This is to certify that the

thesis entitled

Generation Of Compressor And Turbine
Volute Casings Using Visual Basic 4.0

presented by

Michael S. D'Souza

has been accepted towards fulfillment
of the requirements for

Master Of Science degree in Mechanical Engineering

Major professor

Date 06-18-97

GENERATION OF COMPRESSOR AND TURBINE VOLUTE CASINGS USING VISUAL BASIC 4.0

By

Michael S. D'Souza

A THESIS

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

MASTERS OF SCIENCE

Department of Mechanical Engineering

1997

ABSTRACT

GENERATION OF COMPRESSOR AND TURBINE VOLUTE CASINGS USING VISUAL BASIC 4.0

By

Michael S. D'Souza

Regarding compressor volutes, it is strongly believed that quite a substantial additional pressure recovery and gain in efficiency can be obtained from a good volute design. For an industrial or a turbocharger compressor stage with a well-designed volute, typically, there is an efficiency loss of between 2 - 5 percent in the volute. Nearly, all arising from the inability of the volute to use the radial kinetic energy out of the diffuser and from secondary flows.

The objective of this work is to develop a design and analysis tool for compressor volutes and turbine scrolls. The work will review volute and scroll technology, study theoretical volute and scroll flow, and develop a design procedure for compressor volutes and turbine scrolls. The major portion of the work is programming visual basic where the major task is to create subroutines to generate the volute profile and cross-section details. The objective of using visual basic is to make the software a windows based application and easy to use.

This thesis is dedicated to my parents, Vincent and Shirley D'Souza.

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation to several people whose guidance and support made this work possible. I would first like to thank my advisor Dr. A. Engeda for letting me pursue a subject I was interested in. His technical guidance and support was invaluable and very much appreciated; to Dr. C. Somerton and Dr. H. Lee, the other committee members for their time and supportive comments. I would like to specially thank Mr. G. Bruce for his guidance and patience which has helped me gain a vast amount of knowledge on the subject from an industrial point of view.

I must also express my appreciation to Schwitzer US Inc. for their support of this project.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS

a	Major axis of an ellipse
b	Minor axis of an ellipse
b5	Volute passage width
BCR	Base circle radius
c	Velocity
C_L	Center line
m	Mass
p	Pressure
Ph	Parabola height
R	Radius
Rc	Radius of centroid
W	Turbine base width
α	Angle through which R1 spans
ρ	Density
θ	Angle of intersection for R1 and R2

CHAPTER 1

INTRODUCTION TO VOLUTES

1.1 Introduction

Most centrifugal compressor and radial turbine design requirements need solutions to two major problems : stress and aerodynamics. For both, aerodynamic design and stress analysis of a centrifugal compressor stage or component, a complete geometrical modeling is necessary.

The basic elements of a centrifugal compressor are a stationary casing containing a rotating impeller followed by a radial diffuser. 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, which is marked in Figure1.1, is usually referred to as the inducer. In order to increase the angular momentum, the impeller whirls the fluid outwards and turns it into a 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 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 an additional pressure rise in this part of the compressor.

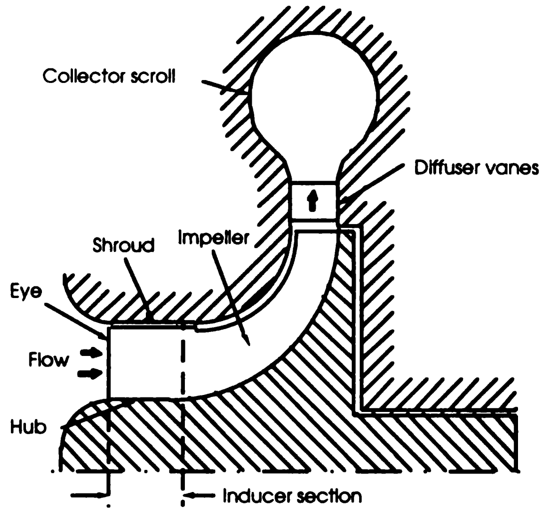

Figure 1.1 Cross-section of a centrifugal compressor

1.2 Basic Elements of a Centrifugal Compressor

Centrifugal compressors are reliable, compact, and robust; they have better resistance to foreign object damage, and are less affected by performance degradation due to fouling. They are found in small gas turbine engines, turbochargers, and refrigerators and are used extensively in the petrochemical and process industry. Since the centrifugal compressor finds a wide variety of application, each application places its own demands on the design of the compressor. The problems usually can be categorized as stress or aerodynamic related.

The stress problems are caused by material strength limitations, and the capability to accurately predict, blade and rotor steady state and vibrational stress for complex impeller shapes and at high rotational speeds.

The aerodynamic problem is to efficiently accomplish, large air deflections and diffusion at high flow velocity, with the added difficulty of very small passage flow areas required to give good efficiency and high pressure ratio. Even though the individual components of the compressor are capable of achieving high efficiency, it is the efficiency of the whole stage that is of great importance. Thus, component matching is an essential aspect of design. It is often required to redesign one or more components of the compressor due to improper matching and sometimes the efficiency of a component is sacrificed to achieve good matching.

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 50% of the total pressure at impeller exit will be dynamic pressure and 50% 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 the engine valves. This recovery therefore must be attempted in the diffuser after the impeller and in the volute.

Compressor volutes and turbine scrolls are widely used for turbochargers, because of their simple structure, easy production and wide operating range. However, there are only very few documented data on internal flow measurements on volutes and scrolls, because of their complex geometry and form, detailed flow pattern have not been obtained. Therefore, the effects of geometrical parameters on performance have not been clarified enough.

The heart of the compressor is the impeller, where all the energy transfer takes place; it would be the likely suspect for any short comings 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 current strong demand, by turbocharger users for turbochargers with good range and efficiency, is forcing the turbocharger designer to review his aerodynamics. Two fruitful sources for potential improvement in turbocharger performance are in:

- proper diffuser and volute design and
- good understanding of the factors affecting the stable operating range and pressure recovery of diffusers and volutes.

Both theoretical and experimental investigations of centrifugal compressors are very expensive and time consuming. Because the flow is unsteady, and flow separation is inevitable even with the best of designs. In addition there is still no clear knowledge for the flow mechanism, particularly at the rotor exit region, impeller-diffuser and impeller-volute interactions.

1.3 General Design Approach

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 scroll then becomes elaborate. A full such analysis is given by Brown and Bradshow /1/. This calculation procedure furnishes the cross-sectional area but gives no guidance as to the shape. Brown and Bradshaw /1/ 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 /2/ showed that certain volute geometry are more efficient than others.

Stiefel /3/ 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 10 to 15 per cent smaller than that which would be designed through the frictionless flow assumption. By reducing the size of the volute by 30 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 to 1 to one of 6 to 1.

For radial turbine scroll Chapple, Flynn and Mully /4/, developed a performance prediction approach and performed design 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.

1.4 Elementary Volute Design

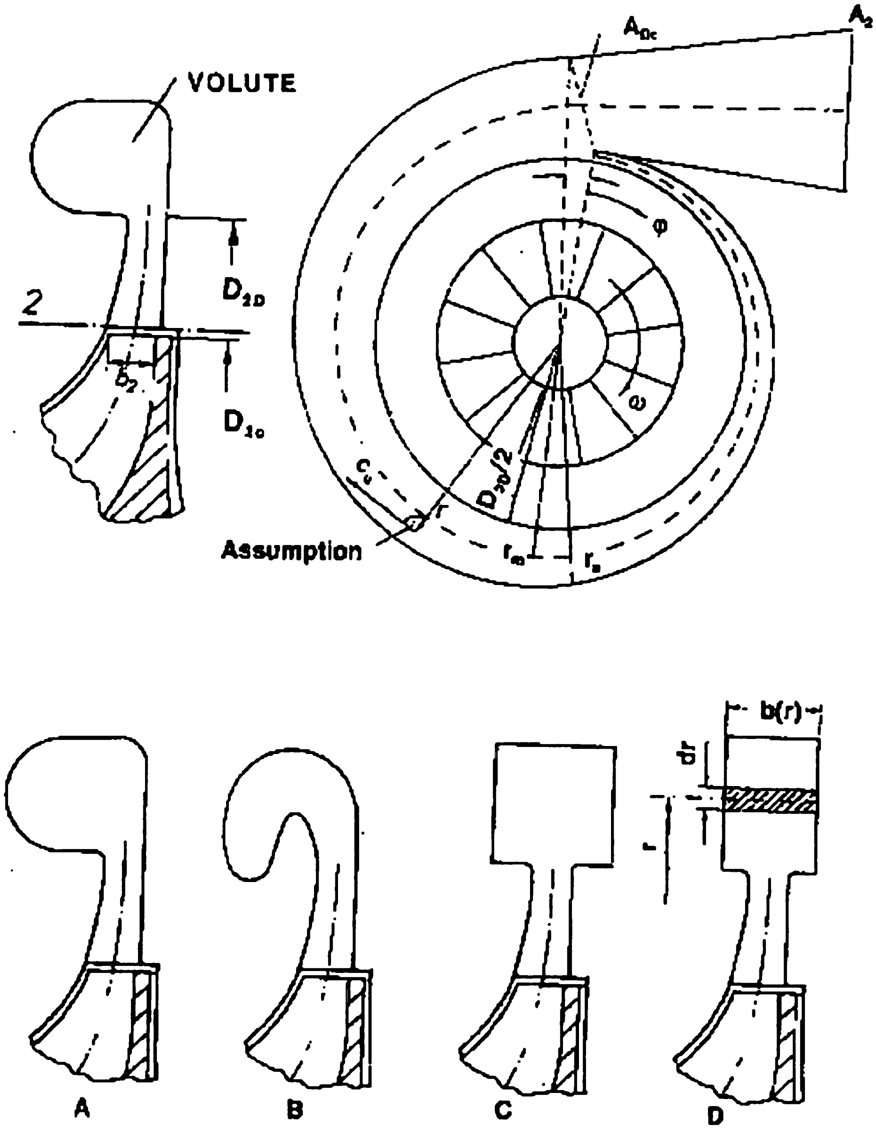

Figure 1.2 Basic Volute Geometry

Figure 1.2 shows typical volute geometry. Using simplified theory, volute flow aims to collect the flow uniformly along the circumference from $\varphi = 0$ to $\varphi = \varphi_{max}$ (usually $360^0 - 8^0 = 352^0$). The mass flow distribution can be described as:

$$m(\varphi) = \frac{m_c}{\varphi_{max}}\varphi + m_o \tag{1.1}$$

where m_c is the effective compressor flow and m_o is the recirculating flow through the tongue gap. Rearranging eq(1.1):

$$\frac{m(\varphi)}{m_c} = \frac{\varphi}{\varphi_{max}} + \frac{m_o}{m_c} \tag{1.2}$$

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

$$\frac{m_o}{m_c} = \mu_z \text{ recirculating mass flow}$$

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

$$\mu(\varphi) = \frac{\varphi}{\mu_{max}} + \mu_z \tag{1.3}$$

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

$$r.c_u = \text{cons tan t} = \frac{D_{2D}}{2}.c_{u2D} \tag{1.4}$$

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

$$dm = dA.\rho.c_u(r) \tag{1.5}$$

$$= b(r).dr.\rho.\frac{D_{2D}}{2}.\frac{c_{u2D}}{r} \tag{1.6}$$

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

$$m(\varphi) = \frac{D_{2D}}{2} c_{u2D} \int_{r=\frac{D_{2D}}{2}}^{r_a} \rho.b(r).\frac{dr}{r} \tag{1.7}$$

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

$$\frac{m(\varphi)}{m_c} = \frac{D_{2D}}{2} c_{u2D} \cdot \frac{\rho(\varphi)}{m_c} \int_{r=\frac{D_{2D}}{2}}^{r_a} b(r) . \frac{dr}{r} = \mu(\varphi) \tag{1.8}$$

Using eq(1.3) one of the basic equations of volutes can be obtained as:

$$\int_{r=\frac{D_{2D}}{2}}^{r_a} \frac{b(r)}{D_{2D}} . \frac{dr}{r} = \frac{\frac{\varphi}{\varphi_{max}} + \mu_z}{\frac{D_{2D}^2}{2} c_{u2D} \cdot \frac{\rho(\varphi)}{m_c}} \tag{1.9}$$

The solution to eq(1.9) describes the distribution of the outer radius r_a as a function of the angle φ.

A very simple solution is usually to assume constant density circumferentially:

$$\rho(\varphi) = \text{cons tant} = \rho_{2D} = \rho_{Dc}$$

and a rectangular cross-sectional area:

$$b(r) = \text{cons tan t} = b_D$$

so that eq(1.9) simplifies to:

$$\int_{r=\frac{D_{2D}}{2}}^{r_a} \frac{dr}{r} = \frac{D_{2D}}{b_D} \frac{\frac{\varphi}{\varphi_{max}} + \mu_z}{\frac{D_{2D}^2}{2} \frac{c_{u2D}}{m_c} \rho_{Dc}} \tag{1.10}$$

and can be solved to give the solution:

$$\ln r\Big]_{r=\frac{D_{2D}}{2}}^{r_a} = \ln \frac{2 r_a}{D_{2D}} = \frac{1}{b_D} \frac{\frac{\varphi}{\varphi_{max}} + \mu_z}{\frac{D_{2D}}{2} \frac{c_{u2D}}{m_c} \rho_{Dc}} \tag{1.11}$$

$$\frac{r_a}{D_{2D}} = \frac{1}{2} e^{\left(\frac{\frac{\varphi}{\varphi_{max}} + \mu_z}{b_D . \frac{D_{2D}}{2} \frac{c_{u2D}}{m_c} . \rho_{Dc}} \right)} \tag{1.12}$$

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

$$\frac{p}{\rho} + \frac{c^2}{2} = \text{cons tant} \tag{1.13}$$

$$\frac{p_{2D}}{\rho} + \frac{c_{2D}^2}{2} = \frac{p}{\rho} + \left(\frac{c_{2D}}{2} \frac{r_{2D}}{r} \right)^2 \tag{1.14}$$

$$\frac{p - p_{2D}}{\rho} = \frac{c_{2D}^2}{2} \left(1 - \frac{r_{2D}}{r} \right) \tag{1.15}$$

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

1.5 Volute Performance

Detailed published information on the performance of volutes is very limited. Japikse /5/ 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 /6/ developed a meanline performance prediction for volutes. Eckert /1/ probably made one of the first attempts to account for friction and secondary losses in volutes. Furthermore Iverson et al /7/, Kurokawa /8/ and Badie et al /9/ 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.

1.6 Secondary flow in volutes

Secondary flows in volutes and scrolls are very little understood. 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 scrolls, 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 scroll 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.

CHAPTER 2

VOLUTE BASICS

2.1 Description

A volute is made up of a circumferential passage around the diffuser exit with progressively increasing cross-sectional area. This area serves to flow fluid velocity and increase static pressure recovery. A full-collection plane is the region where all the mass flow from the diffuser collects. At this location the divider that separates the collected flow from the uncollected flow is called the "tongue". Flow then exits to the connected piping from this location.

The shape of this cross-section can vary and is fundamentally dependent on fluid dynamic principles. The commonly used cross-section are of either circular, elliptical or rectangular type.

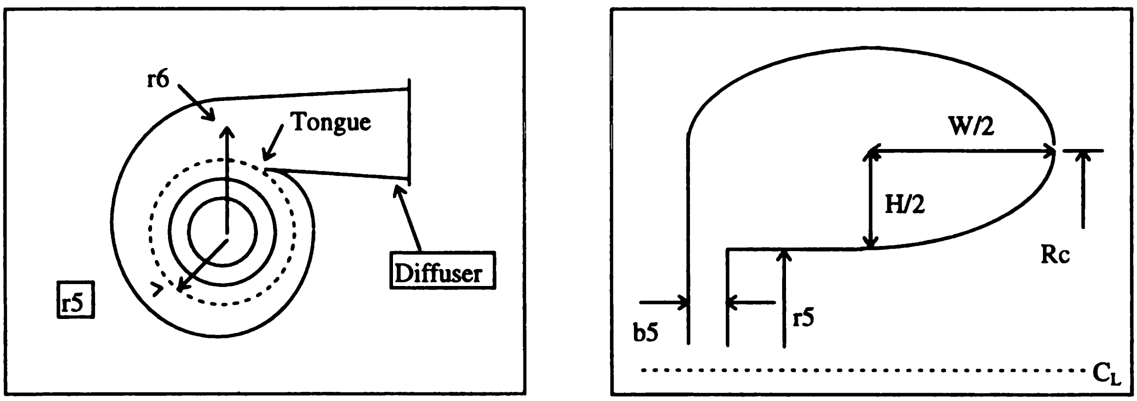

Figure 2.1 Elliptical External Volute

Volutes can be classified as external, internal and intermediate. Figure 2.1 and figure 2.2 is an example of external volutes, where the entire passage area is outside the diffuser exit radius. Figure 2.3 shows an internal volute. The difference is that an internal volute locates as much of the passage area as possible inside the diffuser exit radius. Figure 2.5 shows an intermediate volute. This volute may appear to differ from that of figure 2.1 only by the fact that the diffuser wall is extended into the volute. As the area increases the R_c for the external volute will also increase.

2.2 Design Concepts

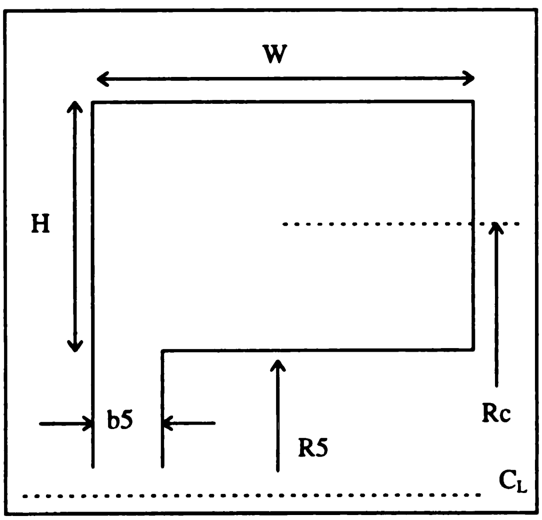

Figure 2.2

Rectangular External Volute

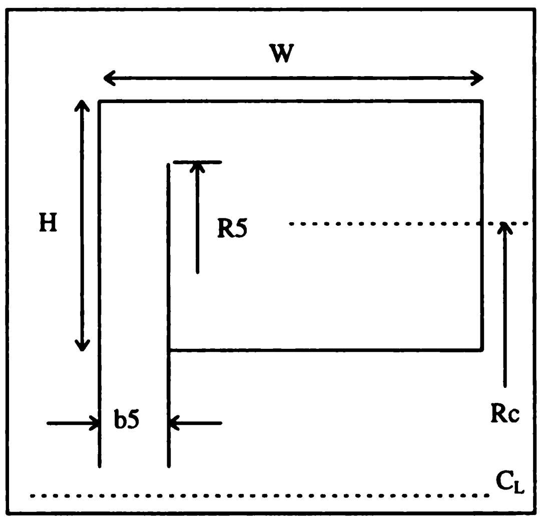

Figure 2.3

Rectangular Internal Volute

To obtain the geometry of cross-section it is necessary to define the circumferential variation of R_c and the cross-sectional area. [In this thesis we concentrate on external volutes and therefore need to define the cross-sectional area variation.] For simplicity the cross-sectional area is assumed to be a function of θ that ranges from 0 -

360 around the volute profile. For various types of cross-sections the area calculations are different and are presented below.

2.3 Constant Centroid and Constant Inner Diameter Compressor Cross-sections

The two type of cross-sections concentrated on in this study are of the "constant inner diameter" and the "constant centroid" type. Figure 2.4 and Figure 2.6 illustrate the difference in cross-section amongst these two types.

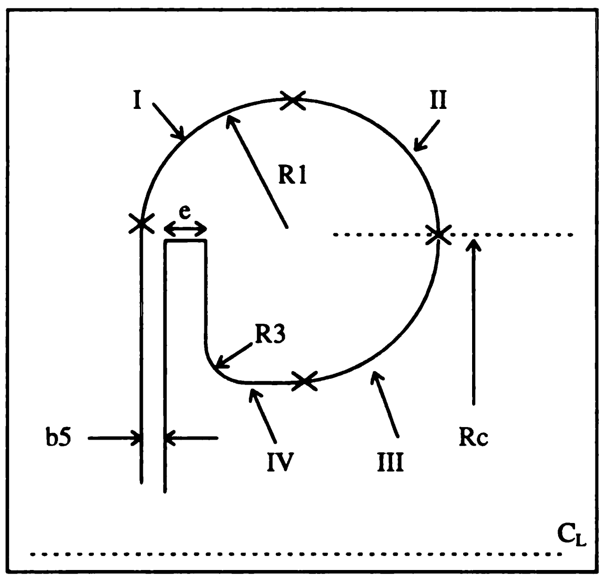

Figure 2.4 Constant Centroid (Circular c/s)

A. Constant Centroid Circular Cross-section

As illustrated in Figure 2.4 the calculation of the area of cross-section depends on the area of the circle and the radius R3. If we divide the circular cross-section into four quadrants we see that the area of three quadrants (I, II, III) are the simple area of a circle. The fourth quadrant (IV) however has to be calculated differently. From the figure we can see that the area of the fourth quadrant can be broken down into two parts, i.e. the area of a rectangle and the area outside of the curve with radius R3.

$$\text{Area (rectangle)} = R1 * (R1 - b5 - e) \tag{2.1}$$

$$\text{Area (outside curve)} = (R3)^2 * \left(1 - \frac{1}{4} * \pi\right) \tag{2.2}$$

Subtracting these two areas we obtain the area of the fourth quadrant. Adding up all four quadrants we then get the area of cross-section.

$$\text{Area} = \tfrac{3}{4}\,\pi\,(R1)^2 + [\text{Area (rectangle)} - \text{Area (outside curve)}] \tag{2.3}$$

B. Constant Centroid Elliptical Cross-section

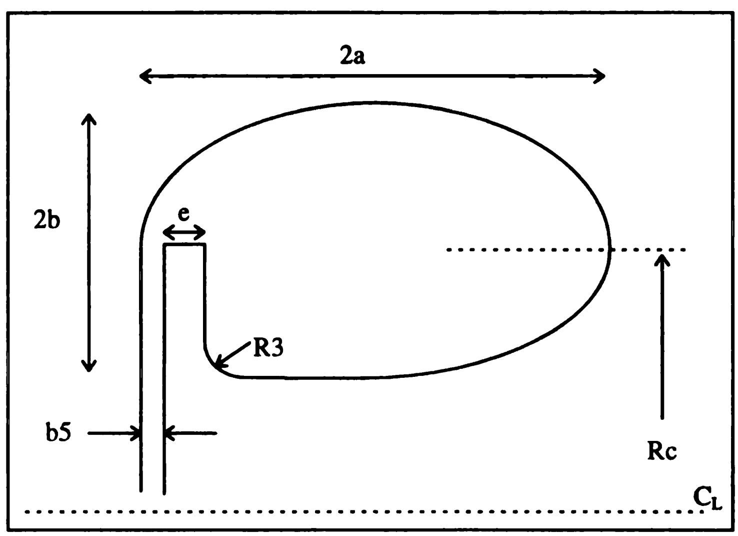

Figure 2.5 Constant Centroid (Elliptical c/s)

The calculation for the cross-section area is similar to the circular cross-section. In this case the major axis and minor axis of the ellipse are known. Again if we breakdown the whole cross-section into four quadrants, we use the formula for the area of the ellipse (Area = πab) to calculate the area of the first three quadrants. The fourth quadrant is calculated similarly as above for the circular cross-section. The area is broken down into the area of a rectangle and area outside the curve R3.

$$\text{Area (rectangle)} = b * (a - b5 - e) \tag{2.4}$$

$$\text{Area (outside curve)} = (R3)^2 * (1 - \frac{3}{4} * \pi) \tag{2.5}$$

Again by subtracting these areas the area to the fourth quadrant is obtained. Adding up all the areas of the various quadrants the area of the cross-section is thus calculated.

$$\text{Area} = [\, ¾\, \pi\, a\, b\,] + [\text{Area (rectangle)} - \text{Area (outside curve)}] \tag{2.6}$$

C. Constant Centroid Rectangular Cross-section

Although rectangular cross-section types are not often used in turbochargers, using similar methods as in elliptical and circular type, different geometry's were derived. Again the calculation of the area of cross-section is done in the same manner. The area is broken down into four quadrants and the first three are calculated by using simple methods. The fourth quadrant is calculated the same way as in the circular and elliptical cross-section.

D. Constant Inner Diameter Circular Cross-section

As in figure 2.6 we see that the circular cross-section can also be broken down into various quadrants. First we calculate the area of the first three quadrants using the three-fourths area of a circle. Then the fourth quadrant is broken up into three parts. These three parts are the area of a square, rectangle and area outside the curve R3.

$$\text{Area (square)} = (R1)^2 \tag{2.7}$$

$$\text{Area (rectangle)} = R3 * (b5 + e) \qquad (2.8)$$

$$\text{Area (outside curve)} = (R3)^2 * (1 - \frac{1}{4} * \pi) \qquad (2.9)$$

$$\text{Area (4}^{th}\text{ quadrant)} = \text{Area (square)} - \text{Area (rectangle)} - \text{Area (outside curve)} \qquad (2.10)$$

Totaling up the areas of the four quadrants the area of the cross-section is obtained.

$$\text{Area} = \tfrac{3}{4}\pi (R1)^2 + \text{Area (4}^{th}\text{ quadrant)} \qquad (2.11)$$

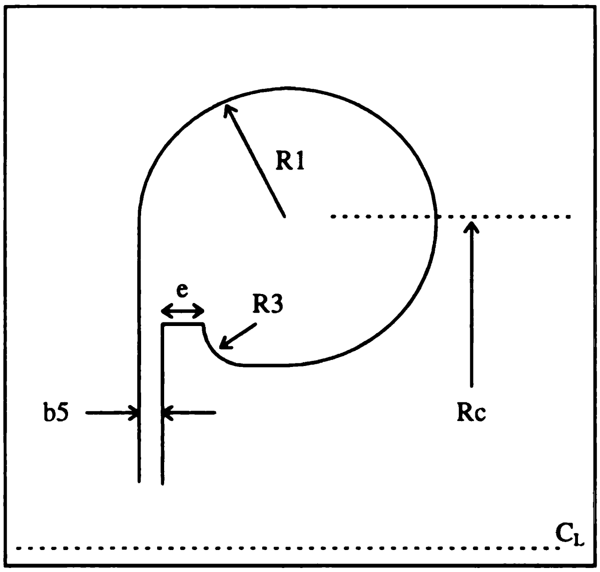

Figure 2.6 Constant Inner Diameter (Circular c/s)

E. Constant Inner Diameter Elliptical Cross-section

Again this calculation process is similar to the constant inner diameter circular cross-section. The total area is sub-divided into 4 quadrants and the area of the first three quadrants are calculated using the area of an ellipse formula (Area = πab). The fourth quadrant is broken down into smaller areas which are similar to the calculation procedure of the circular cross-section. The formulas differ slightly and are as follows.

$$\text{Area (rectangle-1)} = a * b \qquad (2.12)$$

$$\text{Area (rectangle-2)} = R3 * (b5 + e) \quad (2.13)$$

$$\text{Area (outside curve)} = (R3)^2 * (1 - \frac{3}{4} * \pi) \quad (2.14)$$

$$\text{Area (4}^{th}\text{ quadrant)} = \text{Area (rectangle-1)} - \text{Area (rectangle-2)} - \text{Area (outside curve)} \quad (2.15)$$

where Area (rectangle-1) is the area of the rectangle using the sides as of the major and minor axis. Totaling up all the quadrants the area of the cross-section is obtained.

$$\text{Area} = \tfrac{3}{4} \pi a b + \text{Area (4}^{th}\text{ quadrant)} \quad (2.16)$$

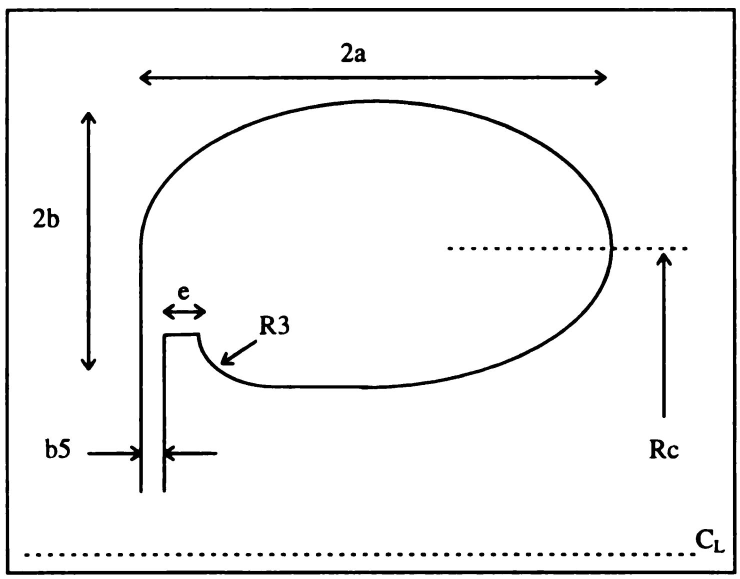

Figure 2.7 Constant Inner Diameter (Elliptical c/s)

F. Constant Inner Diameter Rectangular Cross-section

As in the constant centroid rectangular cross-section this type is not often used, but an approximation was made using the existing circular and elliptical cross-section drawings. The area is calculated similarly to the area of the circular and elliptical cross-section where the area is broken down into various quadrants and calculated individually.

2.4 Open Flow Turbine Area of Cross-section

The open flow turbine internal volute passage is similar to the centrifugal passage and differ only in cross-sectional shape. As described in figure 2.8, the cross-section is defined by three dimensions which are R1, R2, and S. The outline is developed by tangentially intersecting two curves of radius R1, R2 and a parabola. Described below is the method used to calculate the area of cross-section knowing R1, R2 and S. Since the cross-section is symmetrical we can concentrate on the area of one side of the central axis. This cross-section is broken down into 3 different regions and each of these are determined separately.

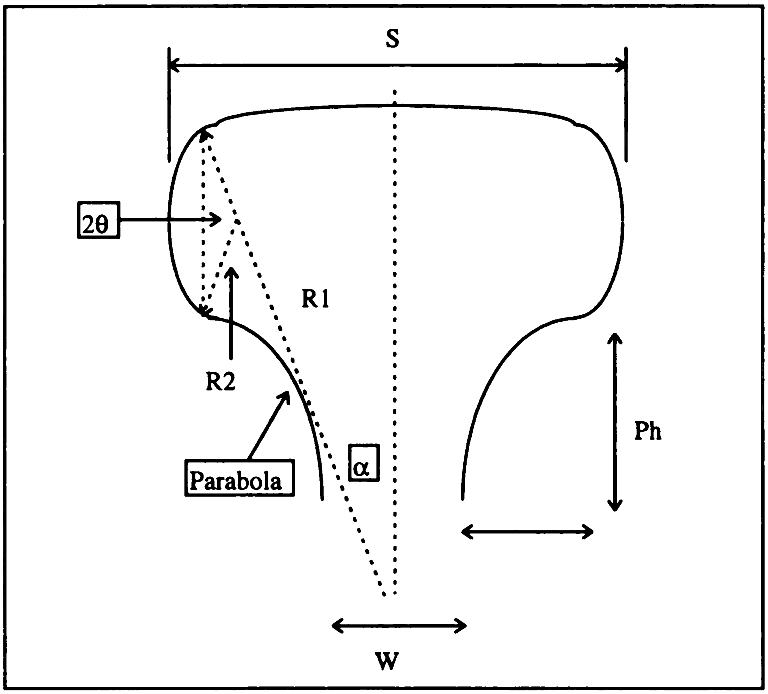

Figure 2.8 Dimension Details.

As in the figure 2.9 the cross-section is divided into 3 parts alphabetically as A, B, and C. Section A is the shaded region and using geometric methods, the formula to calculate the area is as below. The area is subdivided into three smaller regions which are denoted by different shades in the figure below. The two regions can be described as a circular segment of a circle and a rectangle

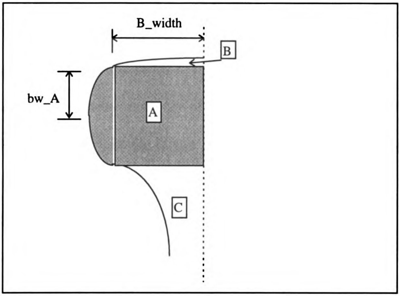

Figure 2.9 Subdivision for area calculation.

$$\text{Area - A (rectangle)} = (\text{B\_width}) * (2 * \text{bw\_A}) \qquad (2.17)$$

$$\text{Area - A (circular segment)} = R^2 * (\theta - \tfrac{1}{2}\,\text{Sin}\,(2\theta)) \qquad (2.18)$$

$$\text{Area (A)} = \text{Area - A (rectangle)} + \text{Area - A (circular segment)} \qquad (2.19)$$

Area B can be described as a circular segment and the area for region B is,

$$\text{Area (B)} = R^2 * (\alpha - \tfrac{1}{2}\,\text{Sin}\,(2\alpha)) \qquad (2.20)$$

Area C can be subdivided into a rectangle and parabola. Using the formulas below the area to this region are calculated.

$$\text{Area - C (rectangle)} = (\text{B\_width}) * (\text{Ph}) \qquad (2.21)$$

$$\text{Area - C (parabola)} = \frac{2}{3} * (\text{B\_width} - \text{W}) * (\text{Ph}) \qquad (2.22)$$

$$\text{Area (C)} = [\text{Area - C (rectangle)}] - [\text{Area - C(parabola)}] \qquad (2.23)$$

Adding up all the areas ,

$$\text{Area (Total)} = \text{Area (A)} + \text{Area (B)} + \text{Area (C)} \qquad (2.24)$$

2.5 Twin Flow Turbine Area of Cross-section

The area of a twin flow turbine is similar to the open flow turbine. The only addition to the open flow is the existence of a divider wall that separates the flows. Also the construction of the cross-section is now dependent on an additional parameter which is R3 (as detailed). To determine the area of the twin flow cross-section the area of the divider wall has to be calculated. This divider wall is broken down into three different area, Div(A), Div(B) and Div(C). The formulas used for these areas are,

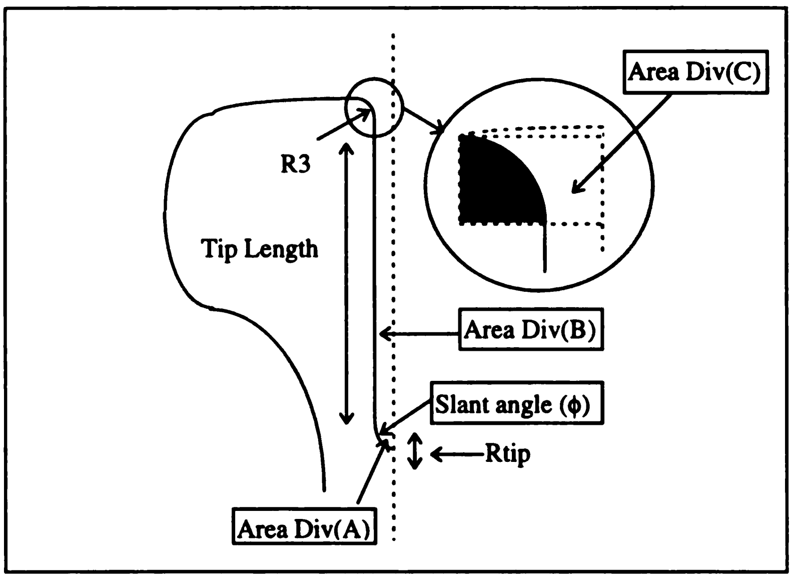

Figure 2.10 Open Flow Divider Wall Details

$$\text{Div(A)} = [\pi * (\text{Rtip})^2] / 4 \qquad (2.25)$$

$$\text{Div(B)} = [\text{Tip Length}] * \text{Rtip} + \frac{[(\text{Tip Length})^2 * \text{Sin}(\phi)]}{2} \quad (2.26)$$

The area for region C (as drawn in the figure) can be considered as the combination of a rectangle and a sector of a circle having radius R3.

$$\text{Div(C rectangle)} = ([\text{Tip Length}] * \text{Sin}(\phi) + \text{R3}) * \text{R3} \quad (2.27)$$

$$\text{Div(C sector)} = [\pi * (\text{R3})^2] / 4 \quad (2.28)$$

$$\text{Div(C)} = \text{Div(C rectangle)} - \text{Div(C sector)} \quad (2.29)$$

Adding up all the areas and using the formula described earlier the area for an open flow turbine can be determined,

$$\text{Area (Divider wall)} = \text{Div(A)} + \text{Div(B)} + \text{Div(C)} \quad (2.30)$$

$$\text{Area (Twin flow)} = \text{Area (Open flow)} - \text{Area (Divider wall)} \quad (2.31)$$

2.6 Radius Of Centroid for various cross-sections.

For compressor cross-section the calculation of the centroid radius is fairly simple. As the detailed figures earlier denoted, it can be seen from the area of cross-section that the centroid radius is considered to be at the central Y-axis of the volute passage. But for the turbine open flow and twin flow a detailed method is to be used.

A. Open Flow

Again the cross-section area is subdivided into the same regions as for the area calculation. Now only the centroid radius for these individual regions are needed for the overall centroid radius. They are calculated as follows,

$$\text{Rc Area (A)} = \text{Ph} + \text{bw\_A} \quad (2.32)$$

$$\text{Rc Area (B)} = \text{Ph} + (\text{bw\_A} * 2) + \left[\frac{\text{R1} - (\text{Ph} + (\text{bw\_A} * 2))}{3}\right] \tag{2.33}$$

$$\text{Rc Area (C)} = \left(\frac{3}{5}\right) * \text{Ph} \tag{2.34}$$

Using this the overall centroid radius is given by

$$\text{Rc} = \left[\frac{\text{Rc Area(A)} * \text{Area(A)} + \text{Rc Area(B)} * \text{Area(B)} + \text{Rc Area(C)} * \text{Area(C)}}{Area(total)}\right] \tag{2.35}$$

B. Twin Flow

In this case the centroid radius is calculated the same way except that the divider wall is taken into consideration. As described earlier the divider wall is broken down into three smaller part and the centroid radius for each of these are,

$$\text{Rc Div(A)} = 0 \tag{2.36}$$

This is due to the fact that this region is below the base circle radius and hence not considered.

$$\text{Rc Div(B)} = \left[\frac{[\text{Tip Length}] - (\text{Bcr} - \text{Dwr})}{2}\right] \tag{2.37}$$

$$\text{Rc Div(C rectangle)} = [\text{Rc Div(B)} * 2] + \left(\frac{R3}{2}\right) \tag{2.38}$$

$$\text{Rc Div(C sector)} = [\text{Rc Div(B)} * 2] + \left[\frac{(R3 * 4)}{(3 * \pi)}\right] \tag{2.39}$$

$$\text{Rc Div (C)} = \left[\frac{\text{Rc Div(C rectangle)} * \text{Div(C rectangle)} + \text{Rc Div(C sector)} * \text{Div(C sector)}}{Div(C)}\right] \tag{2.40}$$

The centroid radius of the divider wall is obtained by,

$$\text{Rc Div} = \left[\frac{\text{Rc Div(B)} \ * \ \text{Div(B)} \ + \ \text{Rc Div(C)} \ * \ \text{Div(C)}}{\text{Area(Divider wall)}}\right] \tag{2.41}$$

The overall centroid radius is determined by,

$$\text{Rc} = \left[\frac{\text{Rc (Open Flow)} \ * \ \text{Area (Open Flow)} \ - \ \text{Rc Div} \ * \ \text{Area (Divider wall)}}{\text{Area (Twin Flow)}}\right] \tag{2.42}$$

Where Rc is the radius of centroid for a twin flow cross-section.

CHAPTER 3

DESCRIPTION OF VISUAL BASIC APPLICATION

This chapter deals with detailed explanation of the Visual Basic Application and all associated forms and procedures. The motivation for using Visual Basic 4.0 was the fact that further interfacing might be possible with the existing Visual Basic application currently used by Schwitzer. Also Visual Basic is windows based, graphical and easy to use.

Overview of the application

The Visual Basic Application uses various forms and procedures that are as listed below.

1) Drawing Form - (frmdrawing.Frm)

2) Compressor Data Form - (frmComp_data.Frm)

3) Turbine Data Form - (frmTurb_data.Frm)

4) Compressor Module - (ModCompressor.Bas)

5) Turbine Module - (ModTurbine.Bas)

6) Parameter Form - (frmpara.Frm)

3.1 Drawing Form - (frmdrawing.Frm)

This form is the main form and the backbone of the application. It consist of three command buttons and a picture box as detailed in Figure 3.1. This form does no detailed calculation but is used only to manage the Compressor data form and Turbine data form. The picturebox also displays the existing drawing except on start up. A few details on its individual operation are listed below.

3.1.1 End Command Button - (cmdend)

This command button ends the application and exits out of the run mode.

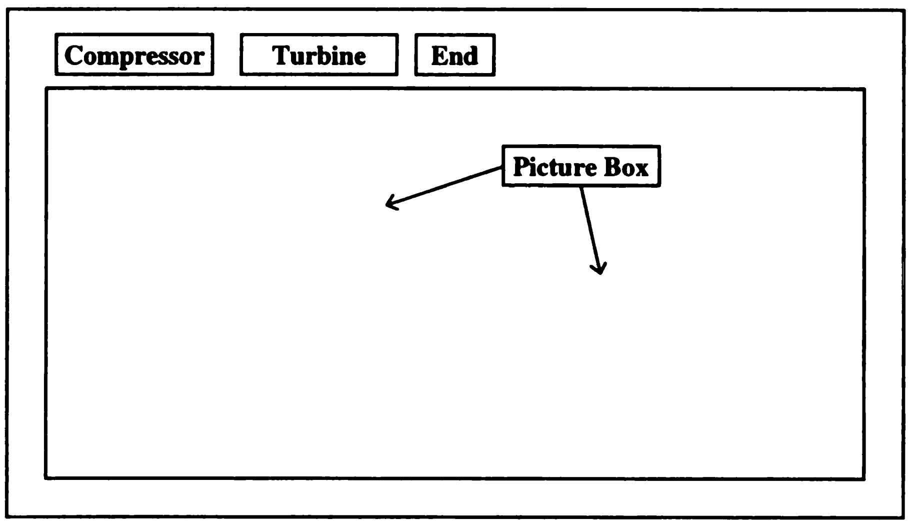

Figure 3.1 - Drawing Form

3.1.2 Compressor Command Button - (cmdcomp)

This button pulls up the Compressor Data Form which list all compressor input parameter details.

3.1.3 Turbine Command Button - (cmdTurbine)

This button executes and displays the Turbine Data Form which in turn list all input parameter details required for the turbine profile generation.

3.1.4 Picture Box - (picbox)

The picture box is used as a drawing sheet and displays the generated plots from the compressor and turbine modules.

3.2 Compressor Data Form - (frmComp_data.Frm)

The compressor data form is used to list and display all assumed data needed for the volute and cross-section plot generations.

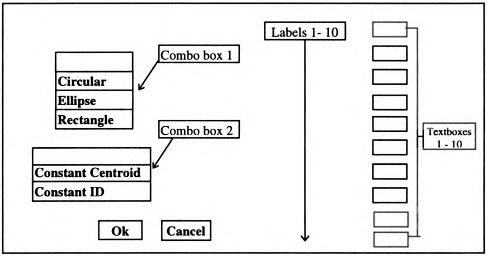

Figure 3.2 - Compressor Data Form

3.2.1 Combo Box 1 - (combo1)

This combo box lists the various types of cross-section that can be chosen. The options are Circular, Ellipse and Rectangular. The default selection is the circular cross-section and is displayed in the box during start up. After the first plot the compressor data form is recalled then the existing selection is displayed in the combo box.

3.2.2 Combo Box 2 - (combo2)

This combo box list the various types of volute generations available. In this project the two options are Constant Centroid and Constant Inner Diameter. The default selection is the Constant Centroid and is displayed in the box after start up. Again, if the compressor data form is recalled at any point the existing selection is displayed for convenience.

3.2.3 Cancel Command Button - (cmdCancel)

This command button disregards any changes made to the compressor data form and then displays the Drawing Form with the existing plot.

3.2.4 Text Boxes 1 - 10

These textboxes define various input parameters for the volute and cross-section generations. They display the most current values used in the last generated plots. These textboxes also facilitate to make changes to the drawings and adjust various parameters so as to obtain the desired output. The list of text boxes are as below,

1) "txtArth" defines the area at the throat.

2) "txtratio" defines the A/R ratio at the throat.

3) "txtGapwidth_factor" defines passage width leading into the volute passage.

4) "txtEdgewidth_factor" defines flat edge width on entry into the volute passage.

5) "txtInn_Radius" defines inner radius of the compressor.

6) “txtRadius_R3” defines radius R3.

7) “txtTolerance” defines the Tolerance limits for area calculation.

8) “txtDiff_length” defines diffuser length.

9) “txtDiff_area_ratio” defines the ratio of the areas of the diffuser and throat.

10) “txtVolute_scale_comp” defines the scale at which the volute profile needs to be drawn.

3.2.5 Labels 1 - 10

The labels used are in accordance to the textboxes available and are used purely to define each textbox. They have no other function other than labeling each textbox on the Compressor Data form. The list of labels used are,

1) “LabArth” displays “Arth=”

2) “Labratio displays “A/R=”.

3) “LabGapwidth_factor” displays “Gap width factor=”.

4) “LabEdgewidth_factor” displays “Edge width factor=”.

5) “LabInn_Radius” displays “Inner radius=”.

6) “LabRadius_R3” displays “R3”.

7) “LabTolerance” displays “tolerance=”.

8) “LabDiff_length” displays “Diffuser length=”.

9) “LabDiff_area_ratio” displays “Area[diff]/Area[throat]=”.

10) “LabVolute_scale_comp” displays “Volute Scale=”.

3.2.5 OK Command Button - (cmdOK)

This is the actual control code on the Compressor Data Form. The "Ok" command first reads data from the text boxes and combo boxes into variables. It then uses "if" statements to decide what procedure to call from the data obtained in the combo boxes. If the data in the combo boxes do not match the existing available procedure then an error message is displayed so the user can correct the data. Once the information from the combo boxes are matched by the "if" statements, the required procedure is called. Along with the call for the subroutine, the data read previously from the textboxes are also submitted to the procedures and the calculation for the plot are initialed.

3.3 Turbine Data Form - (frmTurb_data.Frm)

This form is similar to the Compressor data Form with a few changes. The display of the form is as below.

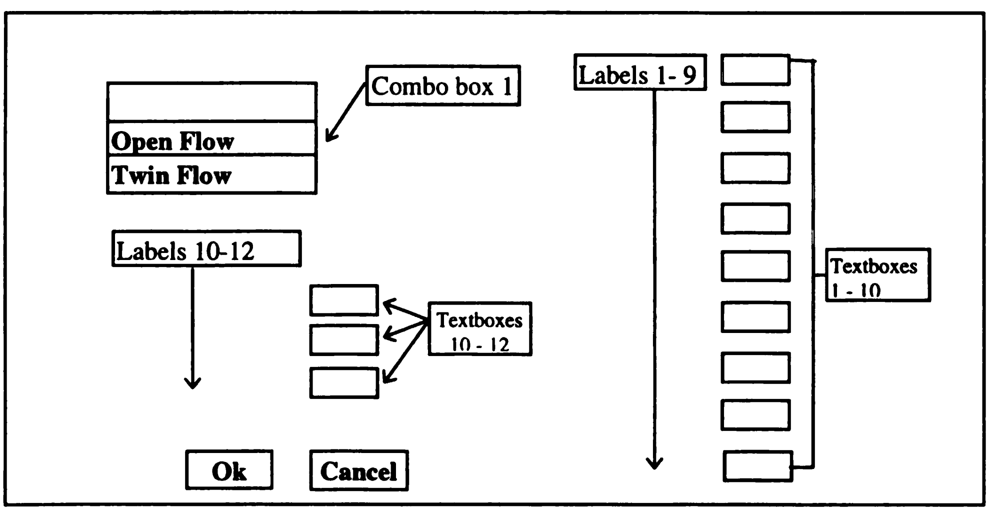

Figure 3.3 Turbine Data Form

3.3.1 Combo Box 1 - (Combobox1)

In the Turbine data form there is only one combo box. The options for the combo box are Open Flow or Twin Flow type of turbine housing. The default selection on start up is the Open Flow and is displayed in the combo box. Again as in the case of the compressor data form combo boxes the recent selection is displayed after the first plot.

3.3.2 Cancel Command Button - (cmdCancel)

This works in the same way as the Compressor Data Form command button and disregards any changes to the Turbine data form and recalls the Drawing Form with the existing plot.

3.3.3 Text boxes 1 - 12

These textboxes work in a similar manner to the Compressor Data Form to define all data needed to generate the turbine volute and cross-section plots. A difference in these text boxes is that from one to nine is used for common data of both open and twin flow while the last three are used exclusively for the twin flow procedure. The list of text boxes used are as below,

1) "txtArth_turb" defines the area at the throat of the turbine.

2) "txtRatio_turb" defines the A/R ratio at the throat.

3) "txtBase_circ_radius" defines the base circle radius.

4) "txtWidth" defines the width of the passage leading into the volute passage.

5) "txtTolerance_turb" defines the tolerance limits for area calculation.

6) "txtLower_base_radius" defines the radius at the lower width of the passage.

7) "txtDiff_Length_turb" defines the length of the diffuser.

8) "txtDiff_area_ratio_turb" defines ratio of the area of diffuser outlet and throat.

9) "txtVolute_scale" defines the scale at which the volute profile is to be plotted.

These are used only for the Twin flow turbine housings,

10) "txtRadius_R3" defines radius R3

11) "txtRadius_tip" defines the radius at the tip of the divider wall.

12) "txtDivider_angle" defines the angle at which the divider wall is inclined at.

3.3.4 Labels 1 - 12

Again these labels are used to define each text box on the Turbine Data Form. The labels used are listed below,

1) "LabArth_turb" displays "Arth=".

2) "LabRatio_turb" displays "A/R=".

3) "LabBase_circ_radius displays "BCR=".

4) "LabWidth" displays "Width=".

5) "LabTolerance_turb" displays "Tolerance=".

6) "LabLower_base_radius" displays "LowerBCR=".

7) "LabDiff_Length_turb" displays "Diffuser Length=".

8) "LabDiff_area_ratio_turb" displays "Area[Diff]/Area[Throat]=".

9) "LabVolute_scale" displays "Volute Scale=".

These are used only for the Twin flow turbine housings,

10) "LabRadius_R3 displays "Radius R3=".

11) "LabRadius_tip" displays "Tip Radius=".

12) "LabDivider_angle" displays "Divider Slant=".

3.3.5 OK Command Button - (cmdOK)

Again this is the actual control code on the Turbine Data Form. The Ok command first reads data from the text boxes and combo boxes into variables. It also reads in data needed for only the twin flow whether that procedure is executed or not. It then uses "if " statements to decide what procedure to call from the data obtained in the combo boxes. If the data in the combo boxes do not match the existing available procedure then an error message is displayed so the user can correct the data. Once the information from the combo boxes are matched up by the "if" statements the required procedure is then called. Along with the call for the subroutine the data read previously from the textboxes are also submitted to the procedures and the calculation for the plot are initialed. It may also be noted that the data needed for only the twin flow turbine housing is not submitted to the open flow procedure.

3.4 Compressor Module - (ModCompressor.Bas)

This module houses all the codes used for the various types or combinations of compressor housings plot generations. The module is executed by the OK command button on the compressor data form. A schematic diagram of the layout is as figure 3.4,

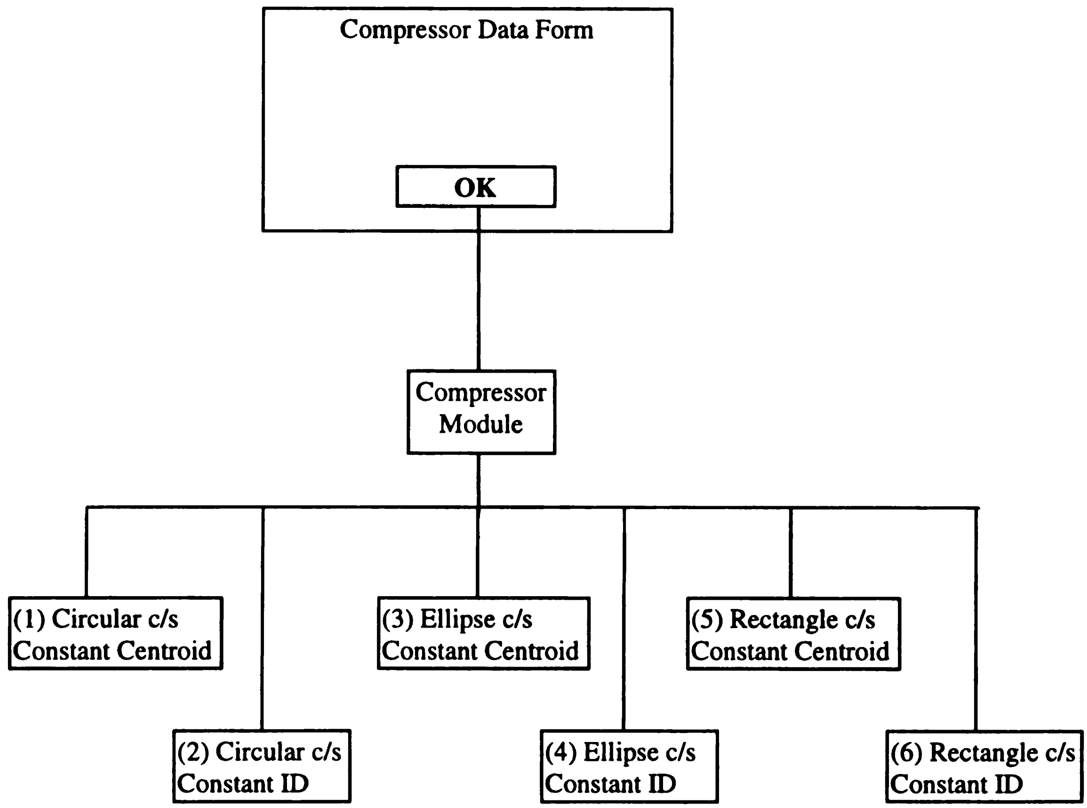

Figure 3.4 Compressor Module

The Ok command button decides which procedure listed under the module is to be called. The Compressor Module contains the procedures and also for general declarations common for all procedures. The procedures are described in detail in the next chapters and a list of the procedures contained in the compressor module is as below,

1) SubCircular_ConstCentroid
2) SubCircular_ConstID
3) SubEllipse_ConstCentroid
4) SubEllipse_ConstID
5) SubRectangle_ConstCentroid

6) SubRectangle_ConstID

7) FillCells

3.5 Turbine Module - (ModTurbine.Bas)

This module contains all the Turbine generation procedures but which are different form the compressor module procedures. A schematic of the procedure control is detailed below,

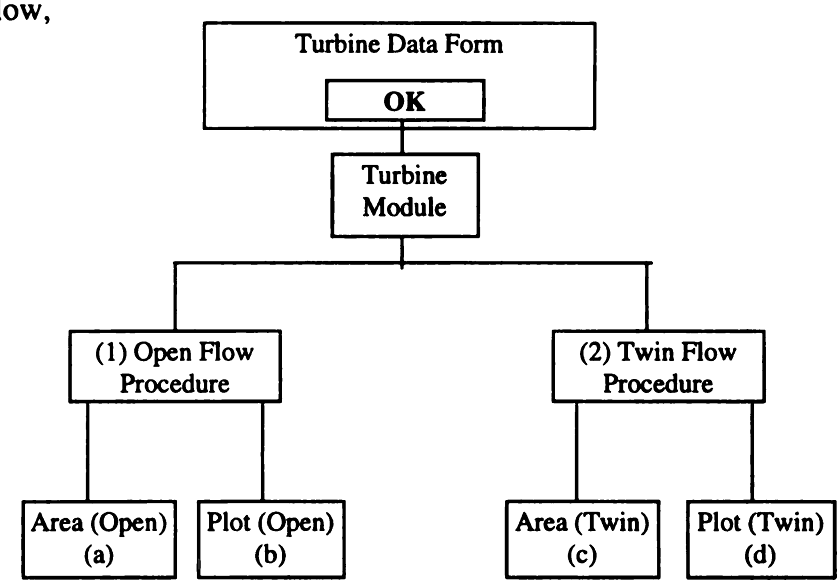

Figure 3.5 Turbine Module

The Ok command in the turbine data form calls the required procedure. The main procedure are the Open Flow and the Twin Flow, these in turn use the Area and Plot procedure to calculate the area of the cross-section and plot, respectively. A detailed explanation of these subroutines are described in the later chapters. The actual names of these procedures housed by the Turbine Module are as described,

1) SubOpen_Flow

a) Area

b) plot_Turbine_profile

2) SubTwin_flow

c) Area_twin_flow

d) plot_twin_flow_turbine

3.6 Parameter Form - (frmpara.Frm)

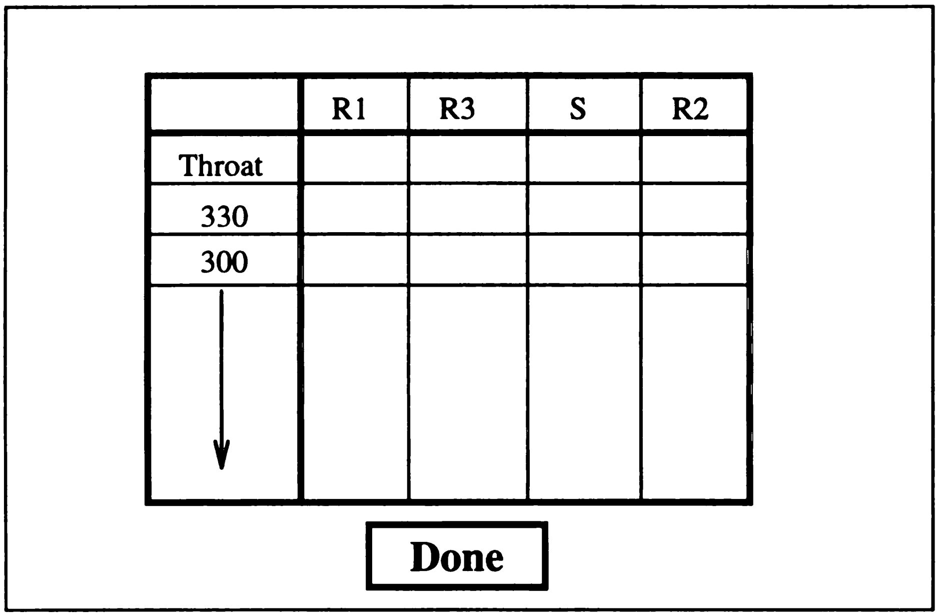

Figure 3.6 Parameter Form

This form is a support form and is used for data output. This form was designed so as to replicate the data tables used in the existing Schwitzer drawings. Also data can be read off this form and therefore found to be a convenient interface the second part of the project. In this form a "grid" is used and the tittles to this grid is specified under "Form Load". On the "click" event of the Dimension command button the form is displayed.

The data is entered in to the grid via calling the "Fillcells" procedure which is located in the Compressor Module.

3.7 FillCells Procedure

This procedure writes data into the cells which were generated previously by the main procedures. A loop is constructed in each of the main subroutines and the number of loops is equal to the number of rows in the grid (12). These loops send data to the "FillCells" procedure column by column and the function of the procedure is to read the data from the "call" command and write it to their respective cells. The data is written column by column and is terminated after all cross-sections of 30 degree intervals are recorded.

CHAPTER 4

DESCRIPTION OF COMPRESSOR VOLUTE AND CROSS-SECTION GENERATION MODULE

This chapter deals with detailed explanation of the compressor module, data generation and methods used to plot the data. The various procedures were grouped into one module so as to call them from other codes. Also having these in a subroutine form makes it flexible for future mating with other Visual Basic applications.

4.1 Compressor Module - (ModCompressor.Bas)

The module consist of six different procedures and is as detailed in Chapter 3. The "Ok" command on the Compressor Data Form matches up the input data and decides using "if" statements on which subroutine is to be called. In the "General Declarations" all values from the input data textboxes are defined as real numbers and are made available to all the procedures in the module. The list of available subroutines are,

1) SubCircular_ConstCentroid
2) SubCircular_ConstID
3) SubEllipse_ConstCentroid
4) SubEllipse_ConstID
5) SubRectangle_ConstCentroid
6) SubRectangle_ConstID

These procedures work very similar to each other with minor differences in cross-section details. A list of procedures are listed and the visual basic application executes these in the order displayed as below,

1) Reading Input Data
2) Dimension and Area check calculations
3) Plotting the Inner Circle
4) Plotting the Centroid
5) Plotting the Volute Profile
6) Plotting the Diffuser
7) Plotting the straight line surfaces (Cross-section)
8) Plotting three-fourths of Cross-section View
9) Plotting radius R3 and rest of Cross-sectional View

4.2 Reading Input Data

This is common for all procedures and is the area in which all the input variables are defined. The various values assigned to the text boxes in the Compressor Data Form are read by the "OK" command and then sent via the call command. These values are then assigned to variables in the procedures. It can be seen in the code that all text boxes as described in Chapter 3 are read. In addition to this, a constant value of center coordinates "(center_x_vol, center_y_vol)" are also assigned. These values were selected to center them for the current size of the picture box and were not provided in the Data Form to be easily interfaced with the Autoscaling procedure currently used by Schwitzer. The initial commands erase the previous plot and insert the current plot drawing tittle.

4.3 Dimension and Area check calculations

In this subgroup the area over a range of 0 - 360 of the volute passage is read and the corresponding parameter that make up the cross-section is determined. For convenience the Area distribution though the volute passage was assumed to be linear and a simple formula to evaluate the area is used,

$$\text{Area (angle)} = \text{Area (throat)} * \left(\frac{angle}{360}\right) \tag{4.1}$$

where angle is the volute passage from 0 - 360 degrees. This formula was kept simple to be modified later on for use in non-linear A/R distributions. Using the area at the throat the diameter, minor axis and height for circular, ellipse and rectangular cross-sections, respectively are determined. In case of the circular cross-section the diameter at all angles of the volute passage can be determined as the area is known. In the case of the elliptical and rectangular cross-sections the width to height ratio is needed. This is obtained using the area, centroid radius at the throat. Keeping this width to height ratio constant the required parameters from 0 - 360 can be determined.

After these areas and parameters are recorded, a check for the actual area is carried out. Using the previously generated values the true value of the areas is determined. This is done by using the methods as defined in Chapter 2. Now using this value, a check is carried out against the required area and if the tolerance limits are met then it proceeds to the next degree of volute passage. If the difference in the areas are greater than the required tolerance then one of the parameters are increased or decreased until the tolerance is met. In case of the circular cross-section it is the diameter while for the elliptical and rectangular cross-sections it is the height. This is done for 0 - 360 of the volute passage and the true dimensions at each degree of cross-section is recorded.

4.4 Plotting the Inner Circle

The inner radius of the volute is specified in the Compressor Data Form and this data is read and transferred to the subroutine by the "OK" command. This radius is then plotted over 360 degrees of the volute and since it is constant, the plot is circular. The curve is plotted using the cylindrical coordinate method and using a loop of "theta" varying from 1 to 360 the following X and Y coordinate points are plotted.

$$X(\text{theta}) = \text{Radius} * \sin(\text{theta}) / \text{drawscale} \quad (4.2)$$

$$Y(\text{theta}) = \text{Radius} * \cos(\text{theta}) / \text{drawscale} \quad (4.3)$$

This method is used regularly for all the plots with minor differences. The drawscale is used so as to scale the drawing, as the volute profile at full scale is large as compared to individual cross-sections at different angles. The drawscale value is also specified in the data form. Using these data points obtained lines are drawn from the X and Y coordinate of one angle to the next. As the angle interval is small the plot appears to be a smooth curve.

4.5 Plotting the Centroid

The centroid values of each degree of volute passage are calculated during "Dimension and Area Check" loops using methods described in Chapter 2. These values are then plotted in a manner similar to the way the inner radius is plotted. The difference is that the centroid radius is not a constant value in the Constant Inner Diameter type.

Again the drawscale is used for scaling and the curve is plotted as a number of lines joining the X and Y coordinate point at each degree of cross-section.

4.6 Plotting the Volute Profile

As the centroid around the volute passage is already known and plotted, to get the volute profile only one parameter is needed. In the case of circular cross-section it is the diameter while for the elliptical and rectangular type it is the height. These profiles can be defined as the "inner and outer radii" for convenience.

4.6.1 Constant Centroid

For the constant centroid type of profiles the centroid radius remains constant, so by adding or subtracting the diameter or height previously determined from the centroid radius value the wet surface radius of the volute profile is obtained. For the outer radius it is added while for the inner radius it is subtracted.

4.6.2 Constant Inner Diameter

For these types, the Inner radius remains unchanged and this is considered the inner radii. By adding the diameter or the height previously obtained to the inner radius the outer radius is determined.

This is done for each degree of volute passage and these radii are then plotted in a manner similar to the inner radius and centroid radius. The drawscale is used for scaling and lines are plotted from one X and Y coordinate to the next and a volute profile is generated. Also it may be noted that the diffuser is assumed to be tangential and to avoid overlap with the 1 - 90 degrees of volute profile an "if " statement is used. This logical operator checks to see if the profile at various angles overlap the diffuser. All Y coordinates higher than the inner radii coordinate at the throat are not plotted.

4.7 Plotting the Diffuser

The diffuser is assumed to be tangential and therefore the inner surface of the diffuser is tangential to the inner radius at the throat. Two variables (X_diff_A, Y_diff_A) are determined. Y_diff_A is dependent on the ratio of the diffuser outlet area to the throat area. The square root of this ratio is multiplied to the height of the throat and this value is then added to the outer wet surface Y coordinate at the throat.

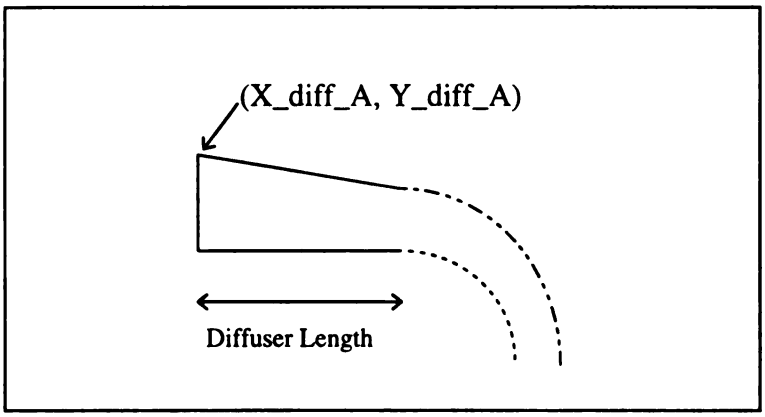

Figure 4.1 Diffuser Construction

The X_diff_A coordinate is dependent on the diffuser length and is obtained by adding that length the X coordinate of the inner radii at the throat. Using these coordinates the diffuser is then plotted as the rest of the coordinates are known.

4.8 Plotting the straight line surfaces (Cross-section)

This part of the code deals with the cross-section generation at different angles of cross-section. For this a new set of center coordinates (center_x_cross, center_y_cross) are defined and these are used to position the cross-section plots. The straight lines are as described in the figure 4.2 and these are used to define the passage leading into the volute. These lines are broken down into two parts and are referred to as "Line" and "Inner Line". Also these plot make use of the Edge width and Gap width factor (diffuser width, b5) which are as detailed in the figure.

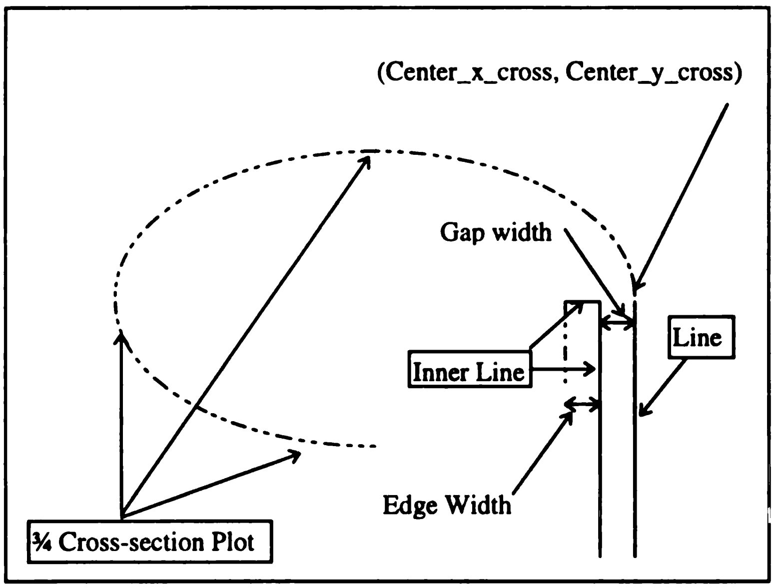

Figure 4.2 Straight Lines and ¾ Cross-section Plot

4.9 Plotting three-fourths of Cross-section View

In this part of the code only three-fourths of the cross-section is plotted. The dashed curve in figure 4.2 is similar to what is carried out. As cross-section at 30 degree intervals are needed a loop is generated from 1 to 360 with a step of 30. The cross-section parameters (diameter, minor and major axis, height and width) at these intervals were calculated earlier during the "Dimension and Area Check" loop. Using these values the cross-section is then plotted. A cylindrical coordinate method is used to plot the circular and elliptical cross-sections while in the rectangular cross-section it is plotted using straight lines. It may also be noted that for the rectangular cross-sections to avoid the display of sharp edges, radius of curvatures were used. These are a set of curves of a constant radii plotted at the edges.

4.10 Plotting Radius R3 and Rest of Cross-sectional View

The method of plotting R3 differ between Constant Centroid and Constant Inner Diameter. Hence it is better to describe them separately.

4.10.1 Constant Centroid

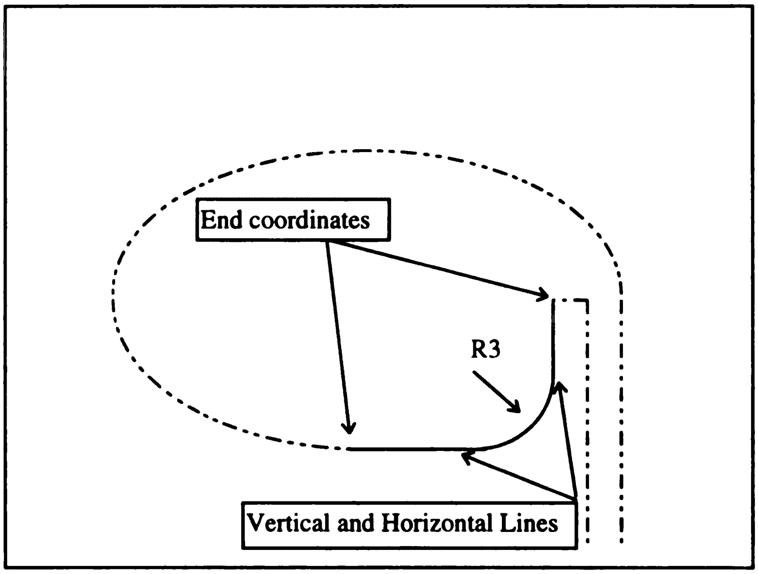

Figure 4.3 Radius R3 for Constant Centroid

This method of plotting is common to all cross-sections of the constant centroid type. Using the end coordinates from the previous plots (Edge plot and three-fourth cross-section plot) a vertical and horizontal line is drawn taking into consideration that these lines will then be tangential to the curve of radius R3. The radius R3 is specified in the data form and this value is then used to connect these lines with smooth curve.

4.10.2 Constant Inner Diameter

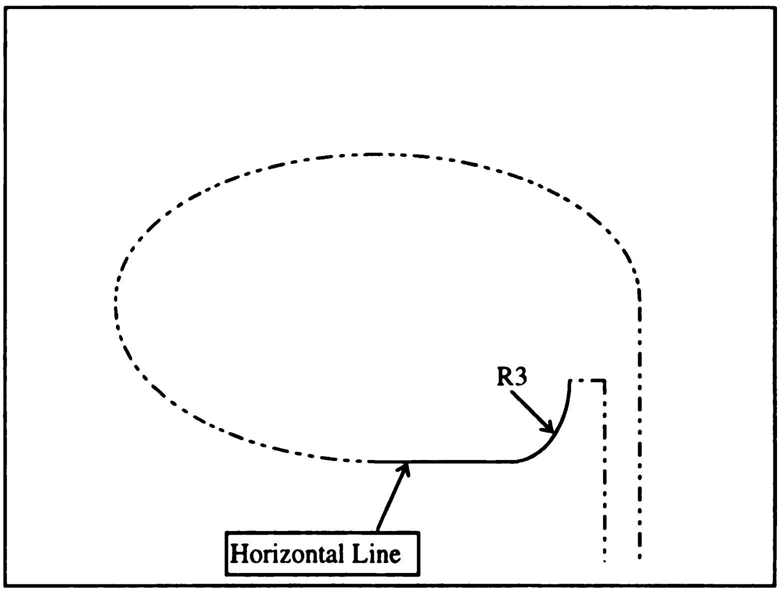

Figure 4.4 Radius R3 for Constant Inner Diameter

As compared to the Constant Centroid this is a little different. As the height of the edge is governed by the value of radius R3. Only a horizontal line is plotted as shown in figure 4.4 from the end coordinates of the previous three-fourth cross-section plot. The length of the line is fixed by a tangential intersection with the curve of radius R3. The curve itself is then plotted after that.

CHAPTER 5

DESCRIPTION OF TURBINE VOLUTE AND CROSS-SECTION GENERATION MODULE

This chapter describes the code used to generate all data needed to construct the cross-section of the Open Flow and Twin Flow turbine. Again it was found convenient to group these as subroutines and house them under a single module. The turbine subroutines differ from the compressor module subroutines in the way that it uses procedures to calculate the area and also to plot the cross-sections. Other motivation for this type of construction was so that at a future stage it could be interfaced with other visual basic applications.

5.1 Turbine Module - (ModTurbine.Bas)

This module contains a number of procedures and are as listed,

1) Open Flow Area - (Area)
2) Twin Flow Area - (Area_twin_flow)
3) Open Flow Plot - (plot_Turbine_profile)
4) Twin Flow Plot - (plot_twin_flow_turbine)
5) Open Flow Main - (SubOpen_Flow)
6) Twin Flow Main - (SubTwin_flow)

5.2 Open Flow Area - (Area)

This procedure is used to calculate the area at different cross-sections of the open flow turbine. The inputs to these procedures are passed with the call command in the open flow main procedure. The method used to achieve this is as described in chapter 2. The area is subdivided into 3 parts, then each is individually determined and then added up. After the area is calculated the radius of centroid for that particular cross-section is also determined. This is also described in detail in chapter 2. This data is then passed back to the main procedure for data recording and further generations.

5.3 Twin Flow Area - (Area_twin_flow)

This procedure works in the same way as the open area calculation procedure. The method for area calculation are detailed in chapter 2. The only difference being the divider wall. All input parameters are passed on by the call command, the area and the radius of centroid is generated using these parameters. This is then passed back to the main procedure for further generation.

5.4 Open Flow Plot - (plot_Turbine_profile)

This procedure is used solely for plotting the turbine cross-section using the given input parameters. This procedure is called from the main subroutine and is used to plot the cross-section at 30 degree intervals. As the cross-section is assumed symmetrical for this part of the project (Bezier curves would be able to handle curves that are not symmetric once the initial plot was obtained.), it was found easier to plot the left half of the cross-section first and then mirror the right half. For the radius R1 to be plotted, the angle through which it spans (α) is needed. This is obtained by using simple geometry as described in the figure 5.1. The value of θ is assumed and this is to ensure that radii R1 and R2 meet tangentially. The angle "θ" can be varied in the code for special cases. (It was also found that variations in θ did not change the plot or area calculation drastically.) Once "α" is calculated the center coordinates are assumed so as to position the plot on the sheet and then the radius R1 is plotted through an angle "α".

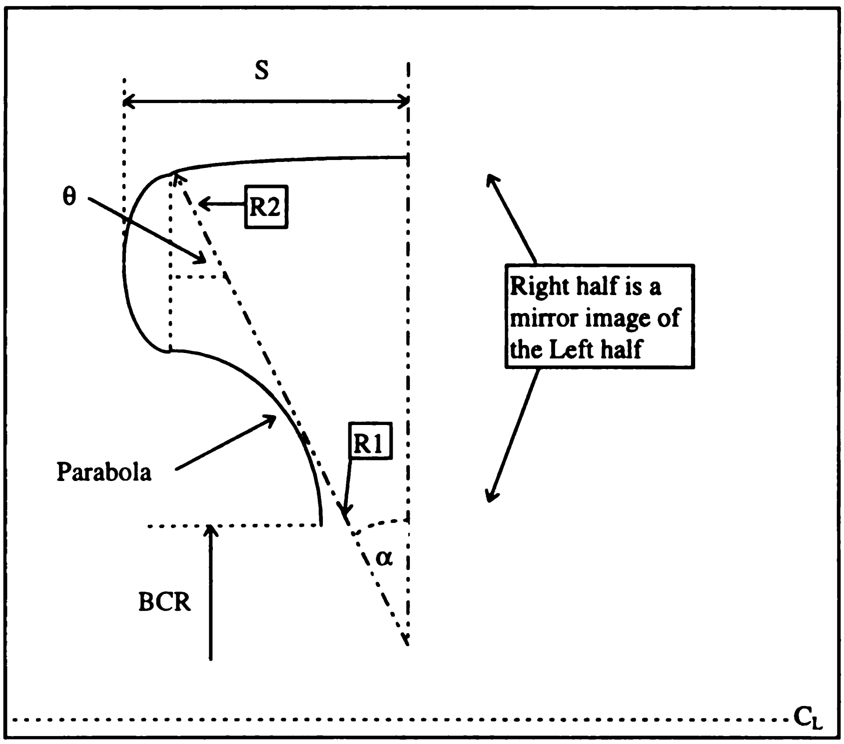

Figure 5.1 Profile Generation (Open Flow)

After Radius R1 is plotted the last point is used to start of the plot of radius R2. Using the input parameters (R1, R2 and S), the center of radius R2 is determined using simple geometry. Using these coordinates radius R2 is plotted over a span of 2θ. Now using the end point of the radius R2 plot and base circle radius (BCR) the dimensions of the parabola are known. Using the start and end coordinates of the parabola, the parabola equation is determined. With this equation the parabola is then plotted. This completes the left hand side of the plot. Using all the X coordinates used to plot the profile a mirror set of X coordinates are generated about the central axis. Using these new set of mirror coordinates the right hand side of the profile is plotted.

5.5 Twin Flow Plot - (plot_twin_flow_turbine)

This plot is similar to the open flow plot described earlier, except now in addition to the open flow profile, the divider wall is introduced.

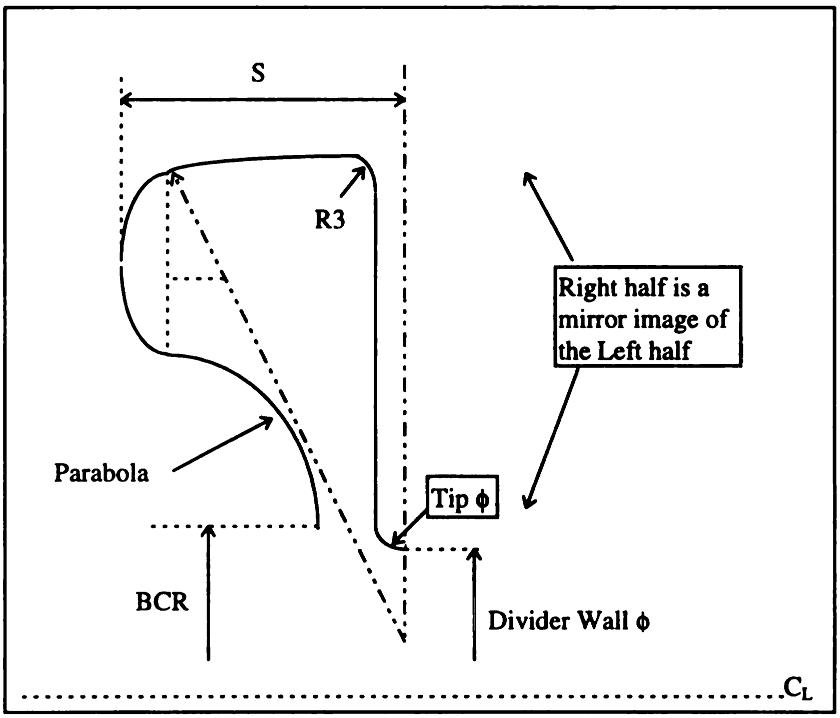

Figure 5.2 Profile Generation (Twin Flow)

In this procedure the divider wall is plotted first. The divider wall diameter is used to obtain the tip radius of the divider wall. Then using the Tip radius a curve is plotted through 90 degrees. Using the end point coordinates of this curve the divider wall is constructed. The length is obtained by using simple geometry. From the end point of the divider wall radius R3 is then plotted through 90 degrees. Then with these endpoints as the start points for radius R1, the rest of the plot is generated similar to the open flow plot described earlier. As the initial overlapping part of radius R1 is not needed to be plotted an 'if' statement is used to check it until the generation reaches the end point of the radius R3 plot.

5.6 Open Flow Main - (SubOpen_Flow)

This procedure is the main manager for the open flow turbine profile plot. It regularly calls the area and plot subroutines to do the major part of the calculation as all these are repetitive. As the evaluation of the open flow parameters give only the area as input is not as easy as the compressor evaluations, an excel spread sheet was constructed so as to study the pattern of variation in the parameters.

This excel sheet is attached in the appendix C. The first part of this spread sheet calculates the area and the radius of centroid for a given set of parameters. This method is as described in chapter 2. The second part of the excel sheet is a detailed examination of all needed parameters with respect to area and centroid radius variation. Data was used from the existing drawing provided by Schwitzer and using this data (R1, R2, and S) the area and radius of centroid was determined. This was done for different types of existing turbine design at every 30 degree of cross-section. A check was carried out on the variation of the ratios of different parameters and it was found that the ratios of (R1/Rc), (R2/R1) and (R2/S) varied least. These ratio were then calculated on the same excel sheet. This was done for each 30 degree interval of cross-section and the average of these for each interval was also determined. These averages are as listed in the table below. The average (Rc/Rc-1) is the ratio of the centroid radius of the current cross-section to the one before. Computing these values it was noted that for some intervals there was a linear variance. From the throat to 210 degree of cross-section all of the above ratio could be obtained using a linear equation. This maybe noted in the code and a loop is used for sequential generation of the plots. Further to 210 degrees of cross-section a linear variation of all ratios simultaneously was found to be a problem and to simplify the procedure loops were constructed for every two 30 degree intervals. These intervals are grouped together in the table below with a different background color. This process was repeated using the ratios below and the profiles for the various cross-sections were plotted

	R1/Rc	R2/R1	R2/S	Rc/Rc-1
Throat	1.230	0.125	0.195	1.020
330	1.220	0.120	0.190	1.020
300	1.210	0.115	0.185	1.020
270	1.200	0.110	0.180	1.020
240	1.190	1.105	0.175	1.020
210	1.180	1.100	0.170	1.020
180	1.160	0.090	0.165	1.025
150	1.145	0.080	0.155	1.030
120	1.130	0.075	0.155	1.035
90	1.115	0.060	0.150	1.040
60	1.100	0.055	0.142	1.045
30	1.050	0.045	0.135	1.050

Table 5.1 Distribution of Parameter Ratios (Open Flow)

Using the already recorded parameter values at various intervals of cross-section the intermediate points of the radius R1 and Rc were evaluated by interpolating between each interval. In this manner a value for the above parameters for each degree of cross-section was obtained. Using these values the volute profile for the open flow turbine was generated and plotted using the cylindrical coordinate method from 0 - 360 degrees. The diffuser was then drawn out as well and is done so as described earlier for the compressor module.

5.7 Twin Flow Main - (SubTwin_flow)

This method is similar to the working of the open flow procedure, except in this case an additional parameter is involved. Again this procedure regularly calls the subroutines for area and plot generation as these task are repetitive. Since the number of parameters involved are large, the determination of these parameters from only the area as an input is not simple. An excel sheet was used again to examine the pattern in which these parameters were varying. This excel sheet is attached in the appendix.

From the excel sheet it can be seen that first a calculation of area and centroid radius was done as described in chapter 2. This was then used to generate the second part of the excel sheet in which (like the one used for open flow) various ratios were calculated. It was found that in addition to ratios (R1/Rc), (R2/R1) and (R2/S) , the radius R3 also varied on a pattern of its own. These are described as in the table 4.2. Examining these ratios it is observed that the ratios vary linearly for throat - 270, 240 - 180, 150 - 90 and 60 - 30 degree intervals of cross-section. Using these linearly varying subgroups, loops were constructed in the code so as to generate the parameters. These subgroups are denoted by a common background color in the table. Using loops for the above intervals, linear equations to generate the required ratios were constructed within these loops. After respective parameters were determined, the area using these determined parameters is calculated using the "area" subroutine and the actual radius of centroid is found and redefined.

	R1/Rc	R2/R1	R2/S	R3	Rc/Rc-1
Throat	1.230	0.120	0.140	0.22	1.010
330	1.220	0.115	0.135	0.21	1.010
300	1.210	0.110	0.130	0.19	1.010
270	1.200	0.105	0.125	0.18	1.010
240	1.190	0.095	0.122	0.17	1.020
210	1.180	0.090	0.119	0.15	1.020
180	1.170	0.085	0.116	0.14	1.020
150	1.150	0.080	0.130	0.13	1.025
120	1.135	0.070	0.120	0.11	1.030
90	1.110	0.060	0.110	0.10	1.035
60	1.095	0.055	0.098	0.09	1.040
30	1.065	0.047	0.097	0.07	1.055

Table 5.2 Distribution of Parameter Ratios (Twin Flow)

As in the case of open flow the already recorded parameter values at various intervals of cross-section the intermediate points of the radius R1 and Rc were evaluated by interpolating between each interval. A value for each degree from 0 - 360 was then obtained. Using these values of R1, Rc and BCR a cylindrical plot was carried out and the volute profile for the twin flow turbine was generated.

5.8 Area Check Calculations

For every 30 degree interval it may be noted that the "call" command was used. The main function of this is to calculate the area and the centroid radius using the parameters generated by the ratios. The Area subroutines returns these values and a check is carried out to see whether these parameters generated were in accordance with the requirements. If the tolerances were not met then a minute increase/decrease is made in radius R1 and all parameters are redefined. This is carried on till the tolerances are met. Also it maybe noted that although the radius of centroid is determined initially by using the ratio values from the spread sheet. Using this generated parameters, the actual radius of centroid is determined by the Area Subroutine and this value is recorded. This ensures that the correct radius of centroid is used which then governs all other parameters.

CHAPTER 6

USE OF BEZIER CURVES FOR SURFACE GEOMETRY GENERATION

6.1 Introduction

This thesis being the first part of the project a few details of future work to be done is detailed in this chapter. The earlier chapters described the method in which the geometric parameters are generated, now using these parameters Bezier curves are to be constructed so as to enhance surface generation. In this chapter a review of a few basic ideas of analytical geometry, introduction to the Bezier polynomial technique and some fundamental properties of Bezier polynomials are presented. The use of these polynomials for flexible volute profile and turbine description can be used to generate simple curves that form part of a volute or turbine cross section along with other elementary geometry's like line, ellipse, parabola and arc.

6.2 Explicit, Implicit and Parametric Functions

In this section a brief summary will be given of some of the classical geometry description concepts which are important for an appreciation of the capabilities of Bezier functions. A more detailed description is given in [1] which is a good introduction to computational geometry, particularly Bezier polynomials.

$$y = f(x) \tag{6.1}$$

The simplest way to define a plane curve is to use the explicit form, 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 single valued 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,

$$f(x, y) = 0 \qquad (6.2)$$

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 calculated directly unless the equation can be reduced to an explicit equation for x or y. For example the equation,

$$x^2 + y^2 - r^2 = 0 \qquad (6.3)$$

is the implicit function for the circle. 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 $y = +\sqrt{(r \ \ - x^2)}$ for the upper half and $y = -\sqrt{(r \ \ - x^2)}$ for the lower half.

An alternative way of describing lines and curves which 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 = x(u), y = y(u). For example the circle $x^2 + y^2 - r^2 = 0$ can be expressed parametrically by the equations:

$$x = r * \mathrm{Cos}\,(u)$$

$$y = r * \mathrm{Sin}\,(u) \qquad (6.4)$$

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

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

$$r(u) = \sum_{n=0}^{N} u^n a_n \tag{6.5}$$

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 (2^{nd} and 3^{rd} order) polynomial parametric functions and the physical meaning of their vector coefficients will now be illustrated.

The segments of quadratic parametric curves and surfaces are described by an equation of the form:

$$r(u) = a_0 + a_1 u + a_2 u^2 \tag{6.6}$$

It can be seen that the three vectors a_0, a_1, and a_2 are required to define the segment of a quadratic curve. The 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 $0 < u < 1$ 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^2r/du^2 at the beginning of the segment. Thus,

$$\begin{aligned} a_0 &= r(0) \\ a_0 + a_1 + a_2 &= r(1) \\ a_1 &= \frac{dr}{du}(0) \end{aligned} \tag{6.7}$$

Solving for a_0, a_1, and a_2 we get,

$$\begin{aligned} a_0 &= r(0) \\ a_1 &= \frac{dr}{du}(0) \\ a_2 &= r(1) - r(0) - \frac{dr}{du}(0) \end{aligned} \tag{6.8}$$

By direct substitution in equation (6.7), we can obtain r in terms of r(0), r(1) and dr/du(0). Thus,

$$r = r(u) = r(0)\left(1-u^2\right) + r(1)u^2 + \frac{dr}{du}(0)\left(u-u^2\right) \tag{6.9}$$

Alternately we may write r = U . C . S where, U, C and S denotes the product of the three matrices given below:

$$r(u) = \begin{bmatrix} 1 & u & u^2 \end{bmatrix} * \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & -1 \end{vmatrix} \begin{vmatrix} r(0) \\ r(1) \\ \frac{dr}{du}(0) \end{vmatrix} \tag{6.10}$$

Cubic parametric equations for the definition of curves and surfaces in aircraft design, can be described by the equations of the form:

$$r = r(u) = a_0 + a_1 u + a_2 u^2 + a_3 u^3 \tag{6.11}$$

Following a similar procedure as done above for the parametric quadratic equation we can write equation (6.11) in terms of the boundary conditions r(0), r(1), dr/du(0) and dr/du(1) as follows:

$$r(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} * \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{vmatrix} \begin{vmatrix} r(0) \\ r(1) \\ \frac{dr}{du}(0) \\ \frac{dr}{du}(0) \end{vmatrix} \tag{6.12}$$

6.3 Bezier Curves

The vector coefficients of the parametric curves described by the equation (6.5) 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 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 curves due to the many cross couplings as can be seen from the non-diagonal elements in the coefficient matrix in equation (6.12).

Bezier [1] has recombined the terms of the polynomial parameterisation 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 Beziers form we write equation (6.6) as follows:

$$r = r(u) = (1-u)^2 r_0 + 2u(1-u)r_1 + u^2 r_2 \tag{6.13}$$

where again $0 < u < 1$ for any given segment. It can be seen that this is a simple rearrangement of the quadratic polynomial form of equation (6.5) with,

$$a_0 = r_0$$
$$a_1 = 2(r_1 - r_0) \qquad (6.14)$$
$$a_2 = r_0 - 2r_1 + r_2$$

The important consequence of this rearrangement is that:

$$r(0) = r_0$$
$$r(1) = r_2$$
$$dr/du(0) = 2(r_1 - r_0) \qquad (6.15)$$
$$dr/du(1) = 2(r_2 - r_1)$$

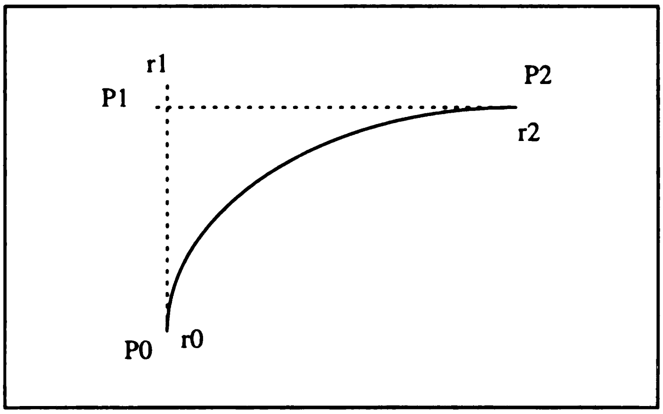

Figure 6.1 Bezier Curve Example

Thus the curve described by Beziers form passes through the points r_0 and r_1 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 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 (6.11) can be written as:

$$r(u) = (1-u)^3 r_0 + 3(1-u)^2 u r_1 + 3(1-u)u^2 r_2 + u^3 r_3 \qquad (6.16)$$

where,

$$
\begin{aligned}
a_0 &= r_o \\
a_1 &= 3(r_1 - r_0) \\
a_2 &= 3(r_2 - 2r_1 + r_0) \\
a_3 &= r_3 - 3r_2 + 3r_1 - r_0
\end{aligned}
\tag{6.17}
$$

Thus the description of an n^{th} order Bezier polynomial equation is:

$$
r(u) = \sum_{n=0}^{N} \frac{N!}{(N-n)!n!} u^n (1-u)^{N-n} r_n \tag{6.18}
$$

6.4 Changing The Order Of The Bezier Polynomial

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.

$$
r^*\left(\frac{n}{N+1}\right) = \frac{1}{(N+1)}\left[ir^*\left(\frac{n-1}{N}\right) + (N+1-n)r^*\left(\frac{n}{N}\right)\right] \tag{6.19}
$$

for $0 < n < N+1$

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

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 Future Interfacing for varying Area Distribution

In this project a linear A/R distribution was considered. This was set up simply as described in chapter 4. This may be modified in the future so as to link itself to a varying Area distribution. This area distribution will be plotted on a chart using bezier curves so if a change is needed to be made it can be done by clicking and dragging the bezier nodes. This makes the area distribution very flexible and as this is an input to the profile and cross-section generation, it will suitably alter the drawings.

7.2 Area Check Calculation.

7.2.1 Compressor

It may be noted that the area check calculation is used for all compressor drawings so as to finalize the exact area at each cross-section and also evaluate the parameters for these cross-sections. [As discussed in Section 3.3 "Dimension and Area Check Calculation".] It was observed that the difference in the actual and assumed area was small and therefore fewer iterations [as detailed in Section 3.3] were needed to fit the tolerances and evaluate the parameters. The output displayed on the "Parameter Form" was checked against the tables existing on the Schwitzer drawings and found to match quite accurately. The reason the word "quite" is used is because the A/R profile was considered to be linear in this project while is not the actual case on the existing drawings.

7.2 Turbine

It may be noted that in the case of the turbine generation codes an "Area Check Calculation Loop" was used in the initial loops but not later on. The reason here being that a linear A/R distribution was used for the Area values and if we refer to the chart in appendix C of the actual A/R distribution of the existing turbine drawings we see that it is not linear. It was found that running the "Area Check Calculation Loops" on all the generations took a long time. This is blamed on the reason that the initial guess on the parameters were way off from the linear A/R distribution (i.e. the formula described in equation 4.1) that they were being checked with. This caused the time delay in meeting the tolerance limits and hence the plot. Therefore to reduce the run time until a more realistic A/R distribution is constructed so as to check the tolerances against.

7.3 Bezier Curves

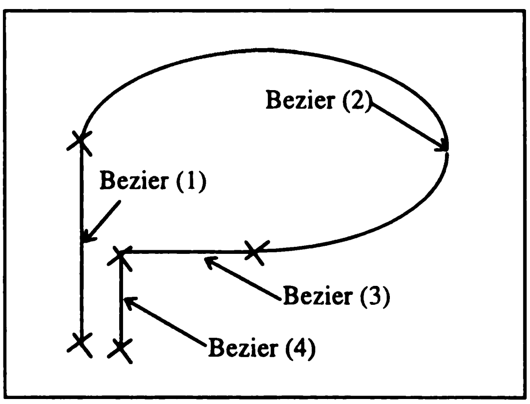

Figure 7.1 Cross-section made up of Bezier Curves

As this thesis was the first part of the main project, it dealt with the initial methods to generate various cross-sections. These can be referred to as node points. Using these node point an attempt can be made to plot the same profile using bezier curves. For example the figure shows a simple cross-section which can be subdivided into a number of different bezier curves. These curves are referred to as Bezier(1), Bezier(2), Bezier(3) and Bezier(4). As the node points are available, we can now use this data to plot it. Bezier (1) would be a straight line and would need only the start and end points. Bezier(2) is a curve of some dimension and would need a number of node points for accurate definition. Bezier(2) would share it start node point with the end node point of Bezier(1). Again Bezier(3) and Bezier(4) are straight lines which would be generated by knowing the start and end points of each. Although this might look simple, it is not as the cross-section can get complicated and methods have to devised on obtaining respective node points.

The main advantage of using bezier curves in cross-sections such as the figure 7.1, is that these node points can be changed to alter the shape and hence the area and parameters of the cross-section. An example for which this may be used is to fit housings into engines with limited space by modifying certain sections.

7.4 Parameter Form

A detailed description of the "Parameter Form" is done in Chapter 3. The main function of this is to display data that was generated by the code in a format similar to the existing tables available on the drawings. It was also designed to be the interface for the

Bezier curve application and all needed input data for that application may be generated and displayed in a similar format. Also if data was changed on the grid it might be possible to automatically update the bezier curve generations. This would make it manipulative. Also it may be possible to update the data in the grid if the bezier curves were altered by the 'click and drag' method. A reverse calculation may be possible so as to provide all parameters such as area, radii etc.

7.5 Approximations

All assumed data was included on the data forms and may be changed as needed. Some more generations that were assumed and may be changed later are,

7.5.1 Centering

A manual method was provided to center the drawings on the form. This was kept simple so as to take advantage of the existing scaling technique used by Schwitzer in their Visual Basic application. The center coordinates if need to be changed have to done by changing the data in the code. This data is well commented on in the code.

7.5.2 Diffuser

The diffuser was constructed by using simple techniques as a lot of data on diffuser construction was not available and is not a priority on this project. The generation of the diffuser may be reconstructed later on if needed and so it is kept simple.

7.6 Recommendations

It maybe recommended that for generating the bezier curves it may be more advantageous to have a single procedure to define the cross-sections and generate the curve based on some input parameters like radius, width etc. Then for each of these cross-sectional types the number of bezier nodes could be arbitrary. The bezier formulation could be written by a summation procedure. Once the number of curves are fixed, a separate code could be used to define each curve that makes up the whole cross-section. This might have an advantage and disadvantage, the advantage being that a construction of this type would give the user a powerful grip on the flexibility of the cross-sectional shape. The disadvantage would be that any future unseen shapes that need more than the number of curves available could cause a problem.

LIST OF REFERENCES

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

2. Eckert B and Schnell E. : Axial - und Radialkompressoren, 2nd ed. Springer Verlag, 1980

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

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

5. Japikse D.: advanced diffusion levels in turbocharger compressor and component matching. IME Conference on Turbochargers and Turbocharging, Paper C45/82, 1982

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

7. Iversen H., Rolling G.R. and Carson J.: 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

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

9. 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

10. Gurewich and Gurewich : Visual Basic 4.0, 3rd Edition.

11. Rothbart : Mechanical Design and Systems Handbook.

12. David F. Roger and J. Alan Adams : Mathematical Formulation for Computer Graphics, 2nd Edition.