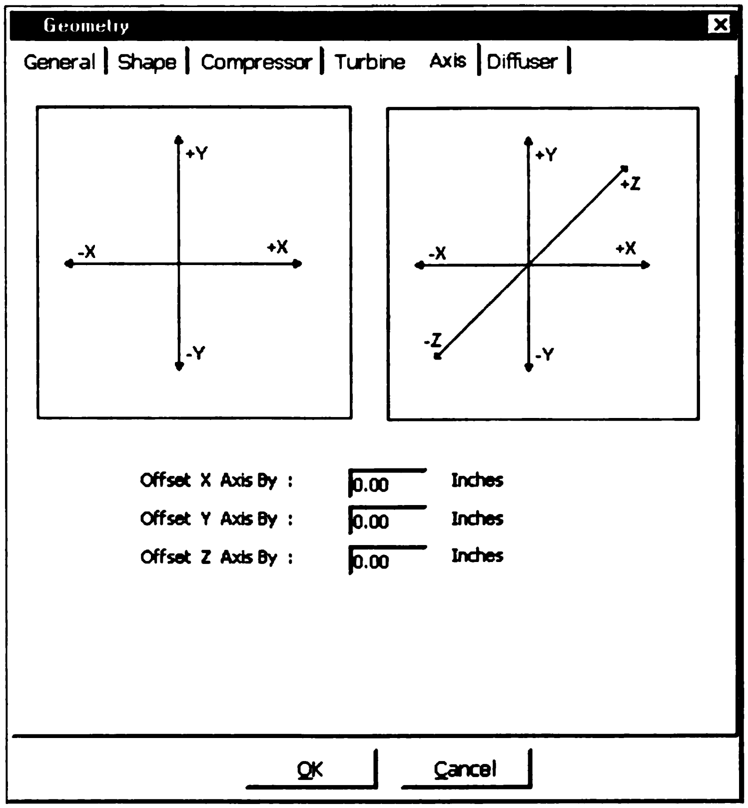


Figure 4.6: Form Geometry, Tab Axis

This sub form is used for the shifting of the axis. The volute is defined with the center of the coordinate system at the center of the volute. If required, the volute could be mated with other components in the overall design layout by moving the center of the volute (or shifting axis) to the desired location. The axis are offset individually i.e. the X-axis, Y-axis and Z-axis.

[*NOTE:*] This is applicable only for designs using the rectangular coordinate system.

4.2.2.6 Diffuser

This sub form completely defines the diffuser. There are three types of diffusers: one for the compressors (refer section 3.4 A), one for the open flow turbine (refer section 3.4 B) and the split diffuser for the twin flow turbine (refer section 3.4 C). There are two ways of designing diffusers. These are as described below.

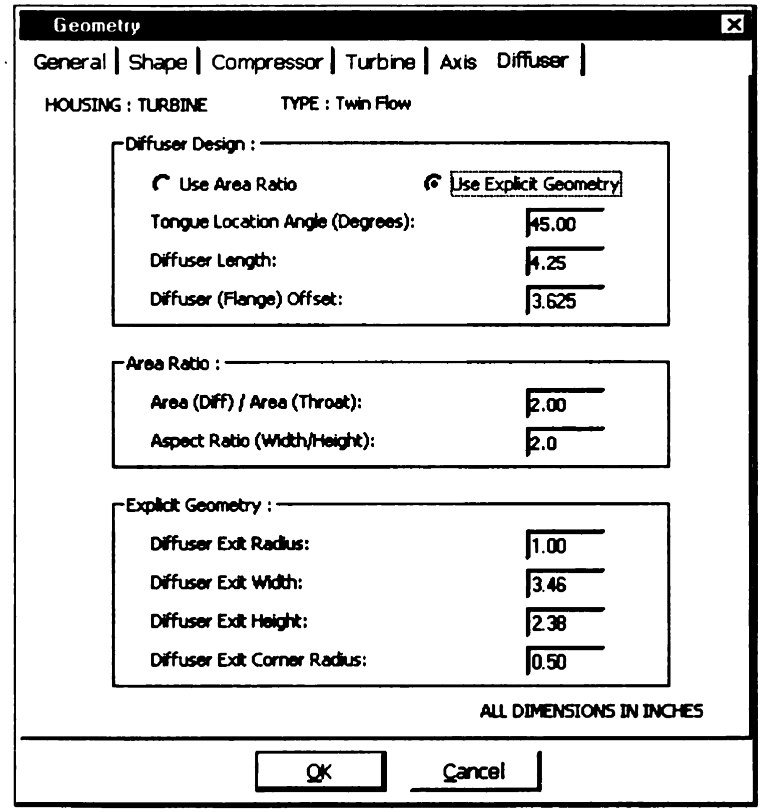


Figure 4.7: Form Geometry, Tab Diffuser

Diffuser Design: One way to design the diffuser is to specify an area ratio in which the area of the diffuser is specified as a ratio of the area at the critical area (i.e. at the cross-section 360^{o}) to the area of the diffuser or by explicitly dimensioning the diffuser. The option buttons *Use Area Ratio* calculates the dimensions from the specified area ratio, whereas the option button *Use Explicit Geometry* uses the specified dimensions by the user to calculate the diffuser geometry. If the option is *Use Area Ratio* then *Use Explicit Geometry* is disabled and similarly vice-versa.

Diffuser Length: Defines the length of the diffuser.

Diffuser (Flange) Offset: Defines the distance of the center of the flange of the diffuser from the Y-axis.

Area (Diff) / Area (Throat): Defines the ratio of the area of the diffuser to the area at the throat (i.e. the area at cross-section 360°)

Aspect Ratio: Defines the aspect ratio i.e. the ratio of the diffuser width to the diffuser height. (Specified only for turbines)

Diffuser Exit Radius: It is the radius of the diffuser exit. (Specified only for compressors)

Diffuser Exit Width: Width of the diffuser inlet. (Specified only for turbines)

Diffuser Exit Height: Height of the diffuser inlet. (Specified only for turbines)

Diffuser Exit Corner Radius: Defines the corner radius of the diffuser inlet. (Specified only for turbines)

There are two command buttons on the form *Geometry*.

OK: This command accepts all the specified data, processes it and generates the volute.

Cancel: Cancel aborts the processing and returns to the currently active screen.

[*NOTE:*] The term 'Diffuser' used above is used to indicate the diffusing element for the compressor volute as well as the inlet section for turbines. It should not be confused with the diffuser of the turbo-machine.

4.2.3 Form ADD NODE

This form adds a node or control point to the current volute profile. Since one cross-section is made up of multiple Bezier segments, it provides the user with the option of the segment to which a node should be added. It provides two pictures. The one on the left is of the current design screen and the second on the right is a picture of the default segment breakup with different coloration for ease of the user. Only one node can be added at a time. Adding a node does not change the shape of the curve; it only provides the user with an increased degree of freedom.

[*NOTE:*] This desired node point is simultaneously added in the same location across all cross sections.

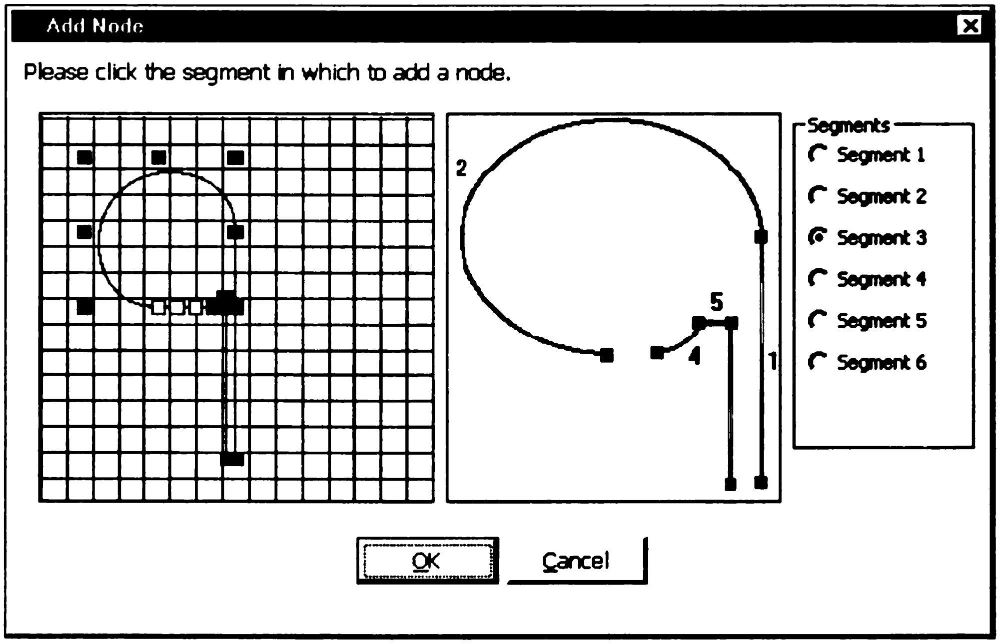


Figure 4.8: Form Add Node

Segments: This gives the user the selection of the particular segment in which the node should be added

OK: Takes the user inputs and adds a node to the desired segment

Cancel: Aborts adding of node and returns to currently active screen.

4.2.4 Form NEW DESIGN

The form *New Design* is for initiating a new design project. It obtains information from the user regarding the name of the project and the path location of the project files.

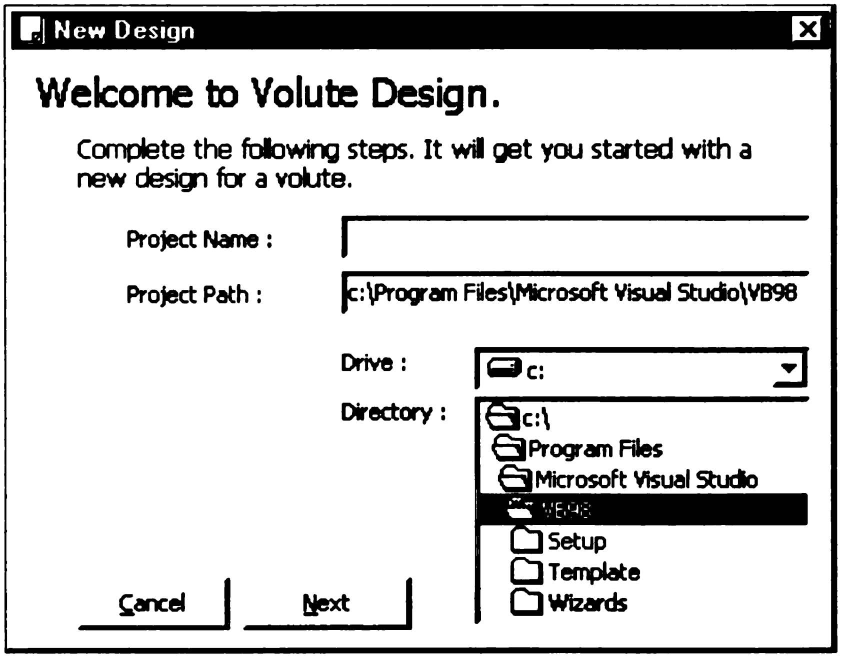


Figure 4.9: Form New Design

Project Name: Name of the project

Project Path: Only a display box, displaying the project file path

Drive: Selection of drive in which the project files should reside

Directory: Selection of directory in which the project files should reside

Next: Is highlighted only after a valid project name has been entered. Displays the *Geometry* form for the inputting of the geometrical data regarding the new design.

Cancel: Cancels the initialization of a new project

4.2.5 Form OPEN DESIGN

The form *Open Design* is for opening an existing design project. It obtains information from the user regarding the location and name of the project.

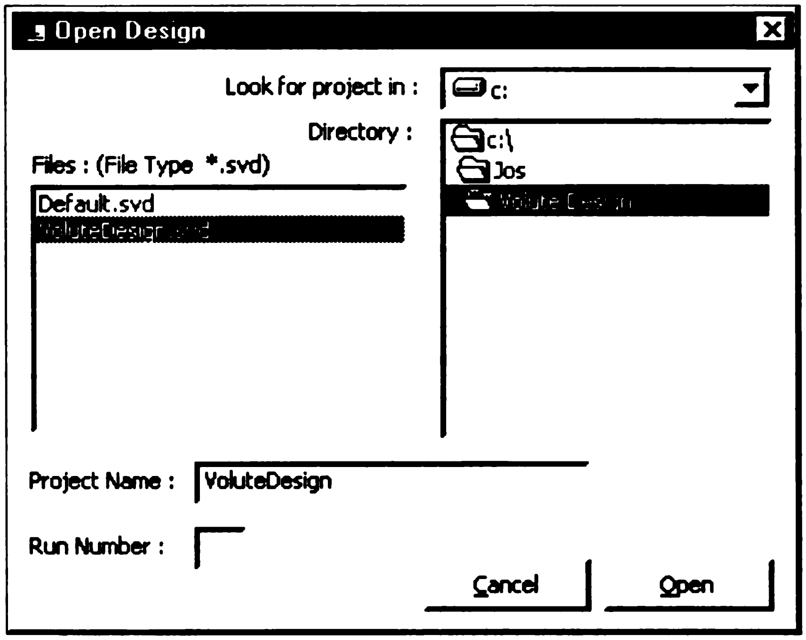


Figure 4.10: Form Open Design

Look for Project in: Selection of the drive in which the project resides.

Directory: Selection of the directory in which the project resides.

Files: Displays a list of all the project files in the selected directory.

Project Name: Display box only. Displays the project name of the project file selected form the *Files* box.

Run Number: Display box only. Displays the run number of the selected project.

Open: Opens the selected project.

Cancel: Aborts opening of a project and returns to currently active screen.

4.2.5 Form THREE DIMENSION

The form *Three Dimension*, is one of two forms for the viewing a 3-D model of the volute. The first form obtains user input regarding the size of the grid, while the second form displays the volute model.

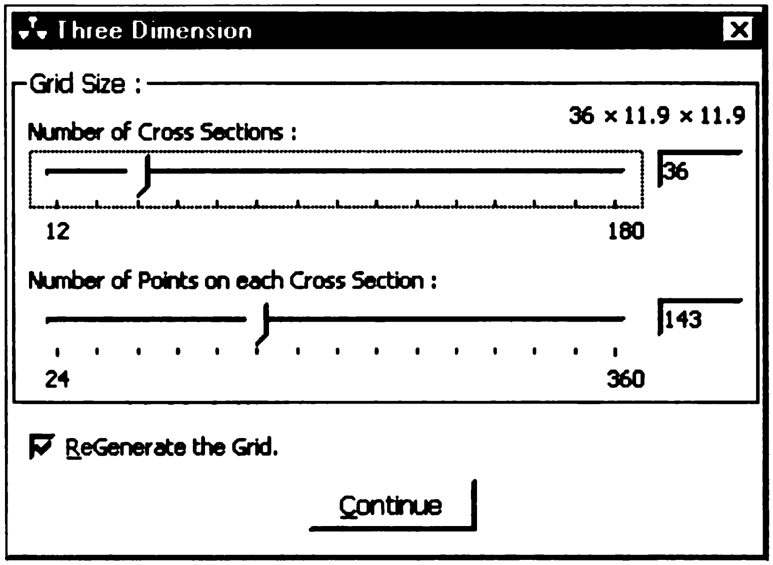


Figure 4.11: Form Three Dimension (Splash)

Number of Cross Sections: This defines the number of cross sections desired for the volute. There are two methods of inputting this data. One, the slider bar could be moved to the desired number of cross sections, which is constantly updated in the text box to the right. Alternatively, the desired number of cross sections could be entered directly into the text box. The limiting values are displayed below the slider bar.

Number of Points on each Cross Section: This is the total number of points that make up a single cross section. The methods of inputting the data are similar to the *Number of Cross Sections*.

ReGenerate the Grid: Checking this box forces the program to regenerate the grid each time the 3-D model is desired to be viewed. If this is not checked, the last model in the program's memory would be displayed.

Continue: Brings up the *Volute Visualization – [Three Dimension]* form that displays the 3-D model.

The form *Volute Visualization – [Three Dimension]* displays the 3-D model and has the capabilities apart from viewing the model, to zoom in or out, to view different parts of the volute, to view the volute at different projections and to rotate the model.

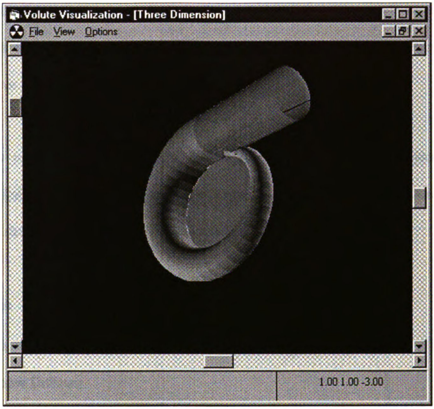


Figure 4.12: Form Three Dimension

4.2.5.1 Functions

1) Left Scroll Bar: This is the zoom bar. Moving the pointer up zooms in, while moving it down zooms out.

2) Right Scroll Bar: The right scroll bar is for rotating the model about the X-axis.

3) Bottom Scroll Bar: The bottom is scroll bar is used for rotating the model about the Y-axis.

4.2.5.2 Menu:

1) ***File***

K File

- Print
- Close

Print: Prints the form

Close: Closes the *Volute Visualization – [Three Dimension]* form and returns to the currently active window

2) ***View***

The view menu is for viewing the model at different angles of projection. The different available angles of projection are *Front*, *Back*, *Side*, *Diffuser Down*, *Offset Front*, *Offset Back* and *Offset Side*.

3) ***Options***

K Options

- Show Volute Core
- Show Diffuser
- Wiremesh Volute Core
- Wiremesh Diffuser

Show Volute Core: Toggle button that displays or removes the volute core section

Show Diffuser: Toggle button that displays or removes the diffuser section

Wiremesh Volute Core: Toggle button that displays the volute core as a wiremesh model or as a solid model

Wiremesh Diffuser: Toggle button that displays the diffuser section as a wiremesh model or as a solid model

4.2.6 Form GENERATE FILES

The form *Generate Files* is for generating the output coordinate files. The user specifies the grid size of the volute model, and the file type desired. The program assigns the file name and generates the file.

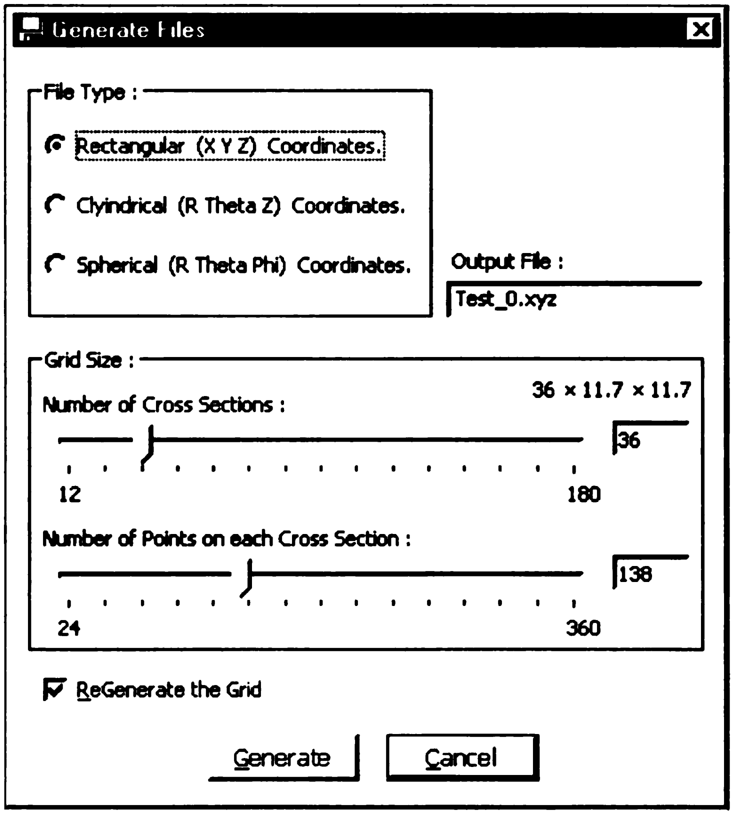


Figure 4.13: Form Generate Files

File Type: This is for the coordinate system of the output-generated file. The user specifies one of the three choices of coordinates namely, rectangular, cylindrical or spherical coordinates.

Output File: Only a view box. Gives the name of the output file. The output file is saved in the same directory as the project.

Number of Cross Sections: This is the number of cross sections desired for the volute. There are two methods of inputting this data. One, the slider bar could be moved to the desired number of cross sections, which is constantly updated in the text box to the right. Alternatively, the desired number of cross sections could be entered directly into the text box. The limiting values are displayed below the slider bar.

Number of Points on each Cross Section: This is the total number of points that make up a single cross section. The methods of inputting the data are similar to the ***Number of*** *Cross Sections*.

ReGenerate the Grid: Checking this box forces the program to regenerate the grid each time the 3-D model is desired to be viewed. Otherwise, the last model in the programs memory would be displayed.

Generate: Generates the output file.

Cancel: Aborts the operation to generate a file and returns to the currently active screen.

4.2.7 Form QUICK VIEW

This form is a text viewer and allows the user to view the geometry file or the coordinate files generated.

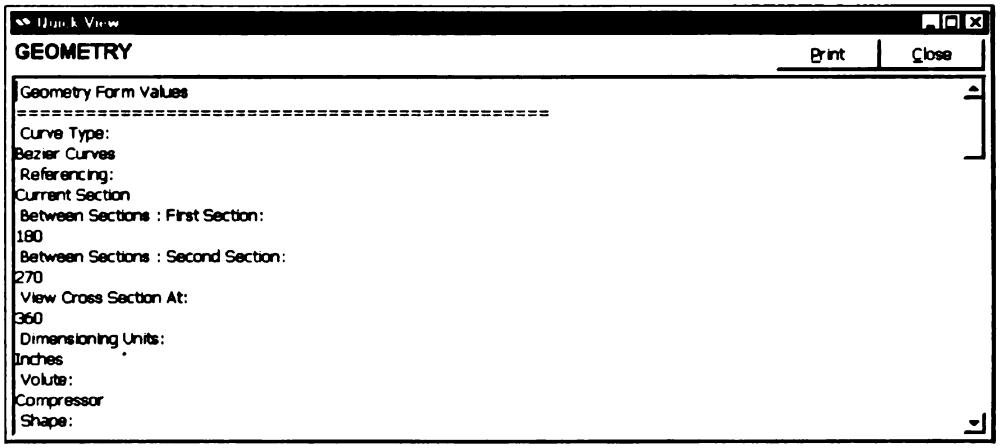


Figure 4.14: Form Quick View

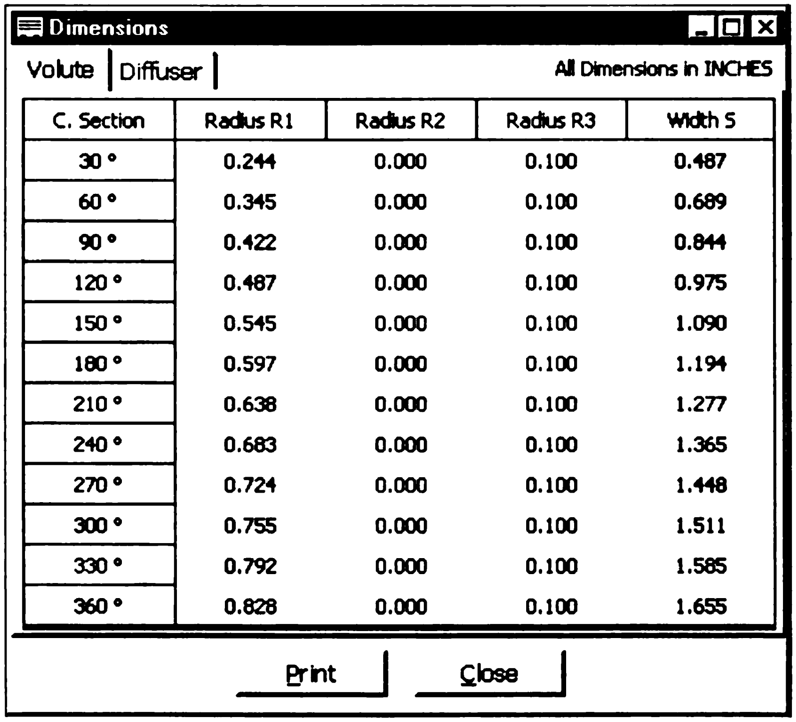
Title: Name of the file being displayed

Print: Prints the form

Close: Closes the form and returns to last currently active screen

4.2.8 Form DIMENSIONS

This form displays a tabulated list of the geometrical data (radii R_1, R_2, R_3 and width S) necessary for to specify the volute. It displays the list at of radii and width at every 30° cross-section interval. These dimensions are useful for the drafting of the volute.


Dimensions

Volute | Diffuser

All Dimensions in INCHES

C. Section	Radius R1	Radius R2	Radius R3	Width S
30 °	0.244	0.000	0.100	0.487
60 °	0.345	0.000	0.100	0.689
90 °	0.422	0.000	0.100	0.844
120 °	0.487	0.000	0.100	0.975
150 °	0.545	0.000	0.100	1.090
180 °	0.597	0.000	0.100	1.194
210 °	0.638	0.000	0.100	1.277
240 °	0.683	0.000	0.100	1.365
270 °	0.724	0.000	0.100	1.448
300 °	0.755	0.000	0.100	1.511
330 °	0.792	0.000	0.100	1.585
360 °	0.828	0.000	0.100	1.655

Print | Close

Figure 4.15: Form Dimension

Print: Prints the form

Close: Closes the form and returns to currently active screen

4.2.9 Form GRAPHS

This is an 'output' form. It displays the two graphs of the area distribution that are necessary to design the volute. The first graph (above) is the graph of the area against the cross section. The second graph (below) is the graph of the ratio of the area by centroid radius against the cross-section angle.

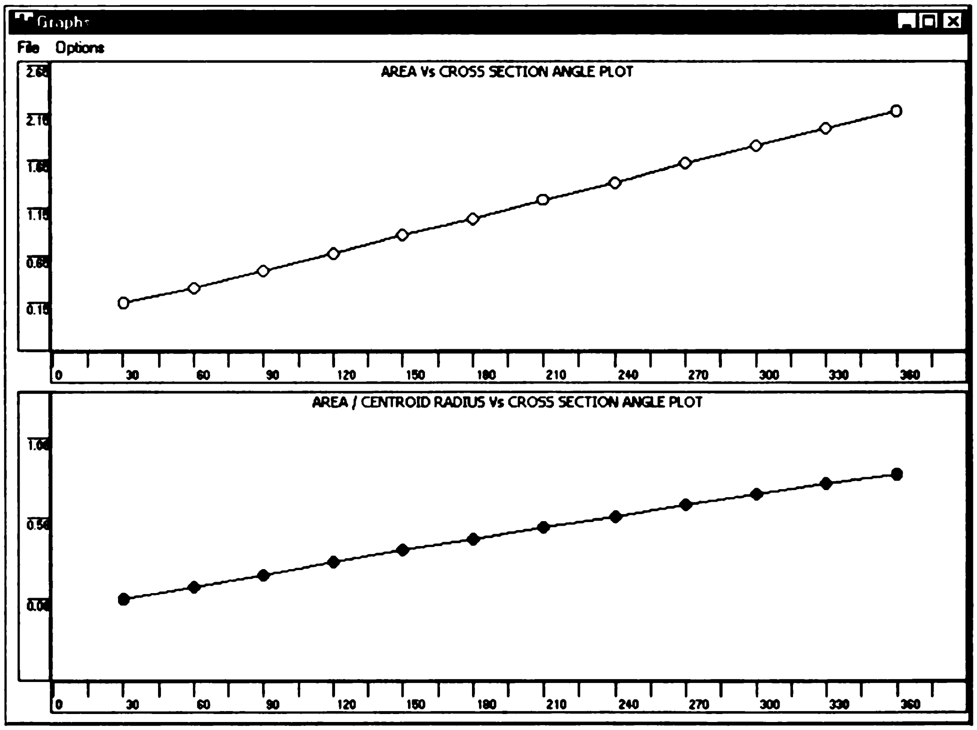


Figure 4.16: Form Graphs

A) Menu:

1) ***File***

K File

- Print
- Close

Print: Prints the form

Close: Closes the form and returns to the currently active screen

2) ***Options***

K Options

- Grid

Grid: Toggle switch that displays the grid or removes the grid

4.3 Sample Design

We shall now model two sample volutes. One for a compressor and the other for a turbine. Their geometry, 2-D cross-section at 360°, their 3-D views (solid) and their area graphs are given below. The geometry is the input data for the *Geometry* form.

4.3.1 Compressor

1) Geometry:

Curve Type:	Bezier Curves
Referencing:	Current Section
View Cross Section At:	360
Dimensioning Units:	Inches
Volute:	Compressor
Shape:	Circular
Compressor Geometry:	Constant Centroid
X offset:	0.00
Y offset:	0.00
Z offset:	0.00
Area at the Throat:	2.24
Area/Centroid Radius:	0.91

Passage Width/Centroid Radius:	0.05
Edge Width/Centroid Radius:	0.025
Inner Radius:	1.80
Radius R3:	0.10
Area Tolerance:	0.07
Diffuser Design Type:	Use Explicit Geometry
Diffuser Exit Radius:	1.25
Diffuser Length:	4.25
Diffuser (Flange) Offset:	3.625

2) 2-D Cross Section:

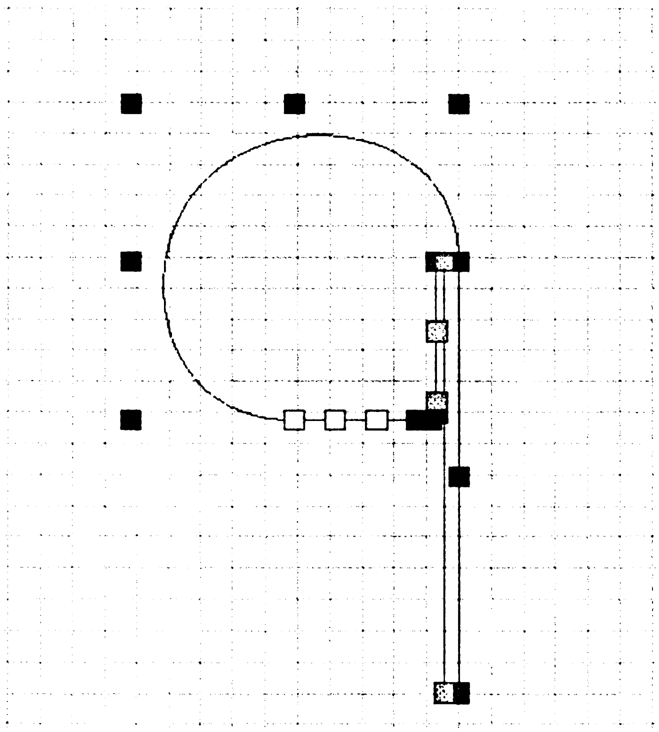


Figure 4.17: 2-D Cross section at 360° for sample compressor design

3) Solid Model:

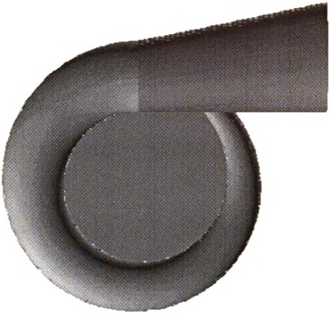


Figure 4.18: Solid model for sample compressor design

4) Area Graphs:

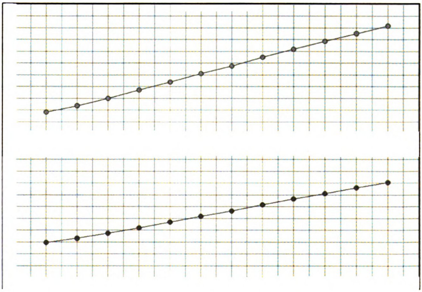


Figure 4.19: Area graphs for sample compressor design

4.3.2 Turbine

1) Geometry:

Curve Type:	Bezier Curves
Referencing:	All Section
View Cross Section At:	90
Dimensioning Units:	Inches
Volute:	Turbine
Turbine Geometry:	Twin Flow
X offset:	0.00
Y offset:	0.00
Z offset:	0.00
Area at the Throat:	3.93
Area/Centroid Radius:	1.256
Base Circle Radius:	2.0977
Passage Width:	0.6115
Area Tolerance:	0.01
Lower Base Circle Radius:	1.7198
Radius R3:	0.50
Divider Wall Tip Radius:	0.11
Divider Wall Slant Angle:	0.00
Divider Wall Radius:	1.884
Tongue Tip Radius:	0.06
Diffuser Design Type:	Use Explicit Geometry

Diffuser Length:	4.25
Diffuser Exit Width:	3.46
Diffuser Exit Height:	2.38
Diffuser Exit Corner Radius:	0.62
Diffuser (Flange) Offset:	3.625

2) 2-D Cross Section:

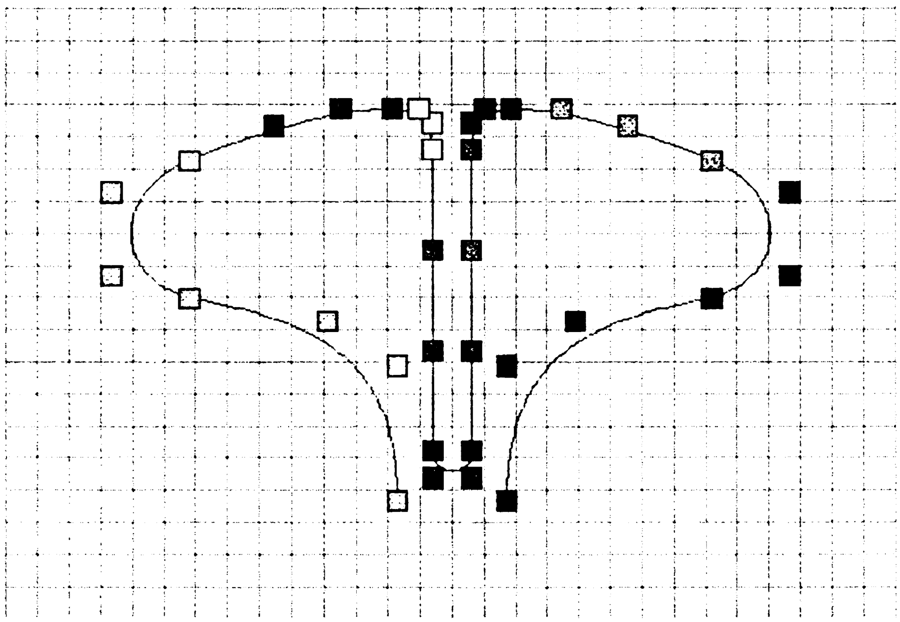


Figure 4.20: 2-D Cross section at 360^{o} section for sample turbine design

3) Solid Model:

Figure 4.21: Solid model for sample turbine design

4) Area Graphs:

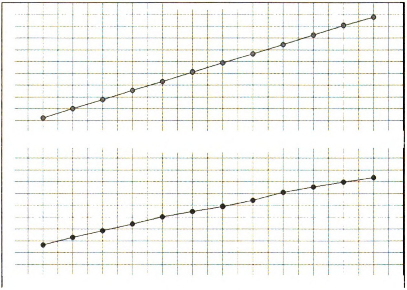


Figure 4.22: Area graphs for sample turbine design

Chapter 5

GEOMETRY VERIFICATION

5.1 Introduction

This chapter, geometry verification of the volute, is a check of the Visual Basic program that has been developed to show that the program generates the volutes that conform to existing standards. The procedure to verify the volutes compares a volute generated by the Visual Basic program to the same volute that has been generated by another software that could check the necessary dimensions. This calibration is important, as the verification should be made outside of the Visual Basic program. This is due the fact that any errors generated by the Visual Basic program are not compounded during the verification.

The verification is done using the software package FIELDVIEW. FIELDVIEW is an interactive data visualization package that assists in the investigation of complex three-dimensional fluid dynamic data sets. FIELDVIEW is a leading visualization package designed specifically for CFD data. Unlike most plotting applications, FIELDVIEW is built upon a full 3D data handling capability, with comprehensive support for both structured and unstructured grids.

The critical design criteria, namely the area and the centroid radius, are measured at four cross section locations: 360°, 270°, 180° and 90°. Measurements of area and centroid radius at these locations are made in FIELDVIEW and then compared with the results obtained from the visual basic program developed. The comparisons are made in a tabular format that displays the percentage errors alongside.

5.2 Geometry Verification

The verifications have been performed for compressors as well as turbines, as listed below:

(1) Compressor Circular Constant Inner Diameter:

Table 5.1: Verification of Compressor Circular Const. ID

DESIGN INFORMATION	
Volute	Compressor
Type	Constant Inner Diameter
Shape	Circular
Area at Throat	1.63
Area Throat / Centroid Radius	0.6
Passage Width / Centroid Radius	0.05
Edge Width / Centroid Radius	0.025
Inner Radius	1.8
R3	0.1
Area Tolerance	0.01

RESULTS			
Design Criteria	**Model** (Inches)	**VB Model** (Inches)	**% Error**
Area - 360°	1.6297	1.6203	0.58%
Area - 270°	1.2181	1.2103	0.64%
Area - 180°	0.8037	0.798	0.71%
Area - 90°	0.3971	0.3884	2.19%
Radius - 360°	2.4558	2.4601	-0.18%
Radius - 270°	2.3681	2.3722	-0.17%
Radius - 180°	2.2632	2.267	-0.17%
Radius - 90°	2.129	2.1368	-0.37%

(2) Compressor Circular Constant Centroid

Table 5.2: Verification of Compressor Circular Const. Centroid

DESIGN INFORMATION			
Volute	Compressor		
Type	Constant Centroid		
Shape	Circular		
Area at Throat	1.63		
Area Throat / Centroid Radius	0.6		
Passage Width / Centroid Radius	0.05		
Edge Width / Centroid Radius	0.025		
Inner Radius	1.8		
R3	0.1		
Area Tolerance	0.01		
RESULTS			
Design Criteria	**Model** (Inches)	**VB Model** (Inches)	**% Error**
Area - 360°	1.6208	1.6231	-0.14%
Area - 270°	1.2047	1.206	-0.11%
Area - 180°	0.7878	0.788	-0.03%
Area - 90°	0.3729	0.3718	0.29%
Radius - 360°	2.6224	2.6226	-0.01%
Radius - 270°	2.6382	2.6384	-0.01%
Radius - 180°	2.6572	2.6568	0.02%
Radius - 90°	2.683	2.6825	0.02%

(3) Compressor Elliptical Constant Inner Diameter

Table 5.3: Verification of Compressor Elliptical Const. ID

DESIGN INFORMATION	
Volute	Compressor
Type	Constant Inner Diameter
Shape	Elliptical
Area at Throat	1.63
Area Throat / Centroid Radius	0.6
Passage Width / Centroid Radius	0.05
Edge Width / Centroid Radius	0.025
Inner Radius	1.8
R3	0.1
Area Tolerance	0.01

RESULTS			
Design Criteria	**Model** (Inches)	**VB Model** (Inches)	**% Error**
Area - 360°	1.6359	1.6226	0.81%
Area - 270°	1.2228	1.2115	0.92%
Area - 180°	0.8083	0.7998	1.05%
Area - 90°	0.3949	0.3897	1.32%
Radius - 360°	2.6362	2.6437	-0.28%
Radius - 270°	2.5244	2.5316	-0.29%
Radius - 180°	2.3912	2.3982	-0.29%
Radius - 90°	2.2178	2.2223	-0.20%

(4) Compressor Elliptical Constant Centroid

Table 5.4: Verification of Compressor Elliptical Const. Centroid

DESIGN INFORMATION	
Volute	Compressor
Type	Constant Centroid
Shape	Elliptical
Area at Throat	1.63
Area Throat / Centroid Radius	0.6
Passage Width / Centroid Radius	0.05
Edge Width / Centroid Radius	0.025
Inner Radius	1.8
R3	0.1
Area Tolerance	0.01

RESULTS			
Design Criteria	**Model** (Inches)	**VB Model** (Inches)	**% Error**
Area - 360°	1.6205	1.6228	-0.14%
Area - 270°	1.2047	1.2059	-0.10%
Area - 180°	0.7891	0.7889	0.03%
Area - 90°	0.3737	0.372	0.45%
Radius - 360°	2.6001	2.6002	0.00%
Radius - 270°	2.6203	2.6204	0.00%
Radius - 180°	2.6447	2.6446	0.00%
Radius - 90°	2.678	2.6782	-0.01%

(5) Open Flow Turbine

Table 5.5: Verification of Open Flow Turbine

DESIGN INFORMATION	
Volute	Turbine
Type	Open Flow
Area at Throat	3.93
A/R	1.256
Rotor Radius (LBCR)	1.7198
Tongue Radius (BCR)	2.0977
Aero Width (Passage Width)	0.6115
Tongue Tip Radius	0.06
Area Tolerance	0.01
Diffuser Exit Width	3.46
Diffuser Exit Height	2.36
Diffuser Corner Radius	0.62
Diffuser Length	4.25
Flange Offset	3.625

RESULTS			
Design Criteria	**Model** (Inches)	**VB Model** (Inches)	**% Error**
Area - 360°	3.941	3.9411	0.00%
Area - 270°	2.9654	2.9655	0.00%
Area - 180°	1.9706	1.9705	0.01%
Area - 90°	0.9905	0.9906	-0.01%
Radius - 360°	3.33	3.3302	-0.01%
Radius - 270°	3.061	3.0616	-0.02%
Radius - 180°	2.772	2.772	0.00%
Radius - 90°	2.398	2.398	0.00%

(6) Twin Flow Turbine

Table 5.6: Verification of Twin Flow Turbine

DESIGN INFORMATION	
Volute	Turbine
Type	Twin Flow
Area at Throat	3.93
A/R	1.256
Rotor Radius (LBCR)	1.7198
Tongue Radius (BCR)	2.0977
Aero Width (Passage Width)	0.6115
Tongue Tip Radius	0.06
Area Tolerance	0.01
Diffuser Exit Width	3.46
Diffuser Exit Height	2.36
Diffuser Corner Radius	0.62
Diffuser Length	4.25
Flange Offset	3.625
Divider Wall Radius	1.884
Divider Wall Tip Radius	0.11
Slant Angle	0
Radius R3	0.5

Table 5.6 Continued...

RESULTS					
Design Criteria	**Model**		**VB Model**		**% Error**
	(Radius Ratio)	(Inches)	(Radius Ratio)	(Inches)	
Inlet Area		7.0319		7.0513	-0.28%
Inlet Length		4.25		4.25	0.00%
Inlet Offset		3.625		3.625	0.00%
Critical Area(Area Throat)		3.9251		3.93	-0.12%
Critical Radius		3.0948		3.129	-1.09%
A/R		1.2683		1.256	0.98%
Wheel O/D		3.4396		3.4396	0.00%
Tip Width		0.6115		0.6115	0.00%
Divider Wall Radius Ratio	1.0953	1.8837	1.0955	1.884	-0.02%
Min. Divider Wall Thick		0.22		0.22	0.00%
Tongue Radius Ratio (Upper)	1.2197	2.0977	1.2198	2.0978	0.00%
Tongue Radius Ratio (Lower)	1.1507	1.979	1.15	1.9778	0.06%

5.3 Discussions:

We see from the above volute models that the critical dimensions conform to the existing model to within a maximum error of 1%. The errors mainly exist in the area calculations, which in turn has a direct affect on the centroid radius. Some of the reasons for the errors in the area calculations are:

(1) Bezier Curves:

As seen from the previous chapters, the volutes are generated using Bezier curves. The Bezier polynomial definition of a circle, however, does not generate an exact circle. It conforms to the circular arc with a maximum deviation of approximately $\pm 0.05\%$ from the mean radius.

(2) Tolerance:

The program calculates the area to within the specified tolerance limit. There is a minimum tolerance limit set, because the program uses an iterative process in calculating the area; That is, it increases the dimensions (radii and width) incrementally until the specified area has been reached within the tolerance limit. This minimum tolerance is required (and set to 0.001) because program would oscillate about the specified area, taking an extended time to converge.

(3) Numerical Integration:

The program calculates the area by numerical integration, following the Trapezoidal rule. Even though a large number of points (typically 500) are taken on each cross-section, a small error arises in the area calculation.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

We have seen from Chapter 1, "Introduction", the need for having a good design tool for volutes. This design tool should be flexible to the different shapes of volutes, should be user friendly in the exchange of data, should be flexible to manipulate the cross-sectional shapes and should be robust. Until recently, volute design has been the neglected part of Turbochargers, with the emphasis on the rotor design. Most companies designed their new volutes by scaling their successful predecessor, and most of the time this scaling design method works. However, being able to design a turbocharger with good performance characteristics, would only be realized if an effective volute is designed.

To be able to design the volute, a good design tool is required. This design tool would generate a model, test the flow paths and manipulate the cross section shape to achieve a good design. The starting point of this design tool is to generate the grid of the volute model. Once the grid is obtained, an engineer can perform an analysis of the volute, manipulating the geometry as and when required.

The Visual Basic Program – Volute Design, as described about in Chapter 4, "Volute Design – The Program", is a good first step in this direction. It provides us with a platform that could be expanded in the future to accommodate other features useful to the design process. Visual Basic has an excellent development environment that one uses to develop user friendly design tools. Bezier curves give the required flexibility to manipulate the volute cross section shapes. They are easy to implement especially by visually implementing the control points in which one is able to move the control points about, thereby changing the shape of the curve. We have also seen the various features of the program. Not only is the program able to generate the volute based on certain design specifications, but it is also able to change the geometry cross sections. The program is also flexible in the size of the grid generated and also the file coordinate system of the output files.

Thus, the design tool *Volute Design* developed in this thesis would lead us in the direction of designing and optimizing a volute with a favorable shape and flow paths.

6.2 Future Work

The work for the future has been divided into two groups. Improvements to the program and the work related to analyzing the volute with a CFD package are as follows.

6.2.1 The Program

(1) Tongue Definition:

The tongue is currently generated for turbines by the interpolation of the two cross sections between 0° and 30°. By proper definition of the radius of curvature of the tongue, it could be generated more accurately and we would have more control over this region of the volute.

(2) Tongue Location Angle:

The tongue location angle is presently set at the cross section at 360°. If we are able to move the location of the tongue across the volute, it would give us additional flexibility in designing different types of volutes for the turbine.

(3) Direct Specification of the Geometry.

If the program could be modified to directly input the critical dimensions ($R1$, $R2$, $R3$ and S) from the user, it would allow the designer by overriding the programs default method of calculating these dimensions, the ability to create a specific volute, thus giving greater flexibility. At present the program calculates theses values from the area and centroid radius specified at the throat [1].

(4) Modifying the A/R Ratio.

Future work could include being able to directly modify the A/R ratio graph that is plotted by the program. Changes in the A/R ratio would then be directly reflected in the cross section area. This gives the designer additional flexibility in the methods of design.

(5) Diffuser Definition:

The diffusers are currently generated by direct interpolation of the end cross sections. If we specify the geometry at additional intermediate cross-sections, it would give the designer better control over the design of the diffusers and thus the volute.

(6) Tongue Tip.

Generating the tongue tip and by proper control of its angle in the volute is of significance importance in directing the flow. Proper control of the angle, radius and length of the tongue tip would improve overall performance of the volute.

(7) Bezier Surface Patches:

If we implement the use of Bezier surface patches (these are two-dimensional Bezier curves) it would result in a smoother grid.

(8) Rectangular Cross section for Compressors:

Generating rectangular cross sections allows for the testing and performance of rectangular cross sectioned volutes, giving the designer a different type of volute to experiment with.

(9) Viewing any Cross Section:

At present, the program displays cross sections only at 30^{o} sections. However, it may be desired to view a section that is not at a 30^{o} interval. This could be implemented by displaying the front view of the volute and allowing the user to pick the cross section they would wish to view.

(10) DLL/OLE Links:

DLL (Dynamic LinkLibrary) and OLE (Object Linking and Embedding) give the program addition flexibility with Visual Basic to use shared code or directly link them to other programs. These other software could be a drafting tool, a CFD analysis package etc.

(11) Interface with TASCFlow:

TASCFlow is a software that would be used to run a computational fluid dynamic analysis on the turbine. All the input requirements in TASCFlow such as pressure, temperature, mass flow rate, etc. could be specified in the program and used when the analysis is performed directly from the program.

6.2.2 TASCFLOW

The flow inside of centrifugal compressors and turbines is highly three dimensional. This makes the design of volutes a difficult task. Most of the volute design methods are based on inviscid flow assumption, for example, Chapple [21], Baskharone [20], Owarish [24], Chen [22] and so on. Meanwhile, however, we still can not fully understand the flow mechanism inside volutes; therefore, we can not design new models aerodynamically. Recently, Computational Fluid Dynamics (CFD) is becoming a powerful tool in both volute design and flow analysis. It can give us the details of the flow, so we can understand the flow mechanism better. Ayder and Van den Braembussche [19] predicted the three-dimensional inviscid flow field inside a centrifugal compressor volute using an Euler solver. To take the losses into account, a loss model was incorporated. They gave a detail comparison between the numerical result and the experiment data. Martinez-Botas, Pullen and Shi [23] analyzed the three-dimensional flow through a turbine volute with non-symmetric circular cross-section by using a three-dimensional Navier-Stokes solver, and the results are satisfactory for most part of the volute and much better than the free-vortex result.

A CFD analysis for the current project is being conducted. Some of the results of a twin flow turbine casing by using a commercial code TASCFlow are shown in the following.

Figure 6.1 is the grid, which is a multi-block mesh. Using multi-block meshes allowes us to build more orthogonal grid near the walls, thereby giving us a better result near the wall.

Figure 6.2 shows the pressure profile on the wall, including the exit surface.

The velocity vectors are in Figure 6.3. It can be seen that the flow complies with the free vortex assumption from the inlet to a quarter of the volute. Thereafter, the flow becomes fully three dimensional, and the free-vortex assumption is no longer valid.

From these results, it is proved that CFD analysis is a useful method in volute design and should be implemented for the analysis of the flow path in volutes.

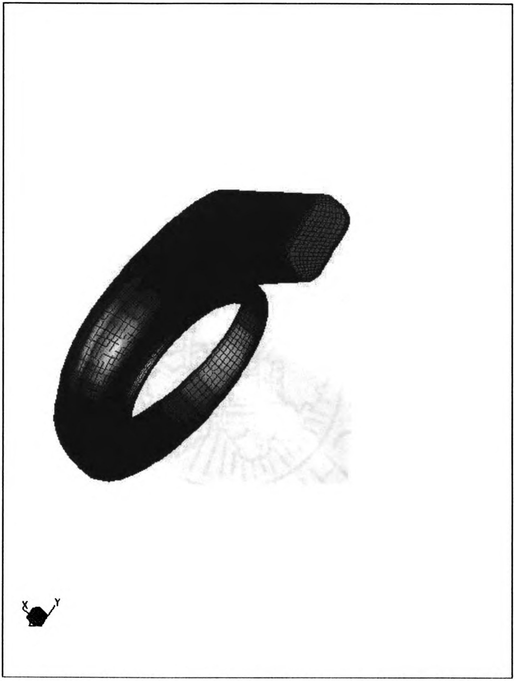


Figure 6.1: Multi-block Twin Flow Turbine Mesh

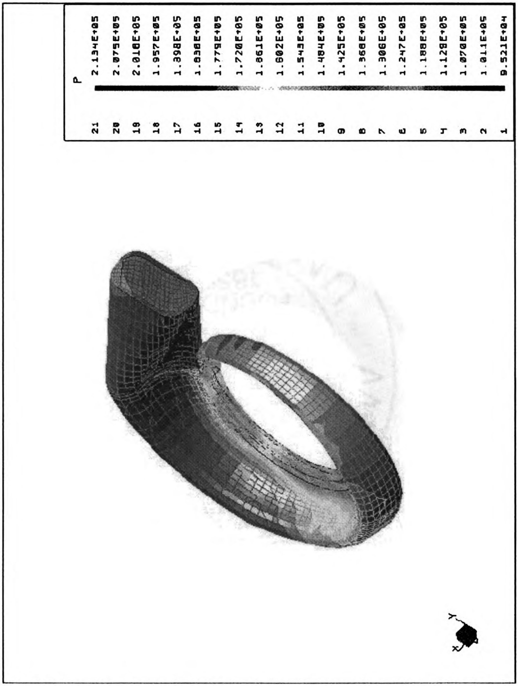


Figure 6.2: Pressure Distribution on the Walls

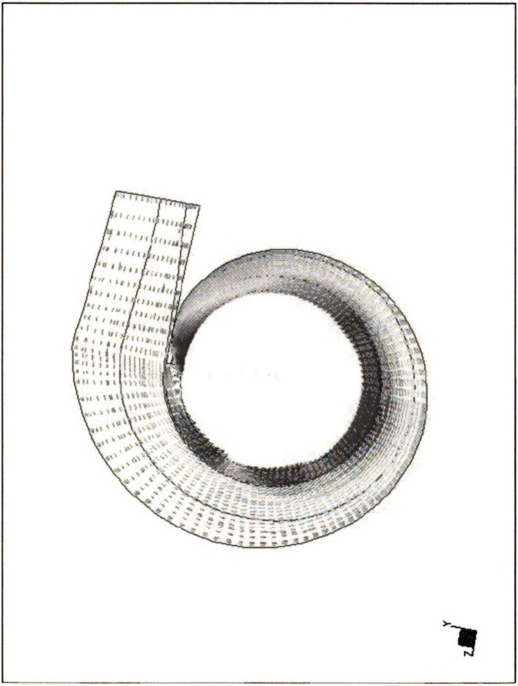


Figure 6.3: Velocity Vectors at a Surface

REFERENCES

REFERENCES

1] Dsouza, Michael, 1997, Generation of Compressor and Turbine Volute Casings Using Visual Basic 4.0, Thesis, Michigan State University.

2] Farin, Gerald, 1990, Curves and Surfaces for Computer Aided Geometric Design, Second Edition, Academic Press Inc.

3] Brown, W.B. and Bradshow, G.R., 1949, Design and performance of a family of diffusing scrolls with mixed flow impeller and vaneless diffuser, NACA Report 936.

4] Eckert, B and Schnell, E., 1980, Axial - und Radialkompressoren, Second Edition, Springer Verlag.

5] Stiefel, W., 1972, Experiences in the development of radial compressors, Lecture Notes VKI, Brussels.

6] Chapple, P.M., Flynn, P.F. and Mulloy, J.M., 1980, Aerodynamic design of fixed and variable geometry nozzleless turbine casing, ASME Journal of Engineering for Power, Vol. 102, January 1980.

7] Japikse, D., 1982, Advanced diffusion levels in turbocharger compressor and component matching, IMECHE Conference on Turbochargers and Turbocharging, Paper C45/82.

8] Weber, C.R. and Koronowski, M.E., 1986, Meanline performance prediction of volutes in centrifugal compressors, ASME-IGTI 1986, Paper 86-GT-216.

9] Iversen, H., Rolling, G.R. and Carson, J., 1960, Volute pressure distribution, radial force on the impeller and volute mixing losses of a radial flow centrifugal pump, ASME Journal of Engineering for Power, Vol. 82, April 1960.

10] Kurokawa, J., 1980, Theoretical determination of the flow characteristics in volutes, IAHR-AIRH Symposium Tokyo.

11] Badie, R., Jonker, J.B. and Van Essen T.G., Calculation on the time dependent potential flow in a centrifugal pump, ASME Paper 92-Gt-151.

12] Faux, I. D. and Pratt, M. J., Computational Geometry for Design and Manufacture, First Edition, Ellis Horwood Limited.

13] Rogers, David F. and Adams, J. Alan, Mathematical Elements for Computer Graphics, Second Edition, McGraw-Hill Publishing Company.

14] Qiulin, Ding and Davies, B. J., 1987, Surface Engineering Geometry For Computer–Aided Design and Manafacture, Ellis Horwood Limited.

15] Bu-qing, Su and Ding-yaun, Liu, 1989, Computation Geometry – Curve and Surface Modeling, Academic Press, Inc.

16] Anton, Howard, 1992, Calculus with Analytic Geometry, Fourth Edition, Wiley.

17] Thomas, George B. Jr. and Finney, Ross L., 1990, Calculus and Analytical Geometry, Seventh Edition, Addison-Wesley Publishing Company.

18] Gurewich and Gurewich, Visual Basic 4.0, Third Edition, Sam's Publishing.

19] Ayder, E. and Van den Braebussche, R., 1993, Numerical Analysis of the 3D Swirling Flow in Centrifugal Compressor Volutes, ASME 93-GT-122

20] Baskharone, E. A., 1993, Optimization of the Three Dimensional Flow Path in the Scroll-Nozzle System of a Radial Inflow Turbine, ASME 83-GT-127

21] Chapple, P. M., 1980, Aerodynamic Design of Fixed and Variable Geometry Nozzleless Turbine Casings, ASME Journal of Engineering for Power, Vol.102 p.141-147

22] Chen H, 1996, Design Method of Volute Casings for Turbocharger Turbine Applications, IMECHE, Journal of Power and Energy, Vol.210, P149-156

23] Martinez-Botas, R.F., Pullen K.R. and Shi, F., 1996, Numerical Calculations of a Turbine Volute Using A 3-D Navier-Stokes Solver, ASME 96-GT-66

24] Owarish, H. O., (1992), A Two-Dimensional Flow Analysis Model for Designing a Nozzle-less Volute casing for Radial Flow Gas Turbines, ASME, Journal of Turbomachinery, Vole 114, p420-410