Titre: Title:	Geometrical modeller and mesh generation turbomachinery
Auteur: Author:	Yi Li
Date:	1995
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Li, Y. (1995). Geometrical modeller and mesh generation turbomachinery [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/31979/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/31979/
Directeurs de recherche: Advisors:	Ricardo Camarero
Programme: Program:	Non spécifié

UNIVERSITÉ DE MONTRÉAL

GEOMETRICAL MODELLER AND MESH GENERATION FOR
TURBOMACHINERY

Yi Li

DÉPARTEMENT DE GÉNIE MÉCANIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services Branch

Direction des acquisitions et
des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

ISBN   0-612-03666-9

Canadä

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

GEOMETRICAL MODELLER AND MESH GENERATION FOR
TURBOMACHINERY COMPONENTS

présenté par: Li Yi

en vue de l'obtention du diplôme de: MAÎTRISE ÈS SCIENCES APPLIQUÉES

a été dûment accepté par le jury d'examen constitué de:

M. GRANGER Louis, M.Sc.A., président

M. CAMARERO Ricardo, Ph.D., membre et directeur de recherche

M. PELLETIER Dominique, Ph.D., membre

Acknowledgement

I would like to express my gratitude to Professor Ricardo Camarero from the department of Mechanical Engineering and to Mr. Robert Magnan from IREQ, for their guidance, encouragement, and continuous support from the inception to the present form of this thesis.

I would also like to acknowledge the help I have received from my friends at Ecole Polytechnique, and CERCA.

In addition, I acknowledge the support of the Ecole Polytechnique for the exemption of the foreign student tuition fees and the financial support of NSERC and IREQ.

Résumé

Ce mémoire concerne la modélisation géométrique et génération de maillage pour des composantes de turbines hydrauliques. La méthodologie de design de turbines nécessite une représentation géométrique de chaque composantes, c'est-à-dire la roue, le diffuseur et la bâche spirale. Ceci sert à la génération de maillage pour la discrétisation du domaine de calcul pour les calculs de l'écoulement.

Le but de ce travail est de réaliser un logiciel pour faciliter et voir automatiser ces différentes étapes. On décrit diverses approches pour réaliser un tel environnement et on les évalue à la lumière des besoins particuliers du domaine hydraulique, et de la nature des données qui seront fournies pour ces calculs.

Le problème est posé dans le cadre de la conception assistée par ordinateur avec comme première étape la caractérisation géométrique.Parmi les différentes techniques de modélisation, on retient l'approche surfacique.

Les composantes géométriques concernées sont la bâche spirale, la roue et la diffuseur. Ces trois géométries peuvent être représentées par le déplacement d'une courbe (appelée profil ou section) le long d'un autre courbe (appelée dorsale). Au cours du déplacement, le profil change de forme selon une loi prescrite. Ces différentes courbes sont représentées par des NURBS et les surfaces engendrées sont représentées par des splines bi-cubiques.

Une bibliothèque de fonctions a donc été créée pour la représentation de courbes et surfaces basée sur des représentation en B-Spline et en NURBS dont:

- le calcul des dérivées,

- la longueur d'arc,

- l'intersection de deux courbes.

La roue est constituée de plusieurs pales disposées de façon axiale autour de l'axe de rotation et est bornée par deux surfaces de révolution, le moyeu et le carter. Les données disponibles pour définir la roue sont deux series de points représentant le moyeu et le carter ainsi que deux treillis de points définissant les deux cotés d'une pale. Le volume à mailler est construit avec ces surfaces et en complétant avec des surfaces de périodicités issues du bord d'attaque et du bord de fuite. Le maillage est ensuite engendré par interpolation transfinie sur chaque surface de révolution. En répétant du moyeu au carter, on obtient un maillage 3D.

Le modèle géométrique du diffuseur est basé sur un produit tensoriel où une surface est engendrée par le déplacement d'une section en profil le long d'une courbe. Au cours de ce déplacement, les caractéristiques de la section varient, et leur spécification est faite selon un langage de représentation géométrique. Le maillage est du type multi-bloc avec une subdivision de la section.

La géométrie de la bâche spirale est une surface engendrée par le déplacement d'une section dont la forme varie. La partie la plus délicate à représenter est le raccord entre la fin de la bâche sur elle-même près du canal d'amenée. Les coupes radiales sont remplacées par des surfaces gauches variant de façon monotone de la dernière section plane à la surface du bec. Le maillage est obtenu par un découpage de la section en plusieurs blocs à l'intérieur desquels un maillage transfini est généré.

Les diverses représentations géométriques pour les composantes de turbines ont été réalisées à partir d'une librairie de modélisation géométrique, et d'un langage pour la description de ces surfaces. Cet ensemble permet la caractérisation de ces surfaces à l'aide de surfaces bi-cubiques. Ensuite, les maillages sont engendrés par blocs en utilisant la technique des interpolations transfinies. Ces outils ont été appliqués avec succès à des géométries réelles issues de composantes de l'industrie.

Abstract

The design methodology for turbomachinery requires a suitable geometric modeller to represent the three major components, runner, diffuser, and spiral casing, and a mesh generation capability to discrete domain. In this thesis, geometric modelling and automatic mesh generation methods are presented for hydraulic turbomachinery components. The purpose of this method is to automate geometry description and grid generation of the three major components of a turbomachinery.

A surface geometric modelling system which uses Non-Uniform-Rational-BSplines (NURBS) representation was created to accurately represent the complex geometric shapes of all components. It is able to model a spiral casing described by a series of sections and a diffuser with or without a pillar at the outlet. It also has the ability of handling the complex three-dimensional shape of blades, and the intersection between the blade and the hub or the shroud. A tensor product format makes it easy to create a Curve/Surface geometry library in one single data structure. This also makes it very efficient to construct the curves and the surfaces of components using a blending method.

To generate a good quality 3D curvilinear mesh, a transfinite interpolation method was developed. It has the ability to generate three dimensional meshes with an appropriate concentration and smoothness between the blades, inside the

diffuser and in the spiral casing.

A software package based on this method with a graphical user interface is also presented. The package has the ability to:

1. Create and modify the 3D geometric model for each components.

2. Construct surfaces delimiting the spiral casing, blade-to-blade channel of the runner and the diffuser boundaries.

3. Calculate a body-fitted mesh of all the components of the turbomachinary.

This software package works for Kaplan, Francis and Bulb type turbines as well for pumps.

Condensé en français

Modélisation géométrique et génération de maillage pour des composantes de turbines hydrauliques.

Résumé

La méthodologie de design de turbines nécessite une représentation géométrique de chaque composantes, c'est-à-dire la roue, le diffuseur et la bâche spirale. Ceci sert à la génération de maillage pour la discrétisation du domaine de calcul pour les calculs de l'écoulement.

Le but de ce travail est de réaliser un logiciel pour faciliter et voir automatiser ces différentes étapes. On décrit diverses approches pour réaliser un tel environnement et on les évalue à la lumière des besoins particuliers du domaine hydraulique, et de la nature des données qui seront fournies pour ces calculs.

Introduction

Le problème est posé dans le cadre de la conception assistée par ordinateur avec comme première étape la caractérisation géométrique. La possibilité d'utiliser un modeleur géométrique commercial n'est pas retenue en faveur d'un logiciel spécialisé aux turbines pour faciliter son utilisation. Parmi les différentes techniques de modélisation, on retient l'approche surfacique pour éviter les ambiguïtés des modeleurs en fils de fer sans toutefois impliquer les difficultés conceptuelles des mod-

eleurs solides. Il est montré que ce niveau de fonctionnalité n'est pas nécessaire à l'application envisagée.
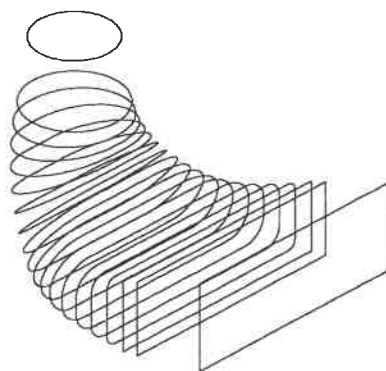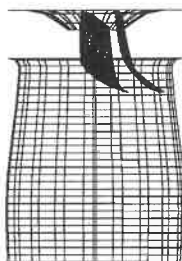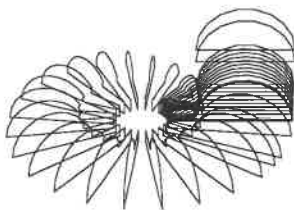

Figure 0.1: Trois composantes

Les composantes géométriques concernées sont la bâche spirale, la roue et la diffuseur qui sont montrés à la figure 0.1. Ces trois géométries peuvent être représentées par un même modèle conceptuel. C'est-à-dire celui d'une surface engendrée par le déplacement d'une courbe (appelée profil ou section) le long d'un autre courbe (appelée dorsale). Au cours du déplacement, le profil change de forme selon une loi prescrite.

Ces différentes courbes sont représentées par des NURBS et les surfaces engendrées, le sont par des splines bi-cubiques. Ces choix ont été dégagés essentiellement afin d'éviter des ambiguïtés, et permettre une caractérisation simple à l'aide de quelques paramètres globaux.

Génération de maillages

Le choix dans les techniques de maillages se situe au niveau des maillages dits structurés ou non-structurés. Les critères sont d'ordre de la complexité des algorithmes et de la fonctionnalité vis-à-vis la complexité géométrique. Dans la première famille, on retrouve les techniques de maillage conformes qui sont facilement éliminées car elles ne permettent pas des extensions en 3D. Les mailleurs issus de la résolution d'équations différentielles elliptiques donnent généralement des maillages de bonne qualité. En plus ils se prêtent bien à une automatisation qui découlent des propriétés des équations à résoudre. En effet, ces dernières assurent l'unicité des mailles, sans plis et très lisses. Elles sont cependant coûteuses.

Les méthodes algébriques, ou interpolations transfinies, offrent des possibilités aussi intéressantes que les méthodes basées sur les équations elliptiques, et ce à moindre coût. Dans leur forme la plus simple, ces méthodes ne peuvent toutefois pas s'adapter à toutes les topologies rencontrées dans les applications qui nous occupent. Pour palier à ces cas, on a recours à une technique de découpage du domaine par

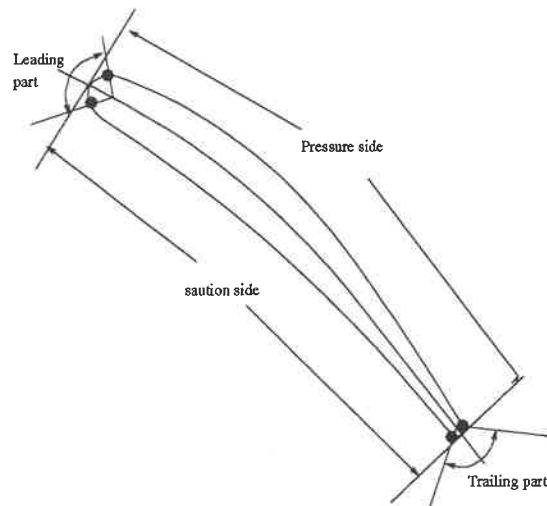bloc et on applique ensuite les méthodes transfinies.
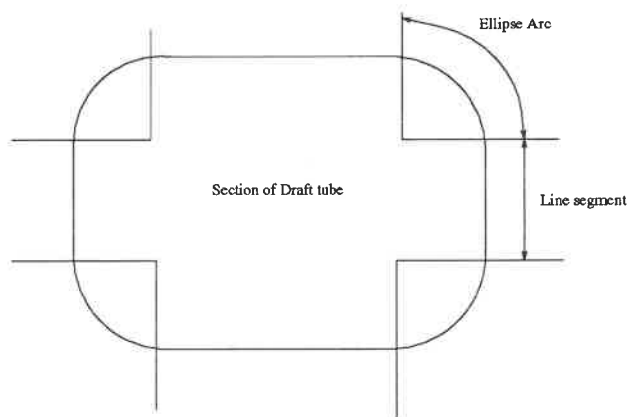

Figure 0.2: Le profil d'une aube
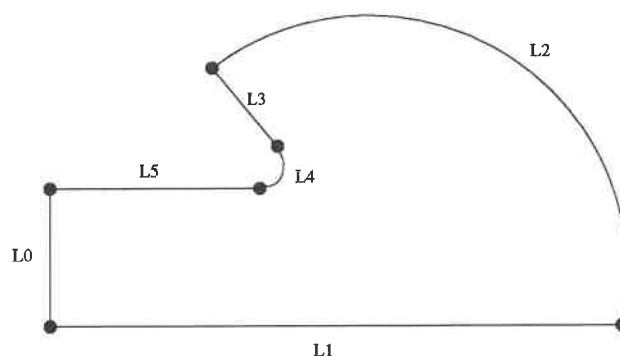

Figure 0.3: La section du diffuseur

Figure 0.4: la section de la bâche spirale

Modélisation géométrique

Pour la réalisation de ce projet, une bibliothèque de fonctions pour la représentation de courbes et surfaces basée sur des représentation en B-Spline et en NURBS a été créée. On retrouve un certain nombre de fonction de base pour la manipulation et la représentation, telles que le calcul des dérivées, de la longueur d'arc, de l'intersection de deux courbes, etc... Ces diverses fonctions sont ensuite appliquées aux composantes de turbines hydrauliques. Cette construction se fait à partir des données telles que disponibles dans notre application avec comme exemple le profil d'une aube donné à la figure 0.2, la section du diffuseur à la figure 0.3 et la bâche spirale à la figure 0.4. Les surfaces correspondantes sont ensuite construites à l'aide de produits tensoriels.

Roue

La roue est constituée de plusieurs pâles disposées de façon axiale autour de l'axe de rotation et est bornée par deux surfaces de révolution, le moyeu et le carter. Le volume à mailler est construit avec ces surfaces et en complétant avec des surfaces de
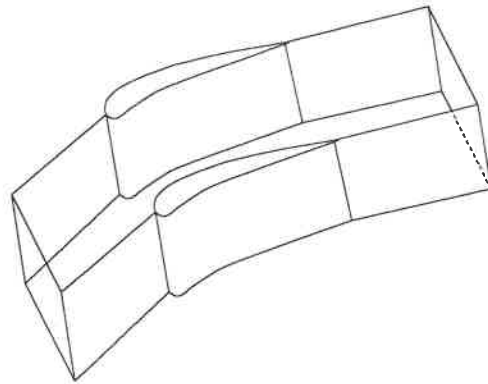
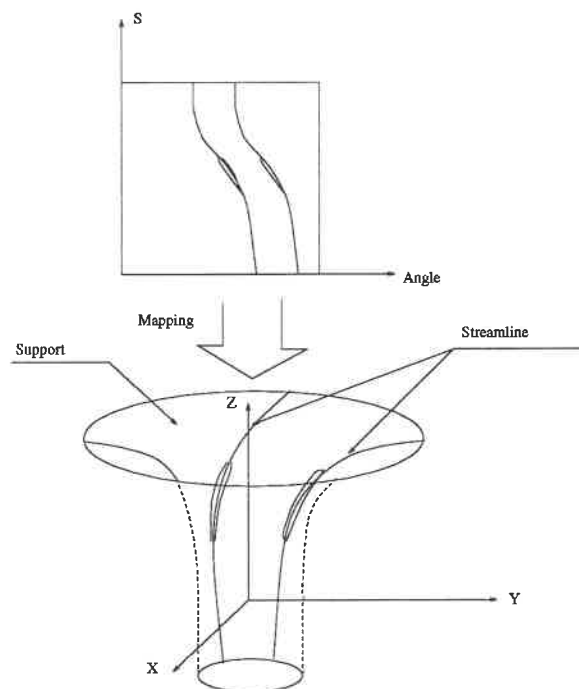Figure 0.5: Le volume à mailler est construit

Figure 0.6: Le construction de volume

périodicités issues du bord d'attaque et du bord de fuite tel qu'illustré à la figure 0.5.
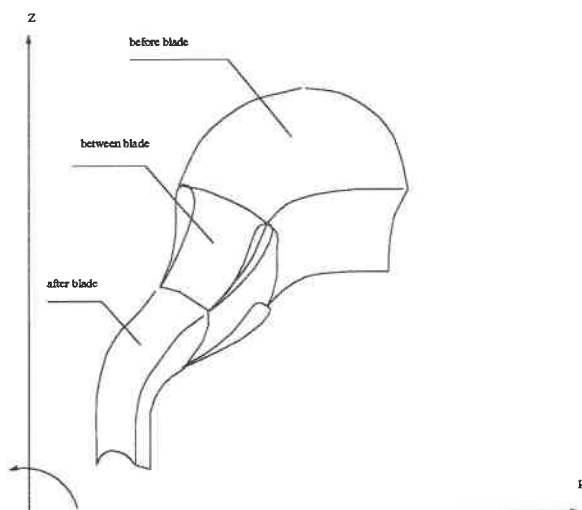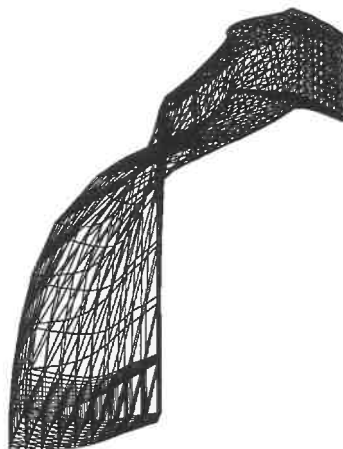
Figure 0.7: Le volume de canal

Figure 0.8: un maillage 3D

Cette construction est montrée à la figure 0.6. Le maillage est ensuite engendré par interpolation transfinie (cf. figure 0.7) sur chaque surface de révolution. En répétant du moyeu au carter, on obtient un maillage 3D (cf. figure 0.8).
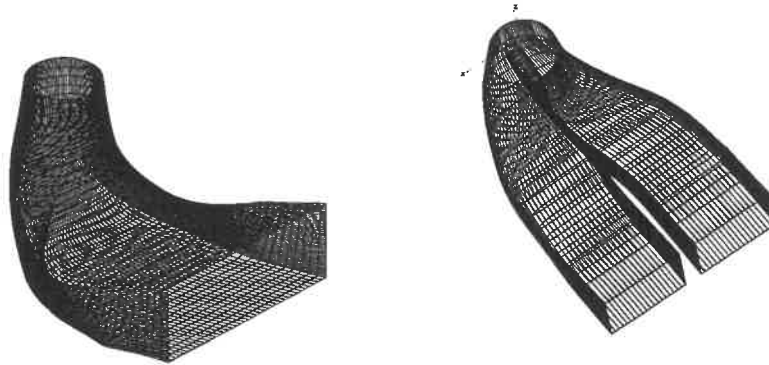
Diffuseur

Figure 0.9: Deux types de diffuseurs

On couvrira deux types de diffuseurs, avec et sans pile. Dans les deux cas, le modèle géométrique est basé sur un produit tensoriel où une surface est engendrée par le déplacement d'une section en profil le long d'une courbe. Au cours de ce déplacement, les caractéristiques de la section varient, et leur spécification est faite selon un langage de représentation géométrique. Les maillages obtenus sont illustrés à la figure 0.9. Ils sont du type multi-bloc avec une subdivision de la section.

Bâche spirale

La géométrie de la bâche spirale est d'un type similaire au diffuseur. C'est une surface engendrée par le déplacement d'une section dont la forme varie comme illustrée à la figure 0.10 et 0.11. La partie la plus délicate à représenter est le raccord entre la fin de la bâche sur elle-même avec le canal d'amenée. Ceci nécessite la redéfinition des surfaces servant de supports tel qu'illustré à la figure 0.12. En effet, les coupes radiales sont remplacées par des surfaces gauches variant de façon
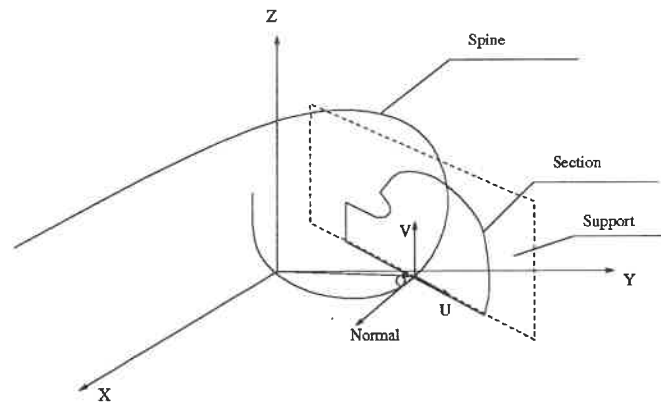
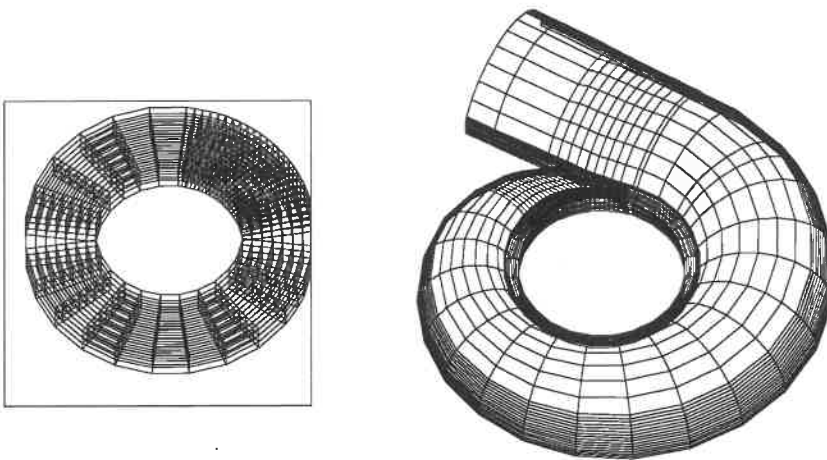Figure 0.10: Le déplacement d'une section

Figure 0.11: Deux partie

monotone de la dernière section plane à la surface du bec. Le maillage est obtenu par un découpage de la section en plusieurs blocs à l'intérieur desquels un maillage transfini est généré.

xix

Origan Section

New Section

Cut Wall

Begin section of join part
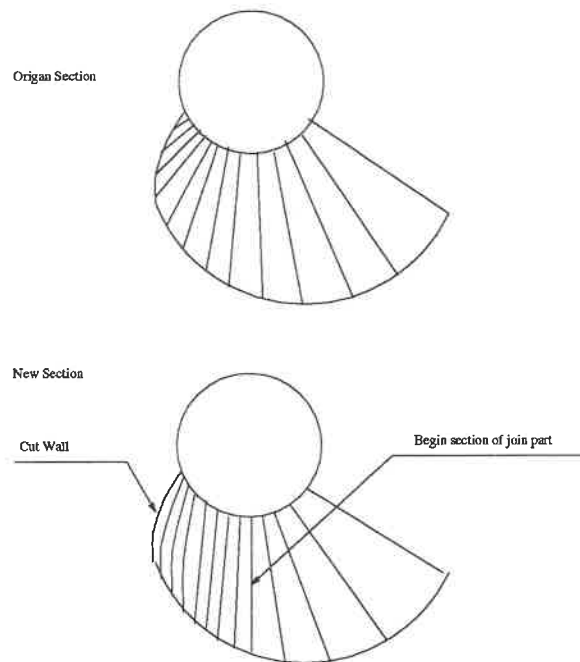
Figure 0.12: La redéfinition des surfaces

Conclusion

Les diverses représentations géométriques pour les composantes de turbines ont été réalisées à partir d'une librairie de modélisation géométrique, et d'un langage pour la description de ces surfaces. Cet ensemble permet la caractérisation de ces surfaces à l'aide de surfaces bi-cubiques. Ensuite, les maillages sont engendrés par blocs en utilisant la technique des interpolations transfinies. Ces outils ont été appliqués avec succès à des géométries réelles issues de composantes de l'industrie.

Contents

List of Figures

List of Tables

List of notations and symbols

T	Homogeneous transformation
U_k	Logical space of dimension k
Ω_n	Physical space of dimension n
X_k^n	Map from logical space U_k to physical space Ω_n
x, y, z	Physical space coordinates
ξ, η, ζ	Logical space coordinates
$\nabla^2 \xi$	Laplace operator with respect to logical space
$\triangle x$	Difference operator with respect to physical space
$X(u, v)$	General function of u, v
$j_{i,j}$	Elements of the metric tensor
$C(u)$	Parametric curve with parameter u
$\tilde{C}(u)$	Homogeneous parametric curve with parameter u
$S(u, v)$	Parametric surface with parameters u, v
$S_u(u, v)$	Derivative of surface with respect to u parameter
H_i^3	Cubic Hermite polynomials
P	Vector in physical space

$\tilde{P}$	Vector in homogeneous space
P_i	Control vector in physical space
$\tilde{P}_i$	Control vector in homogeneous space
$P(u)$	Vector function of parameter in physical space
$B_i^k(u)$	Scalar-valued polynomials of order k in the variable u
$R_{i,j}^{p,q}(u,v)$	Bivariate scalar-valued polynomials of order p, q in the variable u, v
P_i^j	Corner vectors of patch
$\alpha_i, \beta_i, \gamma_i$	Coefficients of tridiagonal system

Chapter 1

Introduction

1.1 Problem Statement

Computer integrated manufacturing(CIM) of turbomachinery calls for the coordination of three areas: hydraulic or aerodynamic design, mechanical design, and automated fabrication with NC machine tools. The hydraulic design phase is subdivided into four steps as shown in Figure 1.1. Our study focuses on the first two steps of this process. First, a suitable geometric representation(geometric modeling) of the major components has to be created. Second, a quality grid that conforms to both the geometric definition and the expected solution gradients has to be generated.

1.2 Geometrical Modeling

Geometric modeling as a distinct discipline appeared in the early 1970s with the rapid development of computer graphics, computer-aided design(CAD) and computer-aided manufacturing(CAM) technologies. It refers to a collection of methods used to define the shape and other geometric characteristics of an object. These
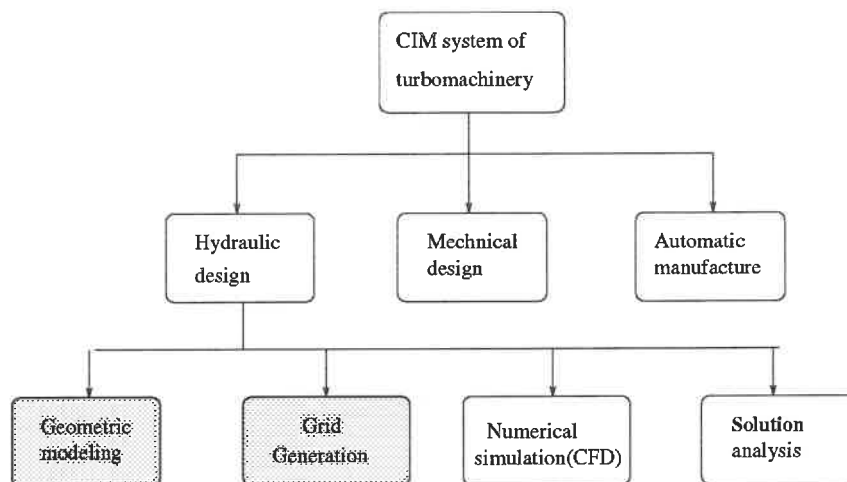
Figure 1.1: Situation of hydraulic design in CIM

can be viewed as an extension of the traditional discipline of geometry combined with numerical or approximation theory. Coupled with computational technology(FEM, CFD,...) this yields a powerful tool for the design and analysis of complete configurations.

The three most widely used modeling methods are wireframe modeling, surface modeling and solid modeling[6]. In the hydraulic design of turbomachinery, geometric modeling is required for an accurate representation of the wetted surfaces of a particular component. From these surface, a closed domain or volume is defined for simulation purposes, and it is these surfaces that will support the boundary of the domain and therefore the boundary conditions.

Because of the complexity of the shapes involved in this research, a Surface geometric modeling representation was chosen and developed. This will form the basis for the following aspects of the geometric design of the turbomachines.

1. Representation: The physical shape of a component is given and presumed to

be fixed. A mathematical approximation is computed.

2. Design: A procedure whereby a new shape is created to satisfy some operational or aesthetic objectives. This involves the manipulation of the variables defining the shape until the objectives are met.

3. Discretization: The domain is subdivided into discrete elements where the partial differential equation(PDE) governing the fluid flow will be solved numerically

These three aspects are, of course, closely related. At first the model represents a physically nonexistent object. After selecting a model suitable for numerical analysis and evaluation, the fluid flow inside the domain is simulated. Analysis of the solution can be achieved by visualization techniques using computer graphics tools. At any point in this process, the geometric model provides all the underlying geometric information.

1.3   Grid Generation

Presented simply, a grid is a mechanism to organize a geometric domain for calculation purposes. This should be done in such a manner that points (nodes) and the surrounding neighborhood (elements) can be conveniently located and easily related in physical space for the discretization.

Grids can be classified into structured and unstructured types depending on the nature of the connectivity. Structured grids (body-fitted or curvilinear) can be generated using the following basic techniques[1]:

- The Conformal Mapping Technique.

- The Algebraic Function Technique.

- The Elliptic Differential Equation Technique.

The last two methods are the most widely used techniques to generate body-fitted grids for flow calculations in turbomachinery. In this research, a transfinite method based on the Algebraic Function Technique is developed to generate an initial three dimensional grid inside each component of the turbomachine. An elliptic differential equation technique is then used to smooth the grid.

1.4   Turbomachinery Components

A turbomachine is composed of three major components: the spiral casing, the runner and the diffuser. These are arranged as shown in Figure 1.2. The geometry of the spiral casing is a three dimensional spiral shell. The outlet of the spiral casing is connected to inlet of the runner. At the junction, there are guiding vanes and wicket gates to control the flow as it enters the runner, therefore the shape of the spiral casing changes from that of a pipe to that of a spiral. The shell of the runner is composed of two surfaces: one is called the hub, and the other is called the shroud. These are connected to the spiral casing. The blade of the runner is designed as a streamlined surface to produce power efficiently. The outlet of the runner is connected to the inlet of the diffuser. The inlet section of the diffuser is circle and the outlet section is a rectangle.

These differences in geometry and topology require different data for their definition. Accordingly different methods are developed to cope with these differences. A special geometry description language and geometry interpreter, designed to represent the geometry of components, have been developed to automate the process

of geometry construction. The geometry of each component is defined as follows:
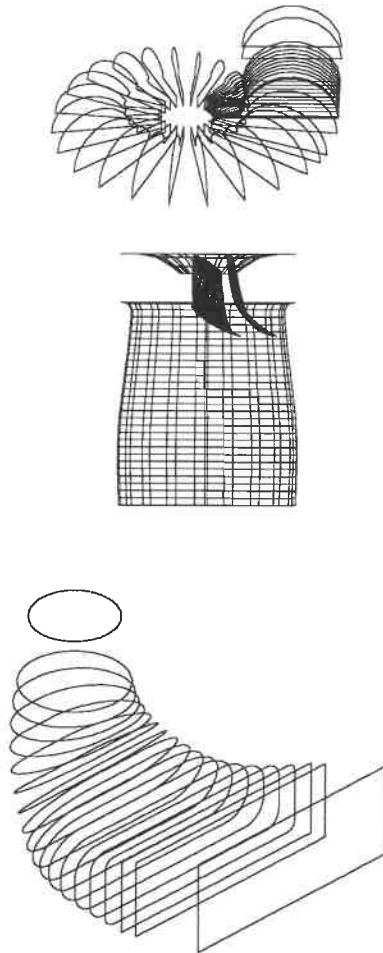
Figure 1.2: The three major components

1. Spiral casing: The data available to define the spiral casing is a series of three-

dimensional sections. All sections are circles attached to both sides of the stay vane. We define a spine as the centre of these circles and it extends into the intake pipe. The geometrical representation of the spiral casing is a surface which is created by skinning the sections.

2. Runner: The runner is defined as a group of blades embedded inside a channel which is composed of two co-axis surfaces of revolution called the hub and the shroud. Inside the channel, blades are placed periodically around the co-axis of the surfaces of revolution.

3. Diffuser: The diffuser is defined by a series of three-dimensional plane sections cutting across the flow. The center of each sections is placed on a spine which follows the direction of the flow. Each section is composed of elliptic arcs joined with segments determining the local shape. This allows for a smooth transition from a circle at the inlet, to a rectangle at the outlet, while still describing accurately all intermediate sections.

1.5   Goals and Objectives

The main difficulties in hydraulic turbomachinery design come from the rapid changes in geometry, and the rotation of the runner relative to the joining part of the spiral, making the flow both complex and unsteady. Most commercial mechanical CAD/CAM systems were developed to deal with general mechanical design. Due to the complexity of the geometry of the turbomachine, to use them in hydraulic design would be very complicated.

The goal of this research is to develop a special purpose CAD system for the hydraulic design of turbomachinery. The users of the system would be engineers

who do computational fluid dynamics(CFD) research to simulate the flow inside each component of turbomachinery. This CAD system should have the ability to model the three major components of the turbomachinery. It should also discretize these model for numerical simulation of the flow inside. An automatic grid generation tool was developed to discretize the geometric representation.

In this system, a surface geometric modeling library which uses Non-Uniform-Rational-BSplines (NURBS) has been implemented using the C language with object-oriented methods. Based on this geometric library, a CAD package for hydraulic design of turbomachinery was built along with a geometry language interpreter and a grid generation capability.

Chapter 2

Geometric Modellers and CAD/CAM Systems

2.1 Computer-Aided Geometry Design

Computer aided geometry design(CAGD) is a primary ingredient in computer-aided design and computer-aided manufacturing (CAD/CAM) systems, computer graphics, computer art, animation, simulation, etc. The main objective of the CAGD research is to develop mathematical and numerical concepts and associated technologies for geometric modeling. There are three common types of geometric modellers:[6]

- Wireframe modellers

- Surface modellers

- Solid modellers

2.1.1 Wireframe Modellers

A wireframe modeller is composed of lines and curves defining the edges of an object. These systems were initially two dimensional and were intended as drafting tools. Extensions are currently available to handle three-dimensional representations. They can define two dimensional objects exactly, but in dealing with three dimensional objects, they are frequently ambiguous. Figure 2.1 shows a good example, since there are two equally likely interpretations of the object.
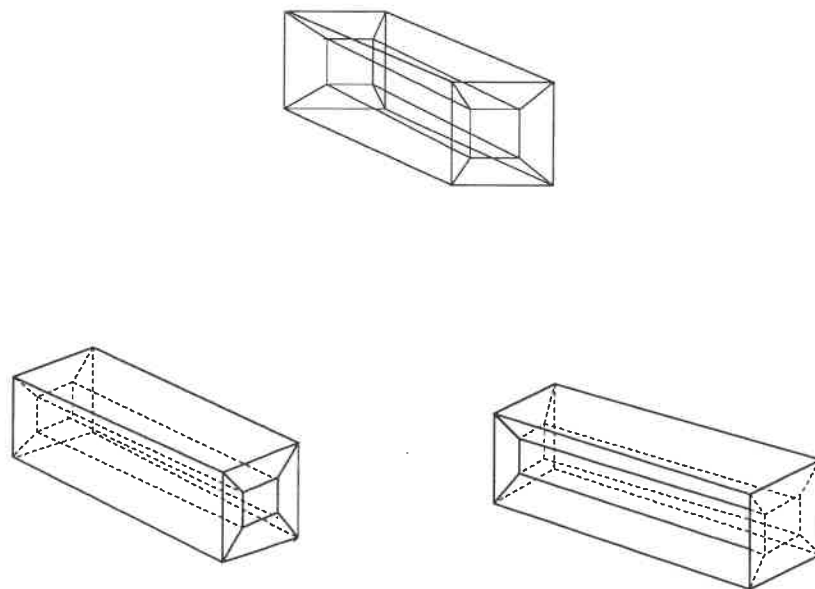
Figure 2.1: Ambiguity in wireframe modeller

Another problem is that wireframe systems usually have no internal logical tests to prohibit nonsense object. Figure 2.2 shows such an object created by wireframe system. Implied faces interpenetrate in a way that makes the interpretation of a

physically realizable solid object impossible.
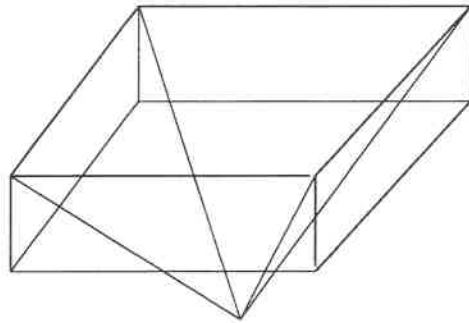
Figure 2.2: Nonsense object in wireframe modeller

2.1.2   Surface Modellers

A surface modeller represents the boundary of an object as a collection of parametric surfaces. This approach was first developed to create complicated sculptured surfaces to replace the classic lofting techniques used in the shipbuilding, automotive, and aircraft industries. Initially parametric cubic patches of Coon and Ferguson and the techniques of Bezier were used to represent these sculptured surfaces. New ground was broken in the mathematics of parametric surfaces with the advent of tensor products, Non-Uniform-Rational-BSpline(NURBS)[4]. The advantage of these lies in using an homogeneous coordinate system. They can represent both the classical and the free forms of curves and surfaces in a unique mathematical form. It has since become a standard representation in geometric modellers, surface modeling systems should have the ability to:

1. Construct free form of curves and surfaces by polynomial representation from a series of points(B-spline, Bezier-spline, NURBS)

2. Construct classic curves and surfaces(ellipse, circle, line, plane, revolution surface, ruled surface, sphere, etc...)

3. Construct more complex three dimensional curves or surfaces by joining separate curves or surfaces(tensor product NURBS).

A surface modeling based on NURBS can exactly define the shape of most three-dimensional objects. In addition, a very important feature found in many surface modellers is the faceting capability. It refers to the generation of polygons on the surfaces which can then be used in many computer graphics algorithms(hidden line removal, shading, rendering) to draw a realistic picture of the three dimensional object.

The limitation of surface modellers is that they lack the concept of volume since they do not, in general, define an inside and outside. To define unambiguously the inside of an object, one requires a closed surface. For practical applications, the whole boundary of an object cannot be seperated by a single surface, but rather by a series of "patches" joining together. The information on how these "patches" are linked to one another is required in order to make a precise difination of inside and outside. This information is general refered as "topology". This is illustrated in Figure 2.3 where the intersection of a cylinder with a sphere is ambiguous because surface modeling lacks the concept of volume. The correct boundary surface of the new object cannot be created automatically. Comparing with wireframe modeling, surface modeling can precisely represent the boundary of 3D objects and can produce all the necessary information for realistic pictures.
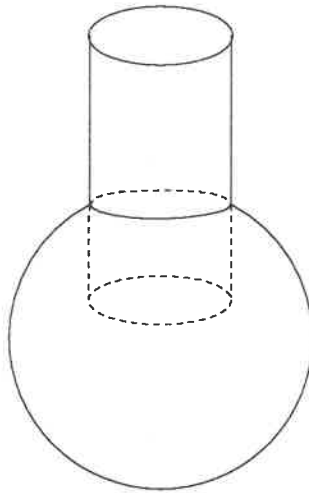
Figure 2.3: Object created by surface modeller

2.1.3   Solid Modeller

Solid modellers were intended to overcome the limitations of the other schemes by adding the concept of a solid interior. They are developed from surface modellers by adding the topological definition of an object. There have been several different approaches. The two most widely used methods are : Boundary Representations(B-Rep) and Constructive Solid Geometry(CSG).

In Boundary representation, a three-dimensional object can be represented as a volume bounded by a closed shell with orientation. The shell boundary is composed of a collection of faces joined to each other along their edges. This solid modeling system is composed of two parts. One is the geometrical definition of basic geometry primitives like points, curves, surfaces etc, and the other is the topology which defines the relationship between all the geometric primitives building the three di-

mensional object. The topological definition of a shell distinguishes the interior and exterior of a three dimensional object by its orientation. Figure 2.4 shows a simple topological representation of a tetrahedron bounded by a shell composed of four triangular faces. Every face is bounded by an oriented loop composed of three edges. Every edge is bounded by two vertices. This topological representation defines the detailed relationship between all the geometric primitives which make up the object. From these relationships, the geometric modeller can determine if a point is on the boundary surface, inside the object, outside the object, or on the surface of the object. In the second approach, Constructive solid geometry, Boolean operations are

Figure 2.4: Topology of B-rep object

used to create the three-dimensional objects. Models are constructed as a combination of simple solid primitives, such as cylinders, spheres, cuboids. The primitives are combined using the union, intersection and difference operators.

2.2   CAD/CAM Systems

Computer-Aided-Design(CAD) and Computer-Aided-Manufacture(CAM) refer to software tools which have been developed to help engineers design and analyse

products. They differ from geometric modeling systems which can only model the geometry of objects. CAD/CAM systems can manipulate objects by applying design and analysis methods to them until some requirements are met. Those design and analysis methods differ from one system to another and depend on specific applications. For mechanical CAD/CAM system, these methods are structural finite element analysis, numerically controlled machine tool path generation, interference detection, mechanical analysis, etc... For hydraulic CAD/CAM system, the most important analysis methods are grid generation and numerical simulation. The relation between hydraulic CAD/CAM system and geometric modeling is shown in Figure 2.5.
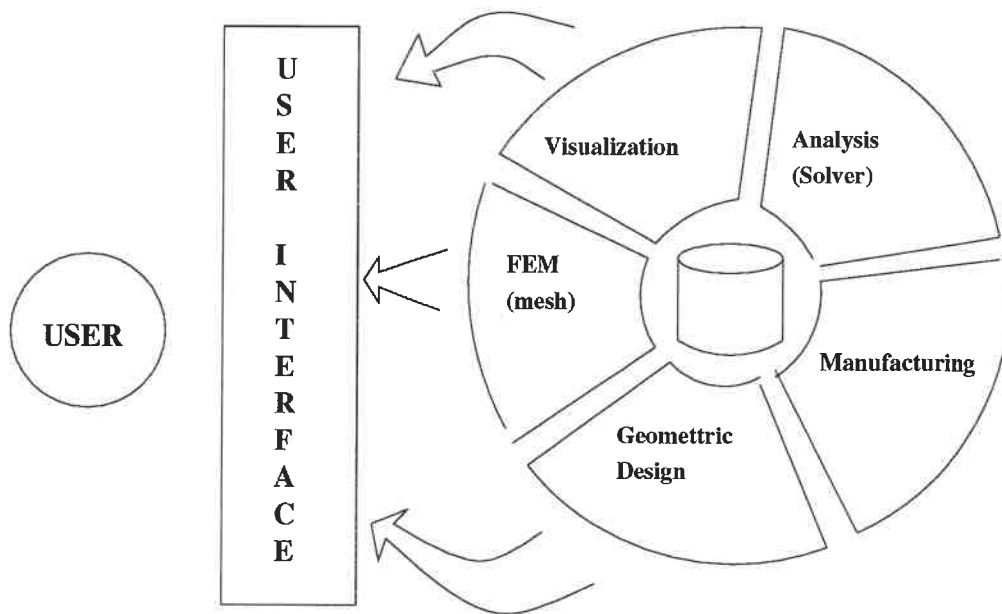
Figure 2.5: Computer Aided Manufacturing for turbomachinery

2.3   Proposed Modeling Approach

The features and characteristics of various modeling approaches were analyzed in section 2.1. The most important requirement was to precisely represent the boundary surfaces of a closed volume. Both solid modeling and surface modeling can meet this requirement. Solid modeling systems can handle the solid interior of objects. However in the present application, since no solid interior geometry manipulation is required. Here Surface Modeling is more suitable.

The shapes considered by the present methodology include different types of blades, surfaces of revolution to represent the hub or shroud of the runner, the skinning type surfaces which represent the shell of the spiral casing and the diffuser. Various curves also need to be considered: lines, classic conic section curves, splines, etc... To meet all these requirements, a surface modeling package using Non-Uniform-Rational-BSplines (NURBS) representation is proposed to represent the geometry of the turbomachinery components. This package should also include basic operations which can be used to manipulate the various curves and surfaces like interpolation, skinning, transferring, evaluating, etc...

Chapter 3

Computer Aided Grid Generation

The numerical methods used in hydraulic design (finite differences, finite volumes, finite elements) require a mesh which represents the discretization of the domain under consideration. Grids can be categorized according to various characteristics or according to the method used for their generation. The most widely used classifications are structured and unstructured grids. The first can be further subdivided into algebraic, body-fitted, or elliptic grids.

3.1 Body-Fitted Structured Grid and Generation Method

The basic characteristic of a body-fitted structured grid is that the coordinate curves (or surfaces) conform to (or align with) the boundaries. The degree of conformity varies from exact at boundaries where these must coincide, to some global alignment in the interior of the region.

In a structured grid, all nodes are surrounded by exactly the same number of el-

ements. These elements can be either rectangular or triangular in shape. The global property of such structured grids is that they can be mapped into a logical square or logical triangle. For complicated domains, the mapping of a structured grid into a single logical square or logical triangle may be difficult or impossible. However such topological limitations can be overcome through domain decomposition, or zoning, which results in a combination of basic zones or subdomains. Several methods to generate body-fitted grids within such basic zones will now be described.

3.1.1 Conformal Mapping Technique

The conformal mapping technique consists in a mapping from the physical domain in (x,y) to a logical domain in (u,v). The domain of the mapping is referred to as the logical space U, while the mapping range is referred to as physical space Ω. Maps or transformations from logical to physical objects yields a system of coordinates on the physical object. Such maps have two important parameters: the dimension k of the logical object and the dimension n of physical object, thus they are labeled X_k^n.

$$X_k^n : U_k \mapsto \Omega_k^n \tag{3.1}$$

It is assumed that $0 < n \leq 3$ and $0 < k \leq n$. In general such maps can be written as:

$$X = X(\xi) \tag{3.2}$$

The coordinate lines in physical space are the images of the coordinate lines in logical space and are thus given by the curves where one of the components of ξ varies while all other components are held constant. Another restriction is that the boundary of the physical space must be the image of the boundary of the logical region under the map. This type of mapping is called conforming mapping.

The most important quantity in a conformal mapping is the Jacobian matrix J. It is defined as the:

$$J_{i,j} = \frac{\partial X_i}{\partial \xi_j} \tag{3.3}$$

In the case $n = k$, the Jacobian matrix is a square matrix and the determinant is defined and referred to as the Jacobian J of the Map X_k^k:

$$j = det(J) \tag{3.4}$$

As a classical example, the choice of

$$Z = f(w) = \frac{w}{2} \pm \sqrt{(\frac{w}{2})^2 - a^2} \tag{3.5}$$

will map the straight line segment in the w-plane

$$
\begin{aligned}
u &= 0 \\
-2a \le\ & v \le 2a
\end{aligned}
\tag{3.6}
$$

into a circle centered on the origin with radius equal to a, in the z-plane.

3.1.2 Elliptic Partial Differential Equation Technique

Differential Equation Techniques are based on solving systems of elliptic partial differential equations. A major advantage of this approach is that the interior grid is very smooth. In our case, an elliptic PDE is used: Poisson's equation[7]. This grid generation method has two disadvantages: it takes more work to solve an elliptic equation, and it is hard to control the interior grid distribution. Adding some concentration will yield a Poisson equation with forcing terms, since the effect of which is not readily predictable.

The most widely used method is the homogeneous Thompson-Thames-Mastin method[5]. For a two dimensional domain bounded by four curves, the method can
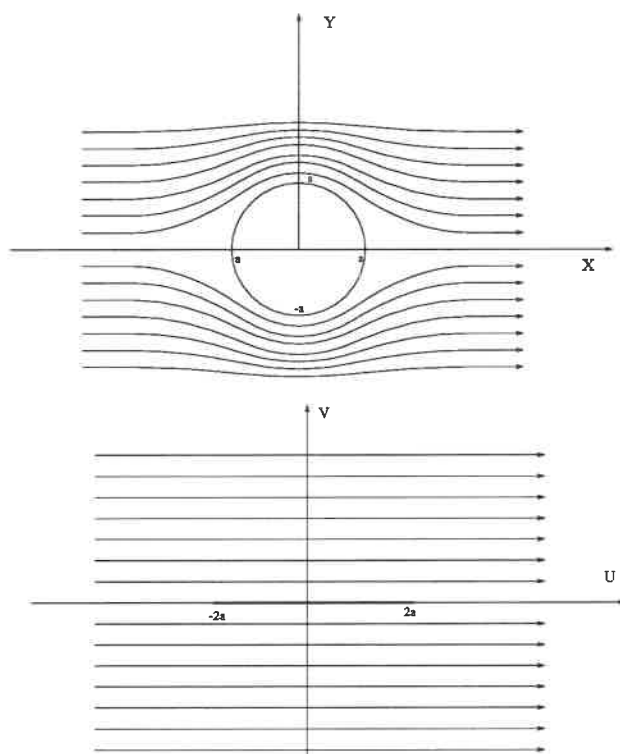
Figure 3.1: Conforming map method

be explained by the following heat-transder analogy: one poses that two opposite sides are thermally insulated while the other pair are set at a given temperature differential. The method arose from the need for an elliptic generator that would produce unfolded grids, which translates to a transformation with positive Jacobian[5]. It requires the inverse transformation,

$$\xi = \xi(x,y) \qquad \eta = \eta(x,y),$$ (3.7)

to satisfy Laplace's equation:

$$\nabla^2\xi = \nabla_x^2\xi = \xi_{xx} + \xi_{yy} = 0$$
$$\nabla^2\eta = \nabla_x^2\eta = \eta_{xx} + \eta_{yy} = 0$$ (3.8)

Where the ξ, η represent the logical parameter space, x, y represent the physical

space. The solution of the resulting field yields a family of constant temperature lines which are to be used as the body-fitted coordinates curves. Figure 3.2 illustrates this method. A second set of curves is obtained by solving the same PDE interchanging the Dirichlet-and-Neuman conditions.
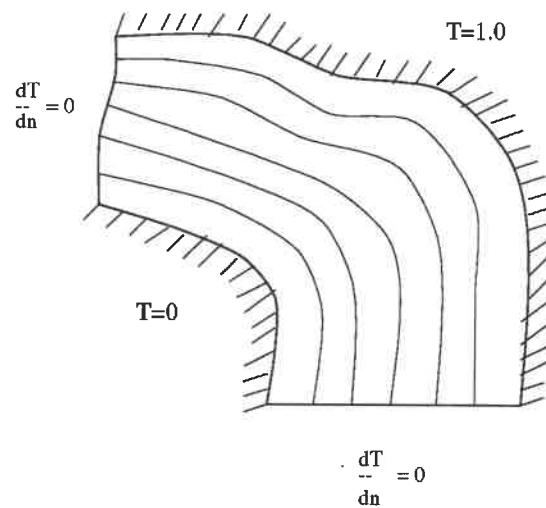
Figure 3.2: Constant temperature line of a domain

3.1.3   Algebraic Function Technique

The Algebraic Technique generates body-fitted grids by using interpolation instead of differential equations. There are two advantages of this technique: rapid computation of the grids compared to the Differential Equation Technique, and direct control over grid point locations. This method consist of a mapping of a parametric space to physical space. The grid generated in the parametric space is mapped by means of explicit algebraic transformation equations. This is both easy

and efficient as the grid is trivially generated in the transformed space and the actual grid in physical space is computed directly by the numerical evaluation of a chosen set of expressions. The control of the grid point locations is obtained by applying some concentration criterion. These advantages are somewhat offset by the fact the interpolation methods may not generate smooth grids; in particular, boundary-slope discontinuities propagate into the interior.

The standard method of Algebraic grid generation is known as transfinite interpolation. In the one-dimensional case, it is the same as linear interpolation. The more interesting case is for a two-dimensional domain. A four sided planar grid-generation problem begins with a description of the boundary of the domain, that is, four equations

$$
\begin{aligned}
X_b(u) &= 0 \quad X_t(u) = 0 \quad 0 \le u \le 1.0 \\
X_l(v) &= 0 \quad X_r(v) = 0 \quad 0 \le v \le 1.0
\end{aligned}
\tag{3.9}
$$

are needed to represent each part of the boundary. The subscripts b, t, l and r stand for bottom, top, left, and right boundary of the logical domain. The simplest example of such a parametrization is the identity map, that is, where the physical region is the unit square just like the logical region. This parameterization is given by

$$
\begin{aligned}
X_b(u) &= (u,0) \quad 0 \le u \le 1.0 \\
X_t(u) &= (u,1) \quad 0 \le u \le 1.0 \\
X_l(v) &= (0,v) \quad 0 \le v \le 1.0 \\
X_r(v) &= (0,v) \quad 0 \le v \le 1.0
\end{aligned}
\tag{3.10}
$$

The first degree Lagrange polynomials $1 - u, u, 1 - v, v$ are used as blending functions in the basic transfinite interpolation formula as:

$$X(u,v) = (1-v)X_b(u) + vX_t(u) + (1-u)X_l + uX_r - (uvX_t(1) +$$

$$+ u(1-v)X_b(1) + v(1-u)X_t(0) + (1-u)(1-v)X_b(0)) \quad (3.11)$$

A bilinearly blended Coon's patch $S(u,v)$ can also be used to generate the grid on three-dimensional surface domain as shown in Figure 3.3. The first two ruled surfaces are defined as:
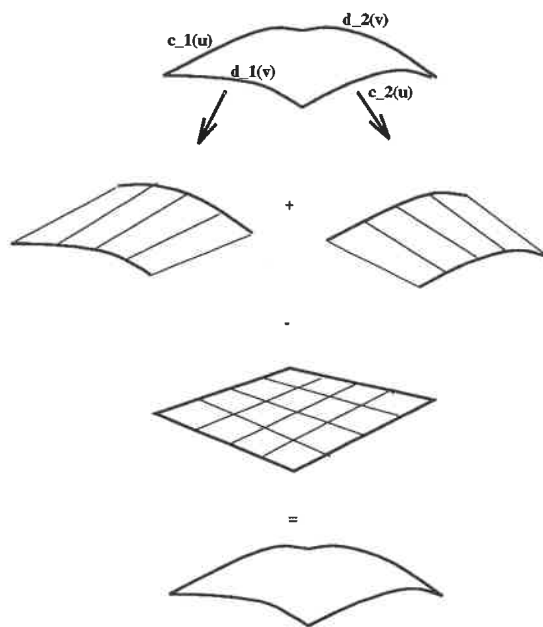
Figure 3.3: Coon's patch construction

$$r_d(u,v) = (1.0 - u) * S(0,v) + u * S(1,v)$$

$$r_c(u,v) = (1.0 - v) * S(u,0) + v * S(u,1) \quad (3.12)$$

where

$$S(u,0) = X_b(u) \quad S(u,1) = X_t(u)$$

$$S(0,v) = X_l(v) \quad S(1,v) = X_r(v) \tag{3.13}$$

are four boundary curves. A linear patch is created by bilinearly interpolating the four corners points as:

$$r_{cd}(u,v) = [(1.0-u) \ u] \begin{bmatrix} S(0,0) & S(0,1) \\ S(1,0) & S(1,1) \end{bmatrix} \begin{bmatrix} 1.0-v \\ v \end{bmatrix} \tag{3.14}$$

Finally, the Coon's patch S(u,v) is given by

$$S(u,v) = r_c + r_d - r_{cd} \tag{3.15}$$

Written in matrix form this is given by:

$$
\begin{aligned}
S(u,v) &= [1.0-u \ u] \begin{bmatrix} S(0,v) \\ S(1,0) \end{bmatrix} + [S(u,0) \ S(u,1)] \begin{bmatrix} (1-v) \\ v \end{bmatrix} \\
&\quad - [1.0-v \ v] \begin{bmatrix} S(0,0) & S(0,1) \\ S(1,0) & S(1,1) \end{bmatrix} \begin{bmatrix} 1.0-v \\ v \end{bmatrix}
\end{aligned} \tag{3.16}
$$

The bilinearly blended Coon's patch does not always yield good results when the geometry of the region is not convex. So to improve accuracy, Bicubic blended Coon's patches can be used. These need more input than just positional. First derivative information is required. The data required now consists of position vector:

$$S(u,0), S(u,1), S(0,v), S(1,v) \tag{3.17}$$

and first derivative vector:

$$S_v(u,0), S_v(u,1), S_u(0,v), S_u(1,v) \tag{3.18}$$

The bicubic blended Coon's patch is obtained by the same method as the blended bilinear Coon's patch. First, two blended surfaces can be constructed from the position vectors and derivative vectors as:

$$
\begin{aligned}
h_c(u,v) &= H_0^3(u)S(0,v) + H_1^3(u)S_u(0,v) \\
&\quad + H_2^3(u)S_u(1,v) + H_3^3(u)S(1,v) \\
h_d(u,v) &= H_0^3(v)S(u,0) + H_1^3(v)S_v(u,0) \\
&\quad + H_2^3(v)S_v(u,1) + H_3^3(v)S(u,1)
\end{aligned}
\tag{3.19}
$$

where the H_i^3 are cubic Hermite polynomials and have the form:

$$
\begin{aligned}
H_0^3(u) &= 2u^3 - 3u^2 + 1 \\
H_1^3(u) &= -2u^3 + 3u^2 \\
H_2^3(u) &= u^3 - 2u^2 + u \\
H_3^3(u) &= u^3 - u^2
\end{aligned}
\tag{3.20}
$$

As for the bilinear patch, a bicubic patch based on four corner is defined:

$$
h_{cd}(u,v) = \left[H_0^3(u)H_1^3(u)H_2^3(u)H_3^3(u) \right]
\begin{vmatrix}
P_{00} & P_{01} & P_{00}^v & P_{01}^v \\
P_{10} & P_{11} & P_{10}^v & P_{11}^v \\
P_{00}^u & P_{01}^u & P_{00}^{uv} & P_{01}^{uv} \\
P_{10}^u & P_{11}^u & P_{10}^{uv} & P_{11}^{uv}
\end{vmatrix}
\begin{vmatrix}
H_0^3(v) \\
H_1^3(v) \\
H_2^3(v) \\
H_3^3(v)
\end{vmatrix}
\tag{3.21}
$$

Finally, bicubically blended Coon's patch can be obtained as:

$$
S(u,v) = h_c + h_d - h_{cd}
\tag{3.22}
$$

3.2 Unstructured Grid

Unstructured grids are composed of two parts: a set of nodes, and the connectivity between them. Compared to the structured grids, unstructured grids have

the ability to represent complicated domains. The most widely used element in unstructured grid is the triangle because the algorithms to generats the triangles are simpler than those for quadrilaterals. For three dimensional grid, the tetrahedron element generally is used. But the generation processing is longer than for structured grids. For the two dimensional domain, one generation technique consists in two steps: first the nodes inside the domain are generated, then the connectivity is created. The nodes in both the interior and on the boundary can be generated by algebraic or elliptic method. One widely used connection phase technique is based on the Delaunay-Voronoi method[2].

3.3   Method Used in This Application

Two facts one must deal with in order to generate grids in turbomachinery components: the geometric complexity and the concentration requirement of the grid. Neither the elliptic nor the Algebraic function methods can generate a grid which respect both the geometry and the concentration.

A method called multiblock structured grid generation is used in this project to generate the grid. It overcomes the difficulty of single domain algebraic method by breaking the complicated domain in several simple subdomains. Then a structured grid is generated inside each subdomain. After the grid has been generated inside each subdomain, a smoothing method is used to smooth the grid inside each subdomain. Finally a global grid is created by combining those subdomain grids. By using this method, a quality grid can be generated with good concentration for each components of the turbomachine.

Chapter 4

Geometric Modeling System

4.1 Introduction

A surface geometric modeling library whose kernel uses a NURBS representation was developped for this project. There are three levels of geometric entities: points, curves, and surfaces. Both the surface and curve entities depend on the point which is therefore the basic entity. Every other entity in the library is created from them. Table 4.1 shows all entities inside the library.

4.2 Point and transformation

The point entity in three-dimensional Cartesian coordinate is defined by a vector with three coordinate values (x, y, z) and is denoted by $\tilde{P}(wx, wy, wz, w)$ in homogeneous coordinates. A projection of a homogeneous point into 3-D space is obtained by dividing the first three coordinates of the point by its homogeneous coordinate w. The normal 3-D space is situated at $w = 1$ as:

$$P(x, y, z) = \tilde{P}(x, y, z, 1) \tag{4.1}$$

Tables 4.1: Geometric entities

Level	Geometry entity
Point entity	Points, Vectors and transform
Curve entity	NURBS curve
	Cubic spline curve
	B-spline curve
	Ellipse and ellipse arc
	Polyline
	Composite curve
Surface entity	BiCubic patch surface
	Plane surface
	Revolution surface
	Ruled surface
	Composite surface

A homogeneous transformation is used to manipulate the geometry objects. It is a 4x4 matrix defined by Eq(4.2), the upper left 3x3 matrix is a rotation matrix, the upper right vector is a translation vector.

$$T = \begin{bmatrix} m_{11} & m_{12} & m_{13} & x \\ m_{21} & m_{22} & m_{23} & y \\ m_{31} & m_{32} & m_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

The coefficients $m_{i,j}$ are defined by rotation angles around the axes as:

$$m_{11} = cos\theta_z cos\theta_y \qquad m_{12} = sin\theta_z cos\theta_x + cos\theta_z sin\theta_y sin\theta_x$$

$$m_{13} = sin\theta_z sin\theta_x - cos\theta_z sin\theta_y cos\theta_x \qquad m_{21} = -sin\theta_z cos\theta_y$$

$$m_{22} = cos\theta_z cos\theta_x - sin\theta_z sin\theta_y sin\theta_x \qquad m_{23} = cos\theta_z sin\theta_x + sin\theta_z sin\theta_y cos\theta_x$$

$$m_{31} = sin\theta_y \quad m_{32} = cos\theta_y sin\theta_x \quad m_{33} = cos\theta_y cos\theta_x \tag{4.3}$$

where the $theta_x, theta_y, theta_z$ are the rotated angle around the axes.

4.3 Curve Representation

4.3.1 B-Spline Representation

A bounded Non-Uniform polynomial B-spline curve $C(u)$ is defined by mapping a bounded segment from a one-dimensional parameter space into homogeneous space as:

$$\tilde{C}(u) = \sum_{i=1}^{n} B_i^k(u)\tilde{P}_i \quad a \leq u \leq b \tag{4.4}$$

where $\tilde{P}_i$ is a set of points called control points, u is the parameter, a and b are fixed values with $0 \leq a \leq b$ and n is the number of the control points. The functions $B_i^k(u)$ are scalar-valued polynomials of order k in the variable u. They are completely defined by the order k and a knot vector u_j. where $a = u_1 = \ldots = u_k \leq u_{k+1} \leq u_{k+2} \leq \ldots \leq u_n \leq u_{n+1} = \ldots = u_{n+k} = b$.

The value of $B_i^k(u)$ for a fixed u is calculated with the recursive algorithm as:

$$B_i^k(u) = \frac{u_{i+n-k} - u}{u_{i+n-k} - u_{i-1}} B_{i-1}^{k-1}(u) + \frac{u - u_{i-1}}{u_{i+n-k} - u_{i-1}} B_i^{k-1}(u) \tag{4.5}$$

where

$$B_i^1(u) = 1 \quad if \quad u_i \leq u < u_{i+1} \tag{4.6}$$

$$= 0 \quad else$$
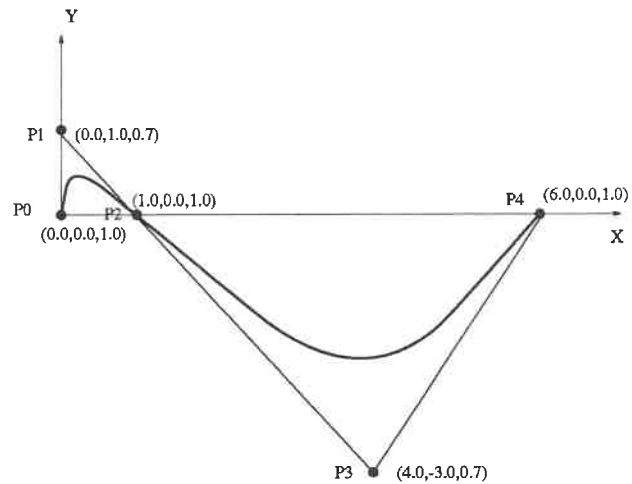
Figure 4.1: Degree 2 NURBS with 5 control point and weight
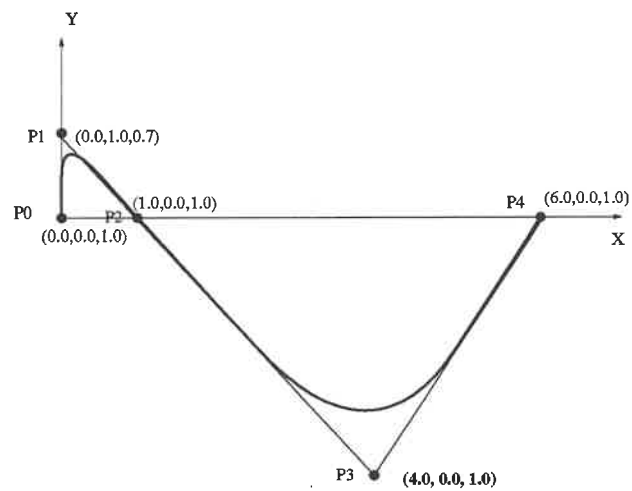
Figure 4.2: Degree 2 BSpline with 5 control point

4.3.2 NURBS Representation

A Rational Bspline $C(u)$ is obtained by projecting a Bspline from homogeneous space(4D) into three dimensional space as follows:

$$C(u) = \frac{\sum\limits_{i=1}^{n} B_i^k(u)\tilde{P}_i w_i}{\sum\limits_{i=1}^{n} B_i^k(u)w_i} \quad a \le u \le b \tag{4.7}$$

The non-rational spline is a special case with $w_i = constant = 1$. The difference between rational and non-rational Bspline is the influence of the control point P_i conforming to the weight w_i. If the weight approaches zero, then the influence of this control point disappears. By using the weight, the NURBS can be made to exactly represent the conic curves[4]. Figure 4.1 and Figure 4.2 show two NURBS curves with same x, y values but different weights.

4.3.3   Some Operations of NURBS

A number of operations for manipulation of the curves are required in the present work. These will be described.

Evaluating a Derivative

The derivative of a NURBS curve is obtained by differentiating deriving Eq (4.7):

$$\frac{\partial C(u)}{\partial u} = \frac{\dfrac{\partial\left[\sum\limits_{i=1}^{n} B_i^k(u)\tilde{P}_i\right]}{\partial u}\sum\limits_{i=1}^{n} B_i^k(u)w_i - \dfrac{\partial\left[\sum\limits_{i=1}^{n} B_i^k(u)w_i\right]}{\partial u}\sum\limits_{i=1}^{n} B_i^k(u)\tilde{P}_i}{\left[\sum\limits_{i=1}^{n} B_i^k(u)w_i\right]^2} \tag{4.8}$$

Any order derivative vector of the NURBS can be calculated by using the recursive algorithm defined by Eq 4.5.

Intersection Between Two NURBS Curves

Computing the intersection of two curves noted $P(u)$ and $Q(v)$ in three-dimensional space requires solving the following systems of equations:

$$
\begin{aligned}
P_x(u) - Q_x(v) &= 0 \\
P_y(u) - Q_y(v) &= 0 \\
P_z(u) - Q_z(v) &= 0
\end{aligned}
\tag{4.9}
$$

where the unknowns are u and v and $P_x(u)$ is the x coordinate of $P(u)$. These are three nonlinear equations with two unknowns. A Newton-Raphson method is first used to numerically find the intersection in two dimensions. The intersection point is then verified in the last dimension to determine whether these two curves do intersect.

Arc Length of a NURBS Curve

The arc length of a parametric curve is defined as[3]

$$
S(u) = \int_{u_0}^{u_1} \left\| \frac{\partial P(u)}{\partial u} \right\| du
\tag{4.10}
$$

If we define $f(u)$ as:

$$
f(u) = \left\| \frac{\partial P(u)}{\partial u} \right\|
\tag{4.11}
$$

we can use the Romberg integration scheme to obtain:

$$
\begin{aligned}
S(u) &= \int_{u_0}^{u_1} f(u)\,du \\
&= \frac{h}{2}[f(u_0) + f(u_1) + 2\sum_{j=1}^{m-1} f(u_j)] - \frac{u_1 - u_0}{12} h^2 f''(\mu)
\end{aligned}
\tag{4.12}
$$

where $u_o < \mu < u_1$, $h = (u_1 - u_0)/m$ and $u_j = u_0 + jh$ for $j = 0,1,\ldots,m$.

Discretizing a Curve

In order to generate the grid, parametric curves need to be discretized. There are two widely used methods: direct evaluation with desired parameter values, or discretization of the curve with respect to arc length. The latter method is used when a specific nodal distribution along the curve is sought. The first method is used when only an approximate distribution is required.

The parametric value u_i corresponding to arc length S_i is obtained by solving equation:

$$S_i = \int_{u_0}^{u_1} \left\| \frac{\partial P(u)}{\partial u} \right\| du \tag{4.13}$$

The position vector is obtained by substituting the value of u_i in Eq 4.7. The Eq 4.13 was solved iteratively by Newton's method.

4.4   Construction of Curve Entity

The method of construction of the various curve entities(polyline, circular arc, elliptic arc, cubic spline, and composite curves) will now be described. A NURBS representation is adopted throughout.

4.4.1   Polyline Interpolation

Polylines can be represented by a NURBS of order 2 and by taking $w_i = 1.0$ which has a uniform parametrization. The NURBS that represents a polyline interpolated through a series of points can be written as:

$$C(u) = \sum_{i=1}^{n} B_i^2(u) P_i \tag{4.14}$$

4.4.2 Ellipse and Circle

A NURBS of order 3 can be used to represent a circle exactly[14]. The NURBS that represents a circle lying on $x - y$ plane with center at the origin and radius of 1.0, as shown in Figure 4.3, can be written as:

$$C(u) = \frac{\sum_{i=1}^{9} B_i^3(u) P_i}{\sum_{i=1}^{9} B_i^3(u) w_i} \tag{4.15}$$

The homogeneous control points of first quadrant arc are $(1.0, 0.0, 0.0, 1.0)$, $(1.0, 1.0, 0.0, \frac{\sqrt{2}}{2})$, $(0.0, 1.0, 0.0, 1.0)$, the other are obtained by symmetry. A uniform knot vector is used in this case. Ellipse can be constructed by using same method[14].
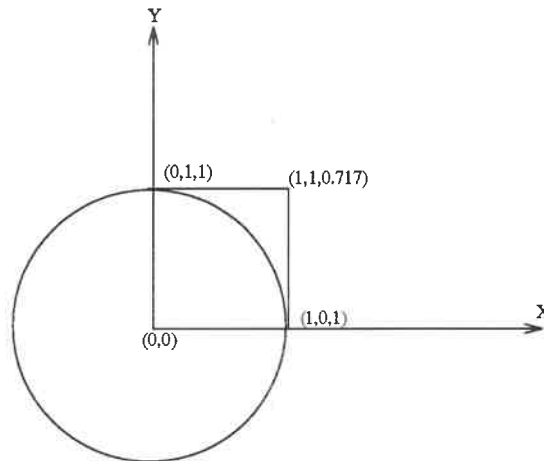
Figure 4.3: NURBS representation of a Circle

4.4.3   Cubic Spline

Given a set of points P_i $(0 < i < L)$ and corresponding parameter values $u_0, ..., u_L$, a cubic spline interpolation through those points with specified boundary conditions[3] can be represented by a NURBS curve $C(u)$ with knot vector u_i and unknown control points $d_{-1}, ..., d_{L+1}$ such that $C(u_i) = P_i$. A cubic spline can also be written as a piecewise Bezier curve[3], with the Bezier control points b_{3i}.

$$P_i = b_{3i} \quad i = 0, 1 \ldots L \tag{4.16}$$

The inner Bezier points $b_{3i\pm1}$ are related to the P_i by

$$P_i = \frac{\triangle_i b_{3i-1} + \triangle_{i-1} b_{3i+1}}{\triangle_{i-1} + \triangle_i} \quad i = 1, \ldots L - 1 \tag{4.17}$$

If we set $\triangle_i = \triangle u_i = u_i - u_{i-1}$, then the relation between $b_{3i\pm1}$ and the control points are defined as:

$$b_{3i-1} = \frac{\triangle_i d_{i-1} + (\triangle_{i-2} + \triangle_{i-1}) d_i}{\triangle_{i-2} + \triangle_{i-1} + \triangle_i} \quad i = 2, \ldots L - 1$$

$$b_{3i+1} = \frac{(\triangle_i + \triangle_{i+1}) d_i + \triangle_{i-1} d_{i+1}}{\triangle_{i-1} + \triangle_i + \triangle_{i+1}} \quad i = 1, \ldots L - 2 \tag{4.18}$$

Near the endpoint of the curve it becomes :

$$b_2 = \frac{\triangle_1 d_0 + \triangle_0 d_1}{\triangle_0 + \triangle_1}$$

$$b_{3L-2} = \frac{(\triangle_{L-1} d_{L-2} + \triangle_{L-2} d_{L-1})}{\triangle_{L-2} + \triangle_{L-1}} \tag{4.19}$$

now b_i can be eliminated to get the equation

$$(\triangle_{i-1} + \triangle_i) P_i = \alpha_i d_{i-1} + \beta_i d_i + \gamma_i d_{i+1} \tag{4.20}$$

The coefficients $\alpha_i, \beta_i, \gamma_i$ are defined as:

$$\alpha_i = \frac{\triangle_i^2}{\triangle_{i-2} + \triangle_{i-1} \triangle_i}$$

$$\beta_i = \frac{\triangle_i(\triangle_{i-2} + \triangle_{i-1})}{\triangle_{i-2} + \triangle_{i-1} + \triangle_i} + \frac{\triangle_{i-1}(\triangle_i + \triangle_{i+1})}{\triangle_{i-1} + \triangle_i + \triangle_{i+1}}$$

$$\gamma_i = \frac{\triangle_{i-1}^2}{\triangle_{i-1} + \triangle_i + \triangle_{i+1}} \tag{4.21}$$

A linear system can be obtained from Eq 4.21:

$$\begin{vmatrix} \alpha_0 & \beta_0 & \gamma_0 \\ & \alpha_1 & \beta_1 & \gamma_1 \\ & & \cdots \\ & & & \cdots \\ & & \alpha_{L-1} & \beta_{L-1} & \gamma_{L-1} \\ & & & \alpha_L & \beta_L & \gamma_L \end{vmatrix} \begin{vmatrix} d_{-1} \\ d_0 \\ \cdots \\ \cdots \\ d_L \\ d_{L+1} \end{vmatrix} = \begin{vmatrix} r_0 \\ \cdots \\ \cdots \\ r_{L-1} \\ r_L \end{vmatrix} \tag{4.22}$$

Here $r_l = (\triangle_{i-1} + \triangle_i)P_i$.

This system has L+3 unknowns with L+1 equations. Extra boundary conditions must be given to eliminate d_{-1} and d_{L+1}. There are two common types boundary conditions: specified tangent vectors at the end points; or specified zero curvature at the end points. For any arbitrary boundary conditions $d_{-1} = b_1$ and $d_{L+1} = b_L$, a tridiagonal system can be obtained as :

$$\begin{vmatrix} \beta_0 & \gamma_0 \\ \alpha_1 & \beta_1 & \gamma_1 \\ & \cdots \\ & & \alpha_{L-1} & \beta_{L-1} & \gamma_{L-1} \\ & & & \alpha_L & \beta_L \end{vmatrix} \begin{vmatrix} d_0 \\ \cdots \\ \cdots \\ d_{L-1} \\ d_L \end{vmatrix} = \begin{vmatrix} r_0 - \alpha_0 * d_{-1} \\ \cdots \\ \cdots \\ r_{L-1} \\ r_L - \gamma_L * d_{L+1} \end{vmatrix} \tag{4.23}$$

The tridiagonal system is solved by Gaussian elimination. Different parameterizations yield different node vectors of the NURBS and coefficients $\alpha_i, \beta_i, \gamma_i$. In this work two types of parameterization methods are chosen. One is uniform, that is

$\triangle u_i = u_{i+1} - u_i = 1.0$. The other is based on chord length, that is, $\triangle u_i = u_{i+1} - u_i$ $= \|P_{i+1} - P_i\|$.

4.5   Composite Curve

A single curve or surface cannot handle all the objects occuring during design activities. The composite curve and surface allow more freedom to the designer when only C_0 degree continuity is required. In this application, composite curves will be used to define the profile, the generatrix curve of the surface of revolution, the sections, etc. A composite curve is constructed by joining two or more curves in sequence. At their junction, two curves must have the same boundary point.

An individual NURBS curve within a composite curve does not necessarily have the same orientation. The operations which apply on a single curve cannot be applied directly on a composite curve. An algorithm was developed to transfer the parameter value between the composite curve and each single curve. The relation between the parameter of the composite and the parameter of constituent curve is defined as:

$$U_composite \;\; = \;\; \sum_{i=1}^{n} IU_i \quad\quad i = 1, ..., n \tag{4.24}$$

$$I \;\; = \;\; 1 \quad\quad same \;\; orientation$$

$$I \;\; = \;\; -1 \quad\quad opposed \;\; orientation$$

$$\tag{4.25}$$

In this section we will apply these concepts and methods to specific objects found in turbomachine.
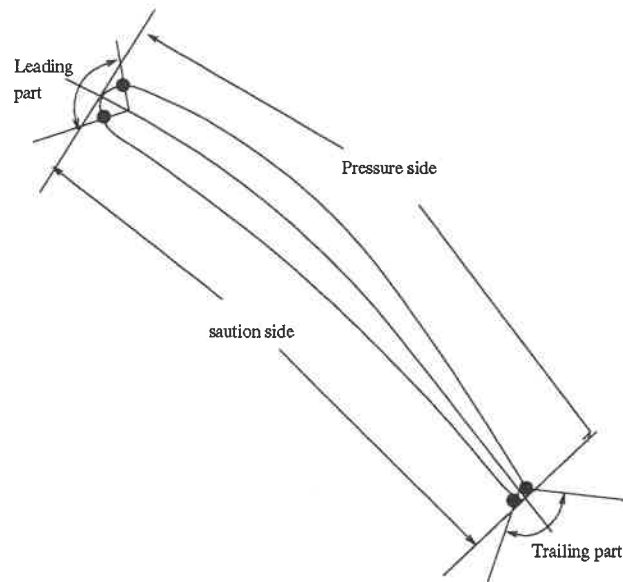
4.5.1 Blade Profile

Figure 4.4: Profile represented by a composite curve

The blade of the runner is defined by a series of cross sections called profiles. These are curves which lie on surfaces of revolution called supports. The profile also can be regarded as the intersection curve between the support and the blade surface. In the present application, it is represented as a composite curve. Figure 4.4 illustrates a profile composed of four curves: the pressure and suction sides, and the leading and trailing edges curves.

4.5.2 Diffuser Section

The diffuser is defined similarly by a series of sections which are also composite curves. Figure 4.5 shows a typical section consisting of eight curves, four line
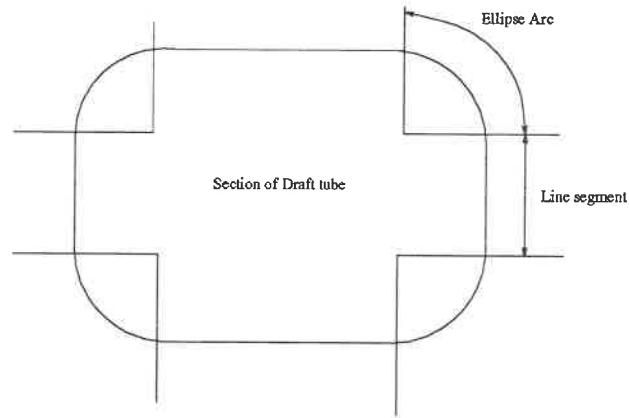
segments and four elliptic arcs.

Figure 4.5: Diffuser Section represented by a composite curve

4.5.3 Spiral Casing Section

Figure 4.6 shows a section of spiral casing represented by a composite curve which is composed of 6 single curves.

4.6 Construction of the Surface Entity

4.6.1 Tensor Product NURBS Surface

Whereas a curve requires one parameter for its definition, a surface requires two: (u, v), where $u_0 \leq u \leq u_1$, $v_0 \leq v \leq v_1$. A tensor product NURBS surface of degree
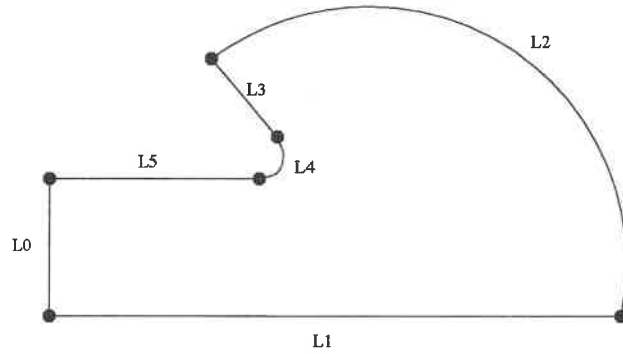
Figure 4.6: Spiral casing Section constructed by using 6 single curves

(p, q) has the form:

$$P(u,v) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} B_i^p(u)B_j^q(v)P_{i,j}}{\sum_{i=1}^{m}\sum_{j=1}^{n} B_i^p(u)B_j^q(v)w_{ij}} = \sum_{i=1}^{m}\sum_{j=1}^{n} R_{i,j}^{p,q}(u,v)P_{ij} \qquad (4.26)$$

The control points P_{ij}, are arranged in a topological rectangular array called the control net. The $B_i^p(u)$ and $B_j^q(v)$ are the univariate rational basis functions defined by Eq 4.7. The $R_{i,j}^{p,q}$ is the bivariate B-spline basis function defined by

$$R_{i,j}^{p,q}(u,v) = \frac{B_i^p(u)B_j^q(v)w_{i,j}}{\sum_{r=1}^{m}\sum_{s=1}^{n} B_r^p(u)B_s^q(v)w_{r,s}} \qquad (4.27)$$

Since the tensor product NURBS surface represents a bidirectional curve scheme, the previously described operations on NURBS curves can be easily extended to surfaces. Fixing one of the parameters, that is $u = u_i$ $u_0 \le u_i \le u_1$, then allowing v to vary from v_0 to v_1 yields a NURBS curve $C(v) = P(u_i, v)$ on the surface. All the operations applicable on the curve are available for the curves on surface.
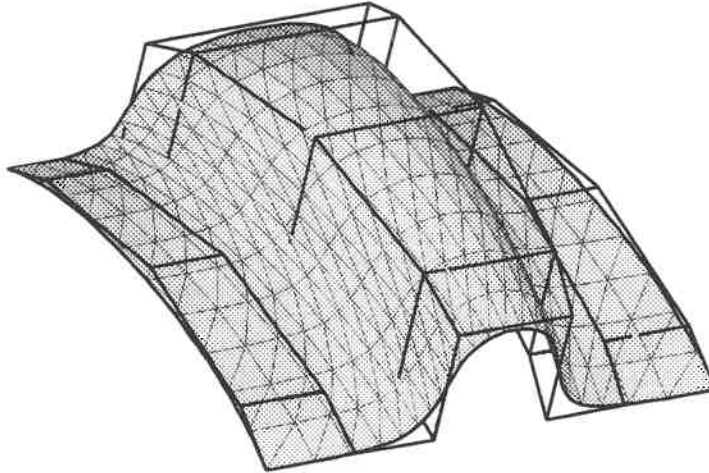
Figure 4.7: NURBS surface with control point net

4.6.2 Surface of Revolution

A surface of revolution is obtained by rotating a curve which lies in the XZ-plane around the Z-axis. A full surface of revolution is obtained by revolving the curve $360°$ about the Z-axis. The required surface has the form

$$S(u,v) = \sum_{i=1}^{9} \sum_{j=1}^{n} R_{i,2;j,q}(u_i, v) P_{ij} \qquad (4.28)$$

Figure. 4.8 shows a sphere created by revolving a circular arc.

4.7 Composite Surface

A composite surface is a collection of individual surfaces which are joined together to form a continuous, more complex surface. In the present work, only the most widely used Bicubic Patch Composite Surface(BPCS) is implemented.
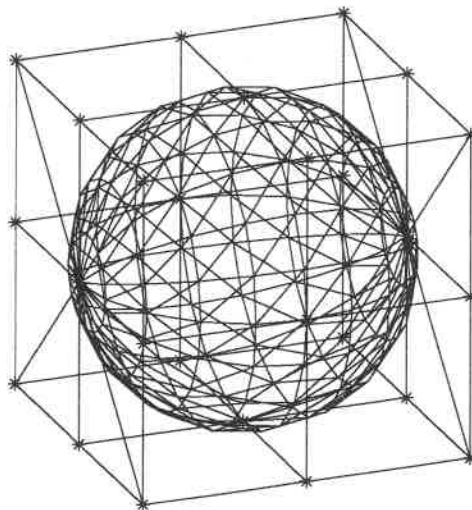
Figure 4.8: Sphere created by revolving a arc

4.7.1 Bicubic Patch Surface

For two given points P_0, P_1 and two tangent vectors P_0^u, P_1^u, an Hermitian interpolation can be defined by

$$P(u) = H_0^3(u)P_0 + H_1^3(u)P_1 + H_2^3(u)P_0^u + H_3^3(u)P_1^u \tag{4.29}$$

For a bicubic patch, bidirectional Hermitian interpolation also can be defined in the form:

$$P(u,v) = \begin{bmatrix} H_0^3(u) H_1^3(u) H_2^3(u) H_3^3(u) \end{bmatrix} \begin{vmatrix} P_{00} & P_{01} & P_{00}^v & P_{01}^v \\ P_{10} & P_{11} & P_{10}^v & P_{11}^v \\ P_{00}^u & P_{01}^u & P_{00}^{uv} & P_{01}^{uv} \\ P_{10}^u & P_{11}^u & P_{10}^{uv} & P_{11}^{uv} \end{vmatrix} \begin{vmatrix} H_0^3(v) \\ H_1^3(v) \\ H_2^3(v) \\ H_3^3(v) \end{vmatrix} \tag{4.30}$$

where the $H(u)$ and $H(v)$ are the Hermite polynomial defined by Eq 3.20. The 4x4 matrix is called the geometric coefficient matrix. It is composed of the positions of

the 4 corner points, the tangent vectors of the 4 corner points and the twist vector of the 4 corner points[7].

4.7.2 Bicubic Patch Composite Surface (BPCS)

A BPCS is constructed by assuring the continuity of the two joined patches at a common boundary. To create a C^1 BPCS from a MxN lattice, first, piecewise cubic curves are fitted to all rows and columns of the lattice using methods which were developed in section 4.4.3 for curves. Note that all curves in the 'u' direction will have the same parametrization, and so will all the curves in 'v' direction. This will create a mesh of C^2 piecewise cubics that will represent the boundary curves of all the patches. From these curves, the 8 corner tangent vectors at the 4 corner points can be evaluated for each patch. There remains to evaluate the four twist vectors at the corners. For simplicity, in our case, zero twist vectors will be assumed, guarantying smoothness at the corner which four patches share. So the bicubic patch with zero twist vector at the corners can be written as:

$$P(u,v) = \left[H_0^3(u)H_1^3(u)H_2^3(u)H_3^3(u)\right] \begin{vmatrix} P_{00} & P_{01} & P_{00}^v & P_{01}^v \\ P_{10} & P_{11} & P_{10}^v & P_{11}^v \\ P_{00}^u & P_{01}^u & 0 & 0 \\ P_{10}^u & P_{11}^u & 0 & 0 \end{vmatrix} \begin{vmatrix} H_0^3(u) \\ H_1^3(u) \\ H_2^3(u) \\ H_3^3(u) \end{vmatrix} \tag{4.31}$$

Figure 4.9 shows how to create a patch with wireframe net.

4.7.3 Evaluation of a Bicubic Patch with Parameters

Eq 4.31 can be written into matrix form as:

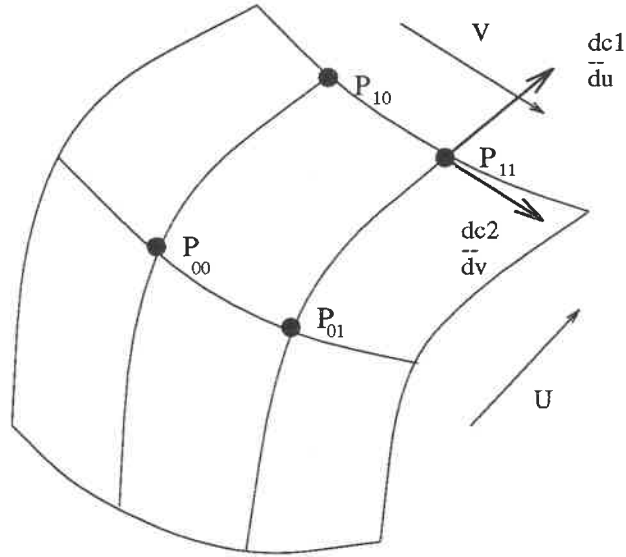$$P(u,v) = \mathbf{H(u)} * \mathbf{M} * \mathbf{H(v)^T} \tag{4.32}$$

Figure 4.9: A patch of composite bicubic surface

The first and second partial derivatives are written as:

$$P^u(u,v) = \frac{\partial \mathbf{H(u)}}{\partial \mathbf{u}} * \mathbf{M} * \mathbf{H(v)^T} = \mathbf{H^u(u)} * \mathbf{M} * \mathbf{H(v)^T}$$

$$P^v(u,v) = \mathbf{H(u)} * \mathbf{M} * \frac{\partial \mathbf{H(u)^T}}{\partial \mathbf{v}} = \mathbf{H(u)} * \mathbf{M} * \mathbf{H^v(v)^T}$$

$$P^{uu}(u,v) = \frac{\partial \mathbf{H^u(u)}}{\partial \mathbf{u}} * \mathbf{M} * \mathbf{H(v)^T} = \mathbf{H^{uu}(u)} * \mathbf{M} * \mathbf{H(v)^T}$$

$$P^{vv}(u,v) = \mathbf{H(u)} * \mathbf{M} * \frac{\partial \mathbf{H^v(v)^T}}{\partial \mathbf{v}} = \mathbf{H(u)} * \mathbf{M} * \mathbf{H^{vv}(v)^T}$$

$$P^{uv}(u,v) = \mathbf{H^u(u)} * \mathbf{M} * \frac{\partial \mathbf{H(v)^T}}{\partial \mathbf{v}} = \mathbf{H^u(u)} * \mathbf{M} * \mathbf{H^v(v)^T} \qquad (4.33)$$

The normal vector can be evaluated by:

$$\mathbf{n} = \mathbf{P^u} * \mathbf{P^v} \qquad (4.34)$$

4.7.4   Evaluation of Composite Bicubic Surface

As for the composite curve, the evaluation of a composite surface with parameters can be transformed into the evaluation of a patch with new parameters. The patch and the parameters are obtained by internal mapping between the BCPS and the constituent patches. There are no orientation problem here, so the mapping is very simple.

The parameter values of the composite surface (u_i, v_j) are arranged in a topologically rectangular array called the knot control net. Each rectangle corresponds to one bicubic patch. So for a given (U,V) value for the composite curve, we can find the corresponding patch $P_{i,j}$ by:

$$P(U,V) = P_{i,j}(u,v) \ \ if \ \ u_i \leq u \leq u_{i+1} \ \ v_j \leq v \leq v_{j+1} \tag{4.35}$$

and corresponding parameter value are:

$$u' = U - u_i \quad v' = V - v_j \tag{4.36}$$

Then, the evaluation is obtained by evaluating the bicubic patches as mentioned in section 3.6.3. Figure 4.10 shows two BCPS surfaces of the blade of Kaplan runner.
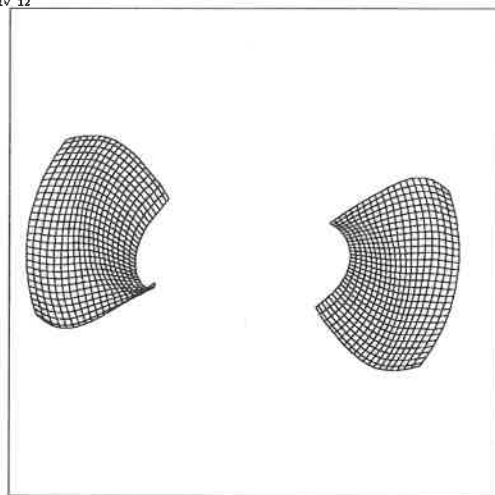
CHLOE IV 12

Figure 4.10: Two BCPS Surface of blade

Chapter 5

Runner

In the current and following chapters, we will apply the various geometric modeling methodologies described in the previous chapters to the practical geometric components of a turbine. A runner is composed of a number of identical blade, positioned around an axis of revolution and within a axisymmetric passage bounded by two surfaces of revolution, the hub and the shroud. For the purposes of analysis, only one fraction of this configuration is required. It is the volume or domain bounded by two consecutive blades, called the 'blade-to-blade' channel. Because of the periodicity of the blade, the flow between two blades is sufficient to represent the flow inside the runner.

Figure 5.1 shows more explicitly this channel bounded by the hub and the shroud with two blades between them. For calculation purposes, this volume should be closed. There are two possible constructions. Figure 5.2 shows the first type of construction. A channel is constructed by two blades and an extensions from the leading edge to the inlet and from the trailing edge to the outlet. These extensions are usually tangent to the camber surface of the profile.

Figure 5.3 shows another type of construction. A channel constructed by creating two periodic surfaces between the blades.

The first type of construction was chosen for this work because it allowed a fines control on concentration near the blade.

5.1 Hub, Shroud and Support Surface

The hub and the shroud are defined as surfaces of revolution obtained by rotating a curve on the X-Z plane around the Z-axis. Because of the sharp corner on the hub, a composite curve description will be required. It is composed of a cubic spline and a polyline is chosen.

Because the hub and the shroud control the flow, it is natural to choose surfaces of revolution between the hub and the shroud as supports. These are created by
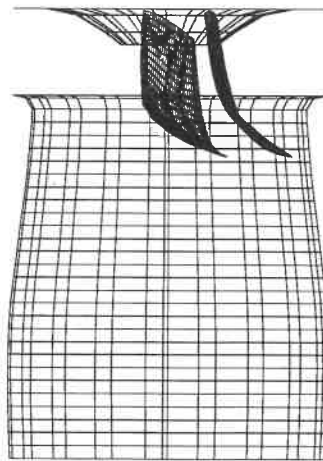
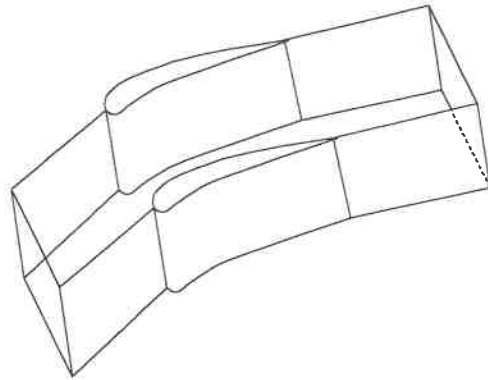Figure 5.1: Francis runner with two blade

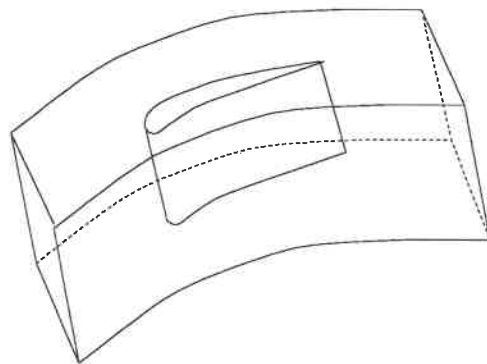Figure 5.2: H type blade-to-blade channel

Figure 5.3: C type blade-to-blade channel

linear interpolation between the hub and shroud. Figure 5.4 shows the generatrix curves that create these supports.
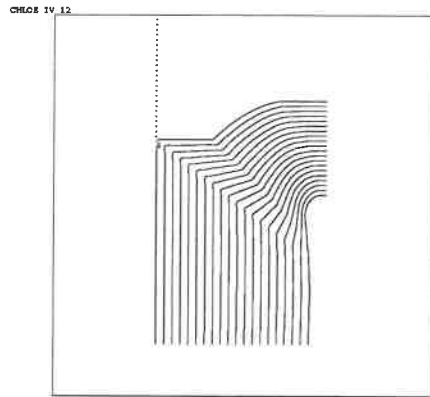
Figure 5.4: Linear interpolation hub and shroud in R-Z plane

5.2   Blade and Profile

The blade is a closed surface bounded by the hub and the shroud. It can be characterized by two sets of curve. The first set traverses from the hub to the shroud, and the second is a set of profiles lying on the support surfaces. These two families of curves form a mesh such that the intersection of one family with the other results in a lattice of nodes. These nodes serve as a basis for defining the blade by means of interpolation as shown in Figure 5.5. This closed surface can be split into two surfaces alone the leading and trailing edges, one is called pressure side, the other is call suction side.

5.2.1   Blade Definition Data

The data that define the blade are two untouched MxN lattices of points in 3D representing the suction and pressure side respectively. These data are obtained by measuring real blades in physical space and do not necessarily correspond to the profiles used to design the blade. An example of two such lattices of points defining
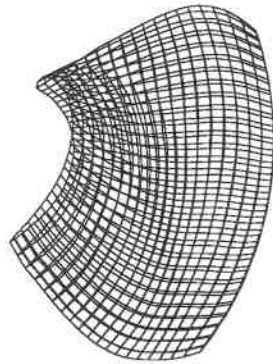
Figure 5.5: Kaplan runner blade

a blade are shown in Figure 5.6. As given, this data does not describe a closed surface and it must be treated to yield such a closed surface to represent the blade for simulating the flow inside runner.

Figure 5.6: Two lattices for representation of the suction and pressure sides

5.2.2   Calculation of the Leading and Trailing Edges

Two distinct BCPS surfaces can be constructed from two lattices using the method developed in section 4.3.3. Because of these two lattices are not connected together, the piecewise cubic curves of those two BCPS are also not connected. Figure 5.7 shows two curves from such surfaces corresponding to the same column number on either side of the blade. We see that the leading and trailing edges of the blade are missing.
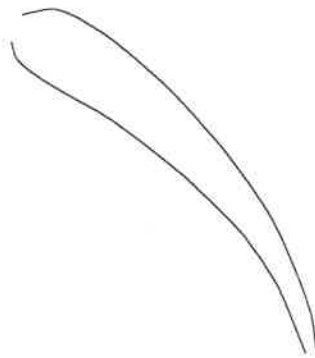

Figure 5.7: Part of profile create by input data

To construct a closed surface to represent the blade, these two surfaces must be connected by recreating the leading and trailing edges. Three possible methods to achieve this construction are consist in closing the gaps by:

1. Connecting two points with a line.

2. Using an elliptic arc to connect the two points.

3. Passing a cubic spline through the two points.

The first method evidently cannot yield good result. As for the second method, it is impossible to find an elliptic arc which will match to the first order continuity the two segments, because the two curves are not plane curves.

The third method will generate the desired result if the boundary conditions are chosen appropriately. By using a cubic spline with tangent vectors at the two boundary points, a curve with geometric continuity can be obtained.

The next step is to identify the leading and trailing edge point. This is arbitrary to some extents and depends on the actual flow field. Because this is a very narrow curve, either the middle point on the spline, or the intersection of the camber line with the spline can be chosen as the leading edge point. The same method is applied to the trailing edge.

Evaluating the curve representing the leading and trailing edges will give two new columns of points that can added to the data to create two M by (N+2) lattices of points which will completely define the blade.

5.2.3   Construction of Blade Profiles

The specification of the profiles for the construction of the blade are such that the first and last profiles do not necessarily lie on the hub and the shroud respectively. There could be small gaps between the blade and the hub, and between the blade and the shroud. However for computational purposes, it is important that the volume bounding the computational domain be closed. Consequently it is necessary to define additional profiles that lie on the hub and shroud. These can be obtained by first extending the blade surface in profile direction, then computing the intersection of the blade surface with the support surface(i.e in this case the hub or shroud). The intersection curve will be served as the profile on the hub and shroud. The

profiles between the first one(on the hub) and the last one(on the shroud) can also be obtained by intersecting blade surface with a group supports between the hub and shroud. This is a very expensive process. The surface-surface intersection algorithm in geometry modeling is carried out using numerical approximation. In this work, the approximation is obtained by intersecting one family of curves of one surface, with the other. This results in a MxN points net lying on the blade surface. The final intersection is obtained by fitting a curve through those points.

By fitting piecewise cubic curves to all rows of the two Mx(N+2) lattice another group of curves is obtained. Those curves will not touch either the hub or the shroud, but can be extended to intersect the hub and the shroud. Linear or quadratic can be used in this case.

The intersection between the surface support and blade can be simplified to a 2D curve-curve intersection if a cylindrical coordinate system is used and the support is a surface of revolution. Which can be written as:

$$R_1(u) = 0, \quad Z_1(u) = 0, \quad \theta = \theta \tag{5.1}$$

The equation of a curve can be written in the form:

$$R_2(v) = 0, \quad Z_2(v) = 0, \quad \theta(v) = 0 \tag{5.2}$$

The intersection between the surface and the curve is formulated as the solution of the following two equations:

$$R_1(u) = R_2(v)$$
$$Z_1(u) = Z_2(v) \tag{5.3}$$

Those are nonlinear equations for which the well known Newton's method can be used. After solving these equations, we can substitute v into Eq 5.2, to obtained the

value (R,T,ψ). The parameter value u of the intersection point along the revolution surface and the angle ψ will later be used to create the blade-to-blade channel. The curve-curve intersection is shown in Figure 5.8.
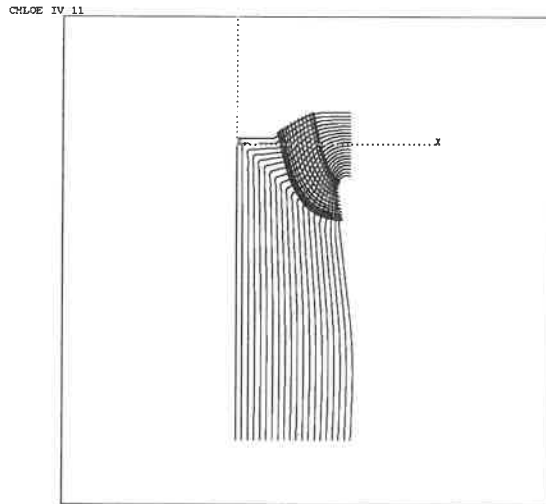

Figure 5.8: Intersection between revolution surface and curve

Figure 5.9 shows a series of profiles created from the input lattice which defines the blade in 3D.

5.3   The Blade-to-Blade Channel

The channel is a six-sided closed volume. It reaches from one blade to the next, from the hub to the shroud and by extending the blade's length from the distributor of the turbine to the outlet plane as shown in Figure 5.10. These extensions result in two surfaces, going from the leading edge of the blade towards the distributor
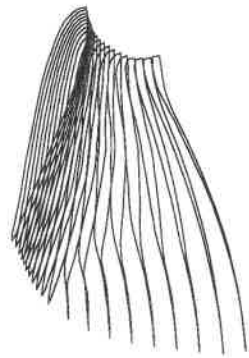
Figure 5.9: A series of profiles to defining a blade

and from trailing edge toward outlet surface. This volume can be divided into three parts: before the blades, between the blades, and after the blades.
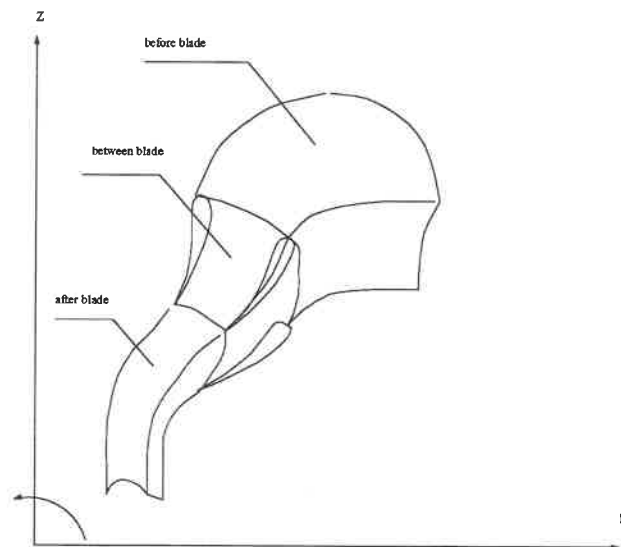

Figure 5.10: Blade-to-blade channel

The part between the blades is bounded by the two blades (the pressure side of one blade and the suction side of the other blade) and the appropriate portions of the hub and shroud. These are the physical boundaries. The angle between two blades is related to the number of blades through:

$$\alpha = \frac{2\pi}{N_{blade}} \tag{5.4}$$

The problem is to create the upstream and downstream section of the domain. A mapping method was used by extending the camber line of each profile on the support. The choice of such a mapping should preserve two parameters, one is the angle ψ, another is the arc length S of the generatrix curve. This will ensure a boundary which will be generally aligned with the flow.

Defining the surface of revolution as:

$$R(u) = 0, \quad Z(u) = 0, \quad \psi = \psi \tag{5.5}$$

a conformal mapping which preserves the arc length S and angle ψ is defined as:

$$dS^2 = dR^2 + dZ^2 \qquad d\psi = d\psi \tag{5.6}$$

Because this transformation is conformal, the result is a mapping of that surface onto a plane $S - \psi$. To preserve the arc length along the generatrix curve of the revolution surface, we set

$$S(u) = \int_{u_0}^{u} dS \qquad \psi = \int_{\psi_0}^{\psi} d\psi \tag{5.7}$$

The profile is obtained by intersecting the blade with the support using a curve surface intersection method. All those intersection points can be mapped from $RZ\psi$ space to $S - \psi$ space by substituting the parameter value u obtained from Eq 5.3 into Eq 5.7.

If we chose the relative arc length, we get:

$$Sr_i(u) = \frac{S_i}{S_{total}} \tag{5.8}$$

Here S_{total} is the total arc length of the generatrix curve. A profile in mapped space can be obtained by interpolating a cubic spline through these points.

Because the blade will control the general flow direction, it is expected that the streamline in the mapping space will roughly match the camber line of the profile at the leading and trailing edge point. Its extrapolation from either the leading or trailing edges is such that the tangent vector of these curves will match the tangent vector of the camber line at these points. Hermite cubic interpolation will be used. It requires position vectors and tangent vectors at each end point. There are two unknown vectors: the position vector and tangent vector at the inlet and the outlet. These vectors can be specified and manipulated by the user to design the channel. Mapping these curves back to $RZ\psi$ space will yield the required 3D periodic boundary on each support. Figure 5.11 illustrates this procedure.

This procedure is repeated on each profile and support and results in as many extension curves as there or profiles in turn these are discretized to create a MxN point lattice from which a bicubic patch composite surface is constructed. Finally, rotating this surface by an angle which is defined by Eq 5.4, another periodic boundary surface is obtained. Those two surfaces together with the hub and the shroud complete the boundary surface of the channel.

5.4   Grid Generation of the Channel

The transfinite interpolation method is used to generate the initial body-fitted grid for the blade-to-blade channel. The grid is then smoothed by a differential
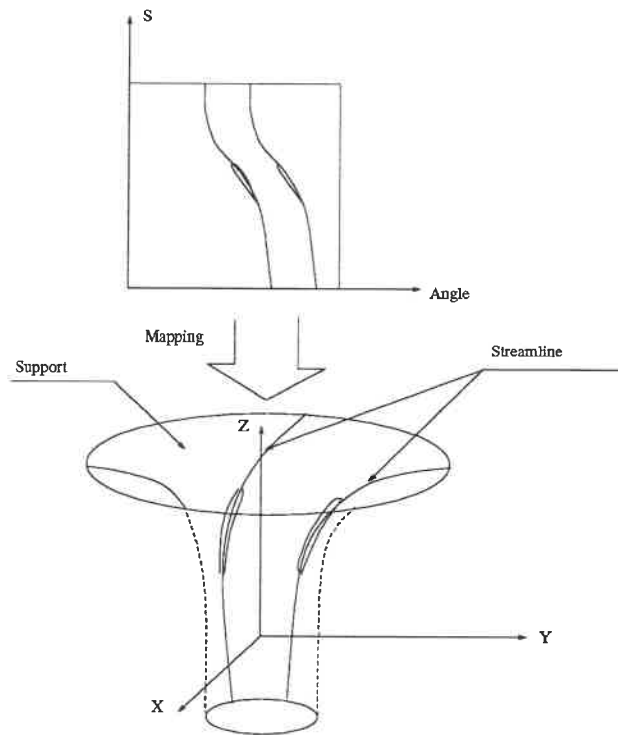
Figure 5.11: Construction of the periodic boundary on the support

equation method. In the mapping space, the blade-to-blade channel is bounded by two profiles and the extrapolated curve which extends from the camber line of those profiles as shown in Figure 5.12.

To generate a grid of good quality for the runner, two curves one joining the leading edges, the other joining the trailing edges are constructed to enforce the distribution of the grid as shown in Figure 5.12. Three Coon's patches are constructed from the inlet to curve1 using the 2 upstream extension curves; from curve1 to curve2 with the pressure side and suction side of the profile; and from curve2 to the outlet using the downstream extension curves. This configuration also allows for controlling the grid near the blade, leading edge and trailing edge.
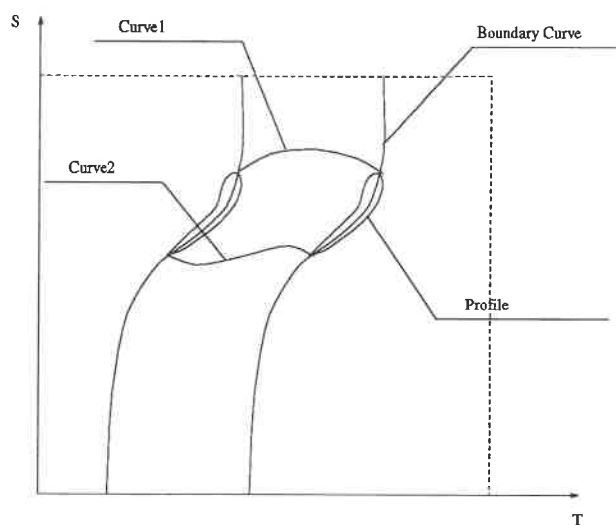
Figure 5.12: Streamline with a profile

5.4.1 Concentration

Grid concentration near the leading edge and trailing edge as well as near the
blade is usually required. This is achieved by controlling the distribution of point
on the curve with a law defined by a NURBS curve. This law can be represented
as a curve $L(u)$ between 0.0, 0.0) and (1.0, 1.0). A uniform distribution can be
represented by a straight line. When concentrating near the beginning point, the
straight line become a curve towarded to point (0.0, 1.0) with a small slope near
(0,0). When concentrating towards the end, the curve towards to (1.0, 0.0) with a
slope greater than 1.0 near (0,0) as shown in Figure 5.13.

This can be achieved by using an order 2 NURBS with three control points
$(0.0, 0.0, 1.0), (1.0, 0.0, w), (0.0, 1.0, 1.0) or (0.0, 0.0, 1.0), (0.0, 1.0, w), (1.0, 1.0, 1.0)$.
The shape of the curve can be control by changing the value of w from 0.0001 to 1.0.
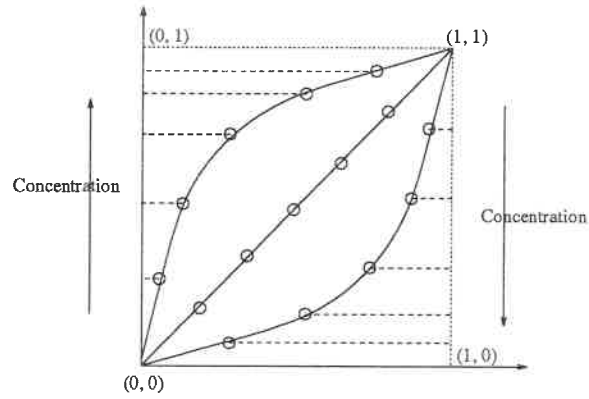
Figure 5.13: Concentrate law of points

When the weight approaches 0.0, the NURBS become a straight line, there is no concentration, w approaches 1.0, very good concentration is obtained. Figure 5.14 shows the control NURBS weight from 0.0001 to 1.0 with step 0.1.

Figure 5.14: Concentrate at one end and at both ends

The Figure 5.15 is the grid in mapping space with concentration. The Figure 5.16

is the grid on the support with concentration.
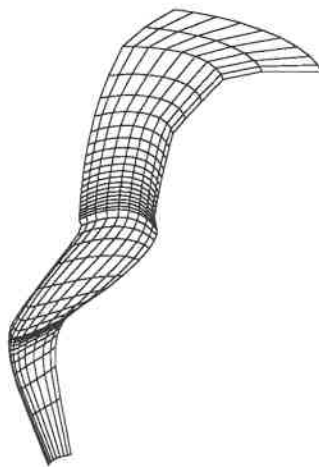

Figure 5.15: Grid in parameter space

Figure 5.16: Grid on the surface of revolution

Once the grid on each surface of revolution is generated, grids on each supports are stack to produce a 3-D grid. Figure 5.17 shows grid of a blade-to-blade channel.
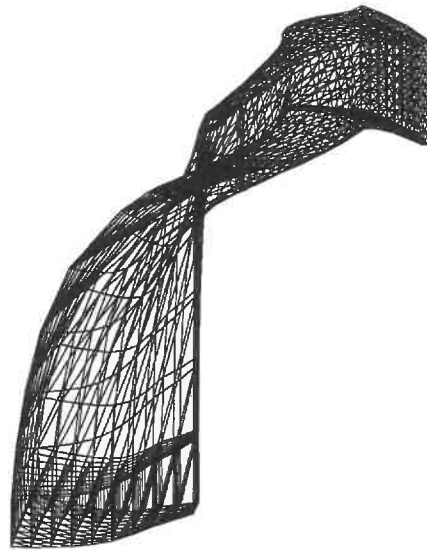

Figure 5.17: 3D Grid of the complete blade-to-blade channel

Chapter 6

Diffuser

There are two different types of diffusers: with and without a pillar at the outlet. Figure 6.1 shows the sections of the diffuser with a pillar. The geometry of both types is defined by two series of sections lying on plane supports distributed along a spine curve. The shape of the sections changes gradually from a half circle at the inlet to a rectangle at the outlet. The difference between those types is that the two sections, in the case without a pillar, will share a common edge, whereas in the case with a pillar, at some intermediate support, those two sections will be separated by the pillar. Both types are treated in the same way, i.e it is assumed the pillar is alway present, either real or virtual. For the case without the pillar, the thickness is zero.
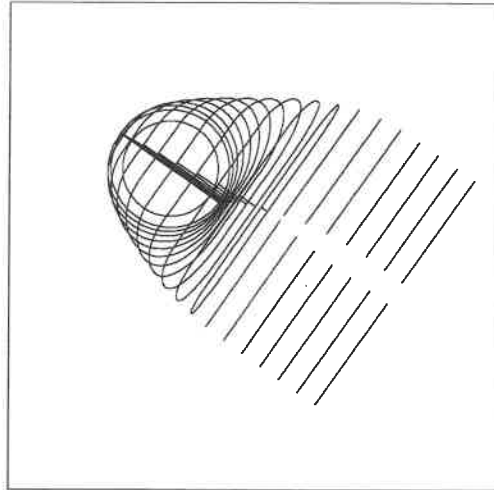
Figure 6.1: Section of diffuser with pillar

6.1 Geometry interpreter and description language

There are different methods to describe a 3D object for a computer system. The most common method in CAD/CAM system is to use a graphical interface with menu, icon, keyboard, mouse etc... to interactively create a 3D object. This method presents difficult when dealing with 3D object on a 2D screen.

In the present work, the geometry objects are a group of objects with a similar geometrical characters. They are more easily defined and described by a group of operations. Instead using a graphical user interface, a geometry interpreter(geometry compiler) was designed to create 3D object from a geometry description language(GDL). Like other computer languages, this requires a compiler, which acts as an interpretor between the GDL and the computer. Besides defining the geometry, this language

can also be used to describe specific manipulations or operations. The advantage of this method is that a user can define the geometry with keywords(token) having a close to mathematical terms. The relation between geometry compiler, geometry library, YACC (a UNIX compiler building tool), and the user is shown in the Figure 6.2. The overall system has 'three' key components. At the root there is the geometry library which created the model. The compiler provides the links between the source description file and this library. For the purpose of display, a graphical image will be generated from the geometry object.
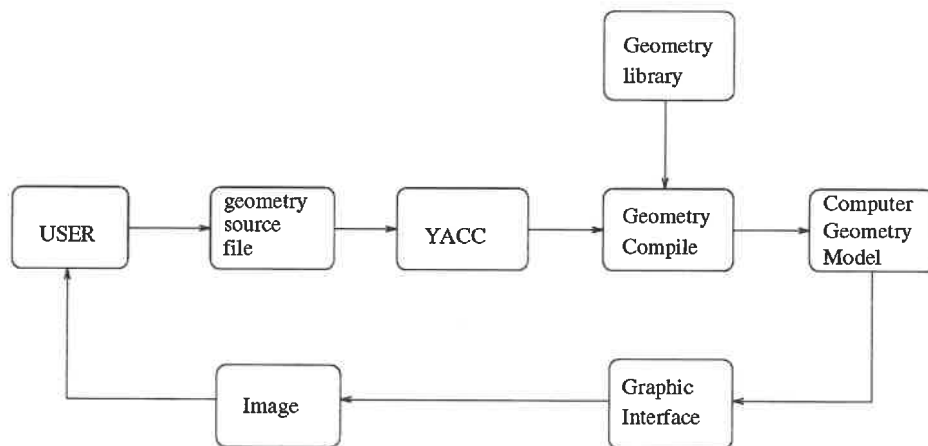
Figure 6.2: Relationship between language compiler and others

Like any computer language, the GDL is composed of grammar rules, tokens, parameters, and delimiters. A sentence must be formed from token, parameter, delimiter. The token can be mathematical terms. A parameter is either a integer or a float number used to define the identity of the geometric entity or quantity. Delimiter include commas(,), semicolon(;) and left or right parenthesis. Here are

some examples of the GDL.

1. A point:

    point 1 {0.0, 0.0, 0.0}

2. A line defined by two point:

    line2 {*point* 1, *point* 2}

3. A composite curve defined by two lines and one arc:

    curve 1 {*line* 1, *line* 2, *arc* 3}

4. A plane support defined by one original vector and two direction vectors:

    plane 1 {*point* 11, *point* 12, *point* 13}

5. A section in three-dimensional space is defined as:

    section 1 {*curve* 21, *curve* 22..... *curve* 29}

6. A diffuser is defined by a series of section in three-dimensional space:

    diffuser 1

    {{section 1, plane 1},

    {section 2, plane 2},

    {section 3, plane 3},

    {section 4, plane 4},

    ....

    ....

    {section n-1, plane n-1},

    {section n, plane n} }

A more complete example is shown in appendix A for a diffuser. The geometry interpreter provides tools for imposing structure on the input to a GDL. When the

GDL source is scanned by the input scanner, a lexical analyzer picks up items from the input stream. When a grammar rule corresponding to a token has recognized, then an action (a C function) is invoked. Actions are fragments of C language code. They return values which can be used by other actions. The geometry compiler is implemented with UNIX's YACC and LEX utility. It has basic error handling abilities.

6.2   Definition of the Diffuser Geometry

6.2.1   Mapping of the Sections Curve to 3D Space

The section is defined by a planar support and a 2D closed composite curve in the parameter space. This curve may consists of elliptic arcs and line segments. The section in three-dimension space is obtained by mapping the 2D parameter composite curve into 3D space using a 4x4 homogeneous transformation.
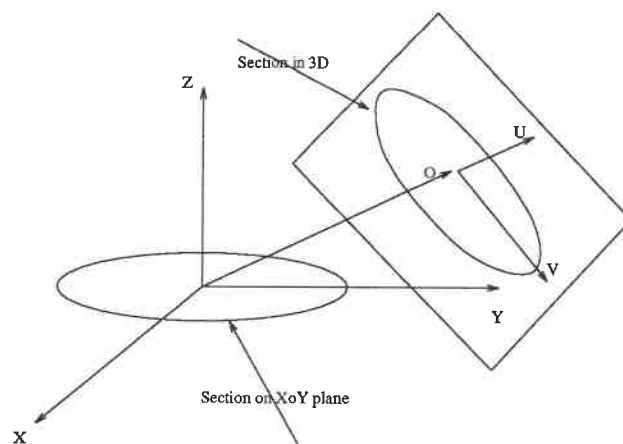

Figure 6.3: Transformation of the section curve to 3D space

The plane support is defined by its origin, O, and two vectors, U, V lying within the plane:

$$
\begin{aligned}
O &= [X_0, Y_0, Z_0] \\
U &= [X_1, Y_1, Z_1] \\
V &= [X_2, Y_2, Z_2]
\end{aligned}
\tag{6.1}
$$

A transformation T can be obtained easily from those three vectors.

$$
T = \begin{bmatrix}
m_{11} & m_{12} & m_{13} & x_0 \\
m_{21} & m_{22} & m_{23} & y_0 \\
m_{31} & m_{32} & m_{33} & z_0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\tag{6.2}
$$

Applying this transformation to the composite 2D curve represented by:

$$
C(u) = \sum_{i=1}^{n} B_i^k(u) P_i
\tag{6.3}
$$

yields,

$$
\begin{aligned}
C^T(u) &= \sum_{i=1}^{n} B_i^k(u) P_i \\
&= \sum_{i=1}^{n} B_i^k(u)[T * P_i]
\end{aligned}
\tag{6.4}
$$

This equation illustrates that the transformation of a NURBS curve is equivalent to applying it to its control points.

6.2.2 Surface definition of the Diffuser

The surface of the diffuser can be generated by a skinning operation, that is the surface swept by a closed section as it travels along a spine. Consider the case

with a pillar. This section starts initially as two half circular sections in contact along a diameter. As the section sweeps through space, the half circles shape, and eventually separate into two distinct sections, thus creating a fork-like object.

Because of the sharp corners appearing on the sections at the outlet, it is impossible to model the surface of the diffuser with a single bicubic composite surface. Therefore in the region the diffuser surface will be represented by four bicubic surfaces.

A typical section is composed of four segments and two elliptic arcs, as shows in Figure 6.4. However at the inlet it is composed of only two elliptic arc and one segment, while at the outlet, the elliptic arcs degenerate to points, so that the section is then composed of four segments. To define the four bicubic surfaces, the section must be divided into four curves. Because the elliptic arcs degenerate to points, the best location to spite the curve is the middle of the arc.

Lattices are required to create the BCSP. They are obtained by discretizing the four groups of curves with respect to the arc length. From these, bicubic composite surfaces can be constructed as described in section 3.7.2. two neighboring surfaces will share a common cubic spline on the boundary. This will ensure that the node on the two side will match. This result is shown in Figure 6.4 for a diffuser without pillar. Figure 6.5 shows a similar construction for a diffuser with pillar.

6.3   Grid Generation of the Diffuser

As discussed in chapter 3, we will now describe a multi-block structured grid technique for the diffuser. This process consists of two steps: first the surface(or shell) is discretized, then transfinite interpolation is applied to generate the interior

grid points. The resulting grid can be further improved by smoothing using an elliptic differential equation method.

6.3.1 Surface Mesh.

Each of the four bicubic surfaces is represented by a patch defined as:

$$P = S(u,v) \quad u_0 \le u \le u_1 \quad v_0 \le v \le v_1 \tag{6.5}$$

v is along flow direction u is across. On Each patch, a number of curves corresponding to fixed values u_i of the parameter u are generated. These curves are piecewise
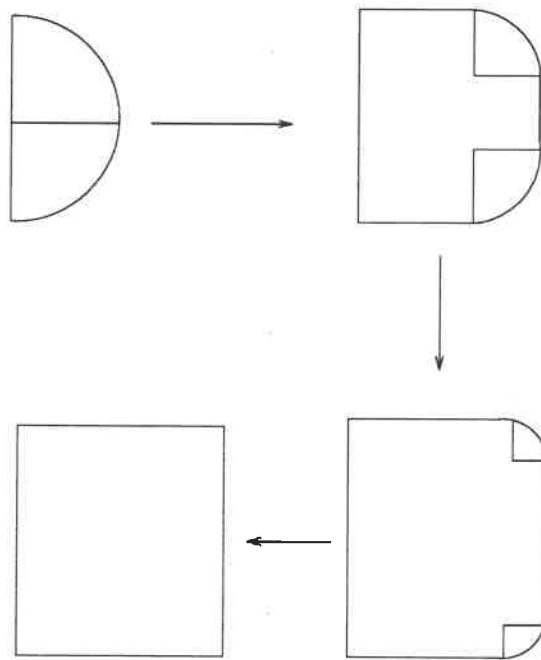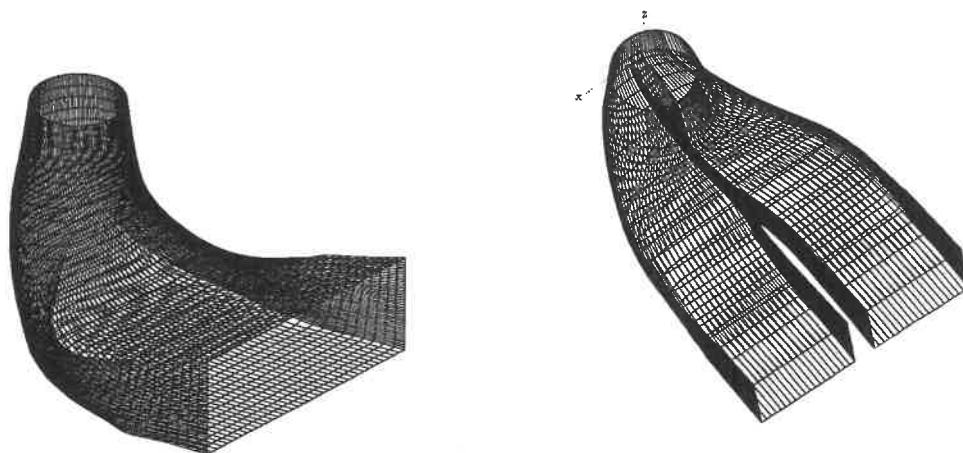

Figure 6.4: Different section of diffuser

Figure 6.5: Shell of two types diffuser

cubic splines $C(v)$ represented as:

$$C(v) = S(u_i, v) \tag{6.6}$$

A group of new sections(composed of four curves) alone the spine are obtained by evaluating these four surface with u_i. Discretizing these new sections with fixed values v_i of parameter v, yields a net or surface grid as shown in Figure 6.6.

The grid point concentration near the pillar is obtained by controlling the distribution of u. Figure 6.6 shows the new sections generated with concentration near the pillar. It should be noted that the sections obtained from $u = $ constant do not correspond exactly to the original defining sections from blue prints. Furthermore, the number and distribution of the sections for grid generation purposes are different from the original sections used for defining geometry. Because these sections are generated from surface, they are not even planar sections.
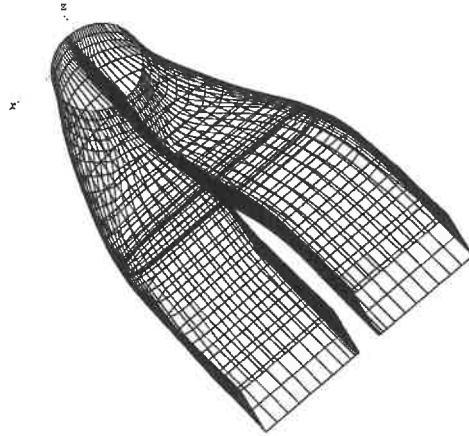
Figure 6.6: New section of diffuser with pillar

6.3.2 Multi-Block Volume Mesh

A typical mesh for a closed section using the standard elliptic differential technique is shown in Figure 6.7. Inspection shows two drawbacks. First the corner is not well represented and second it is difficult to generate a concentration toward the solid surface with this configuration. The multiblock method can solve these problems by dividing the domain into a number of simple zones, and then generating a mesh within each of them.

There are many different ways of subdividing a section, and Figure 6.8 shows a particularly appropriate way for CFD calculations. The configuration is controlled by specifying the ratio between the area of the central rectangular zone and the area of the complete section. The bilinear Coon's patch method is used to generate the grid inside each zone.
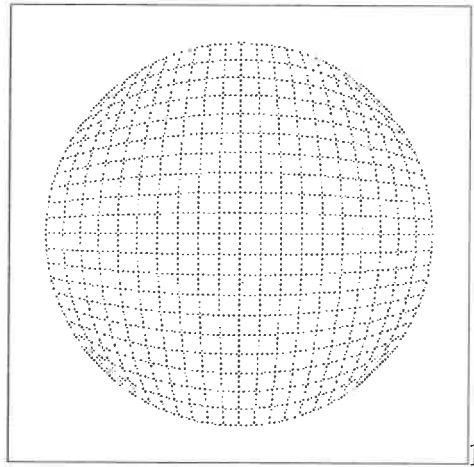
Figure 6.7: A typical mesh of elliptic technique

6.3.3   Grid Control

A quality grid with good concentration can improve the results from numerical computations. For the present components this is required at two places: near
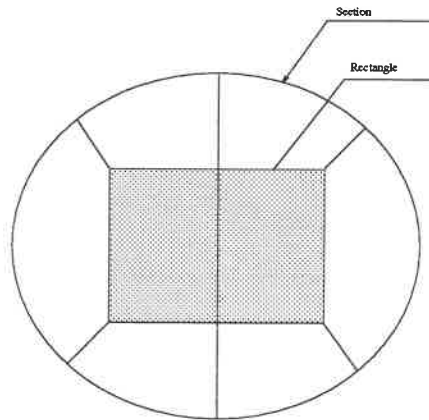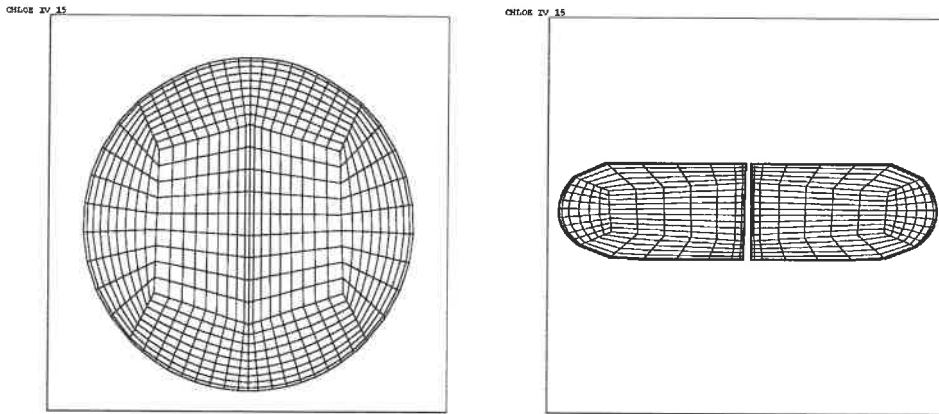

Figure 6.8: Divide section

Figure 6.9: Grid of inlet and middle section

the pillar, and near the solid boundary. Concentration near the pillar has already been mentioned in the previous section. The concentration toward the boundary is applied by controlling the distribution of the parameters when discretizing the bilinear Coon's patch of each zone as described in chapter 4. For the multiblock grid, it is important to match the nodes on the internal boundaries. The bilinear Coon's patch interpolation will ensure this if the same concentration is applied to adjoining zones. Figures 6.9 and 6.10 show the grid at three different sections with concentration towards the boundary.

These figures show that bilinear interpolation can generate good grids for this type of sections, specially near the boundary.

6.4 Generating an Unstructured grid

An unstructured grid includes two parts: the coordinates of each node, and the connectivity between them. From the previous section, the nodes on internal
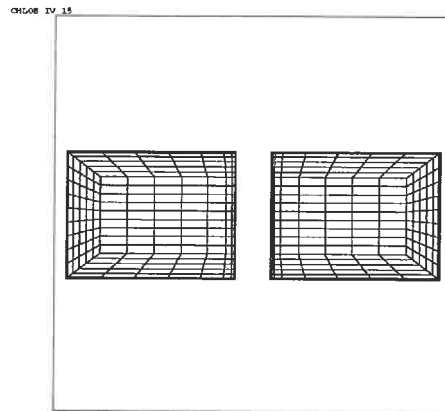
Figure 6.10: Grid of outlet section

boundaries are duplicated as each zone has its own structure. These nodes must be merged. The Figure 6.11 shows an example of nodes that need to be merged alone internal boundaries.
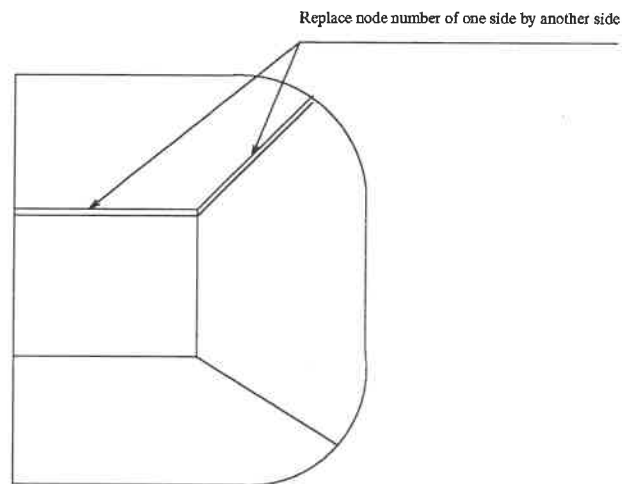

Figure 6.11: Merge the node on internal boundary

The diffusers with and without pillar are treated in the same way. For diffuser

without pillar, a zero thickness pillar is given. The nodes on this zero thickness pillar are duplicated and then need to be merged. The diffuser with pillar consists of two volumes sharing a common boundary surface from inlet to pillar. The nodes before the pillar are merged and the connectivity is established. The Figure 6.12 shows a unstructured grid for a diffuser with pillar.
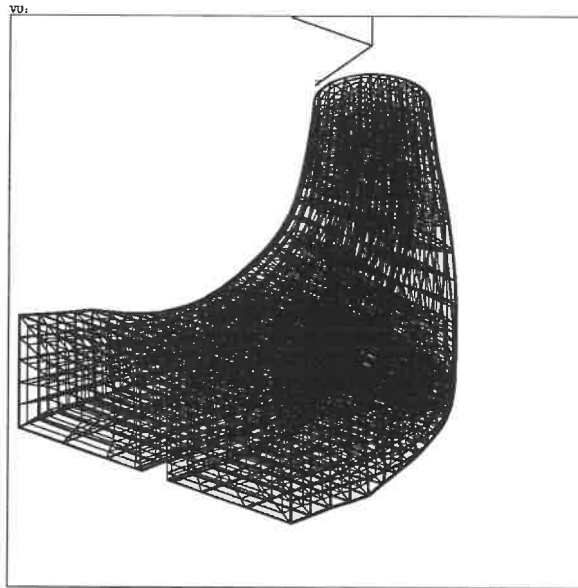

Figure 6.12: Grid of diffuser with pillar

Chapter 7

Spiral Casing

The spiral casing consists of a straight cylindrical tube connected to a spiral tube of decreasing cross-section. It is described by a series of sections lying on planar supports distributed along a spine curve. The spine curve is a straight line at beginning, it then turns into a spiral curve. The supports are positioned in a way such that their normal vectors are tangent to the spine. At the inlet, they are parallel to each other, in the spiral sections they are oriented along radial planes. Because of symmetry, only half of the spiral casing need to be modeled. The figure 7.1 shows the sections distributed along the spine.

The method used to create the spiral casing is the same as for the diffuser: it is based on a geometric description language and a geometry interpreter. The sections along the spine curve are obtained by a transformation of a basic section in 2D space into a three-dimensional space on a given support as shown in Figuer 7.2.
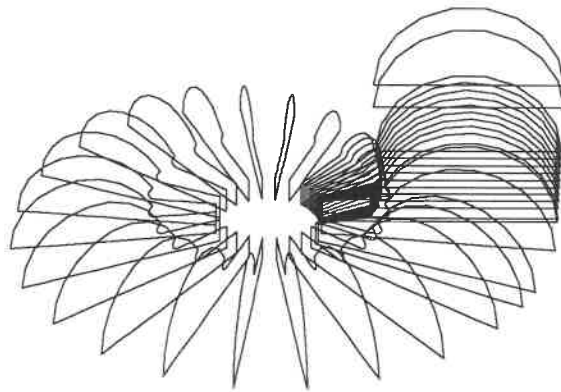
Figure 7.1: Sections of spiral casing along spine

7.1 Geometry of the Spiral Casing

Although similar in principle to the diffuser, the spiral casing presents several important differences at the junction of the surface. To simplify the problem, the spiral casing is broken into two parts: a spiral pipe of circular section and a section ring of rectangular section as shown in Figure 7.3.

A typical section is shown in Figure 7.4. The square section touches one side of the spiral section. The difficulty is to model the spiral part after it has gone around 360° and must be joined to the inlet. Because in this region, the sections of the spiral casing touches a stay vane which controls the direction of the flows from both side Figure 7.5. The problem is like to finding the intersection of the stay vane with two portions of the spiral casing. The sections are cut by the stay vane. Figure 7.5
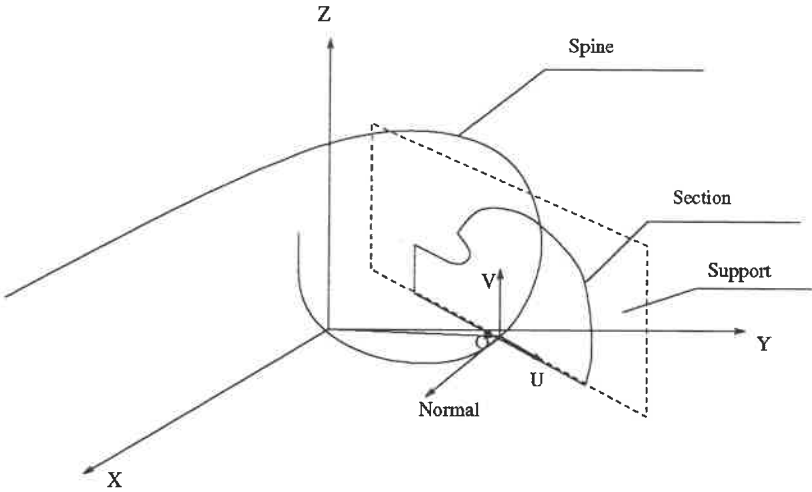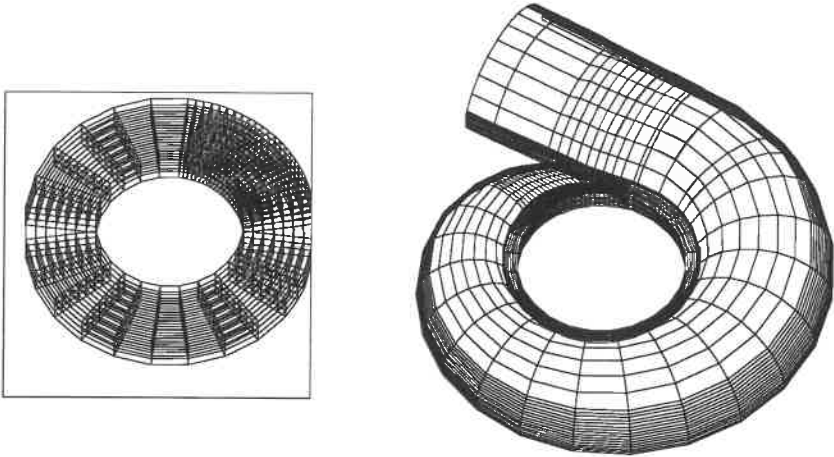
Figure 7.2: Transformation of the section

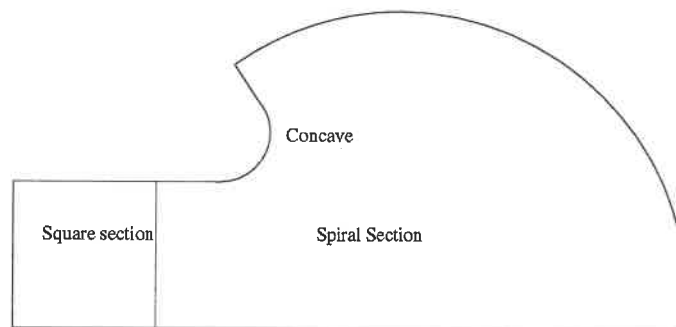Figure 7.3: Two parts of spiral casing

shows this region.
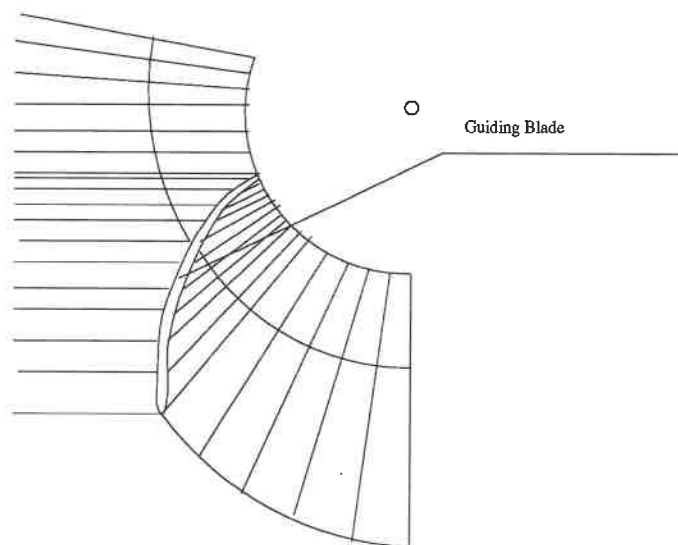
Figure 7.4: Typical section of spiral casing

Figure 7.5: Project of junction part

7.1.1 Sections

The evolution of the sections from the inlet to the stay-vane is shown in Figure 7.6. It starts with a full circle, one part is cut by the stay vane, the section develops into complex shapes. At the end, the section will again touch the stay vane, but from the other side. At the very end of the casing, only one vertical line is
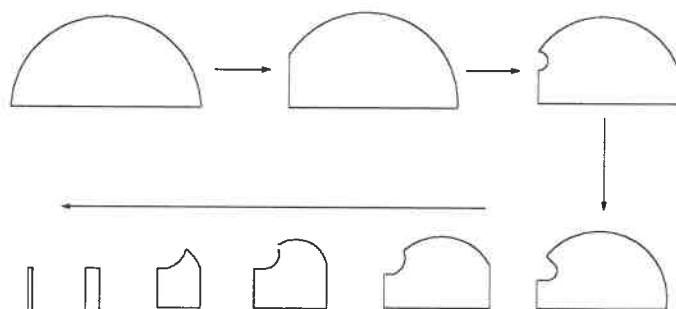
Figure 7.6: Evolution of original sections along the spine curve

left. This will generate a grid with a singularity at this location. The problem can be overcome by defining the sections. To prevent the singularity, the new sections are constructed so they all have the same topology. They are built as a blending between the initial first section and the back side of the stay-vane. These have the same topology from the beginning to the end, that is, they are always composed of five curves. It is noted that the new sections are not plane sections, because they do not lie on a planar support. Figure 7.7 shows these new supports. The new sections should lie on the surface of the casing which is defined by the original sections. This surface is a ruled surface through each of the original sections and is cut by the stay vane as shown in Figure 7.8.

The new sections are built this way: first, a ruled surface through each of the original sections constructed, then this surface is discretized resulting in a lattice of points. Finally, a series of cubic spline curves are obtained by interpolating each column of the lattice points. Because of the presence of sharp corners, five ruled surface are created. Therefore, five curves of a composite curve are obtained from those surfaces. Part of a ruled surface between two sections is shown in Figure 7.9. Figure 7.10 shows the new sections lying on the original ruled surface. The new shell of the spiral casing which is defined by combining these new sections is shown
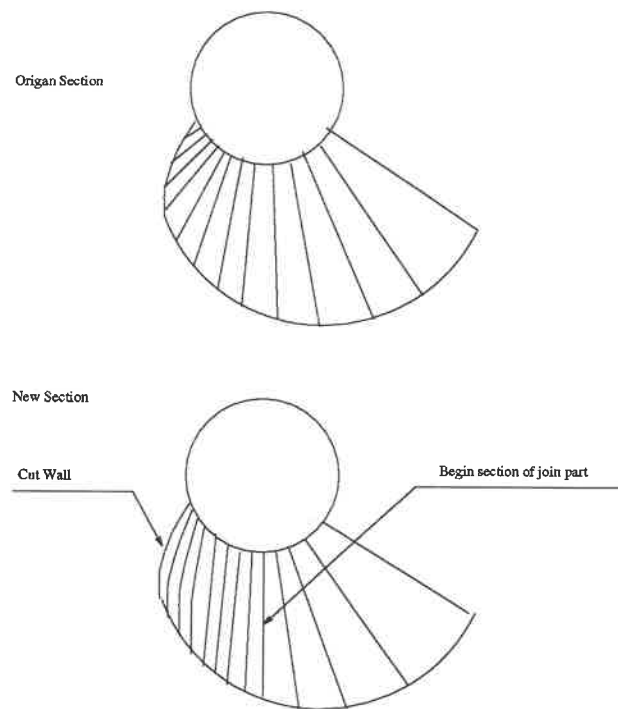
Figure 7.7: New section near junction part

in Figure 7.11.

7.2 Grid Generation of the Spiral Casing

The multiblock method will be used to generate the grid of the spiral casing. The method to generate the nodes inside the spiral casing is as the same as the one used in the diffuser. Each section is divided into four zones as shown in Figure 7.12. The concentration is applied near the boundary and near the small concave. The nodes on the internal boundary between the zones are eliminated as described in the previous chapter.

The section ring has a section which is a simple square attached to the spiral
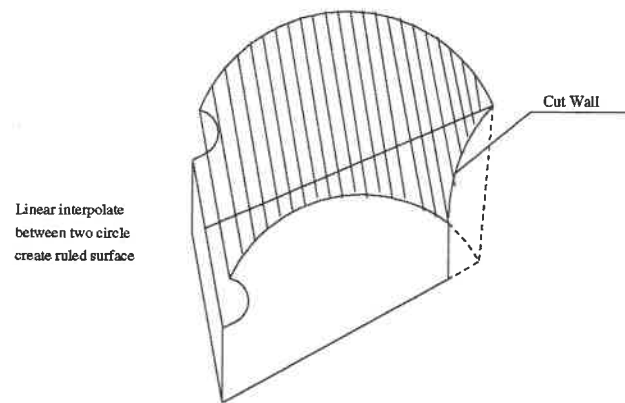
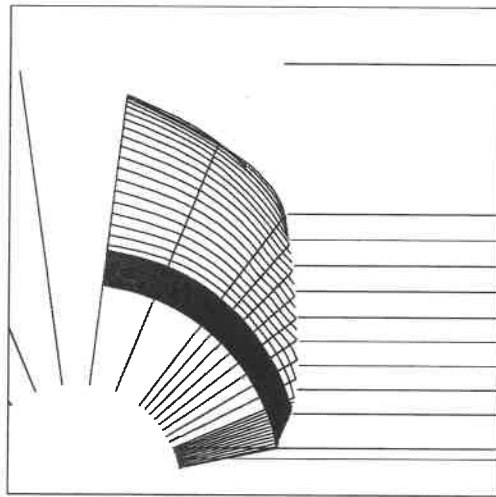Figure 7.8: Intersection of the spiral casing with stay vane

Figure 7.9: Part of the Ruled surface between two sections

section. This is easily meshed, note that the nodes on the boundary between the section ring and spiral casing also need to be eliminated. Figure 7.13 shows a grid for a typical spiral casing.
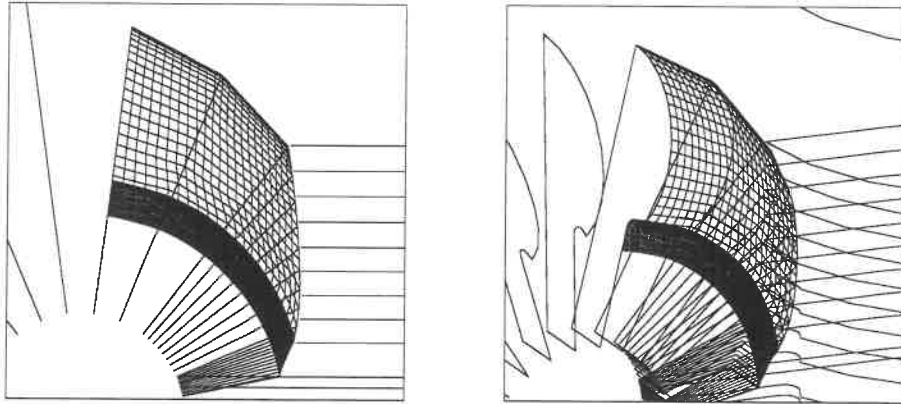
Figure 7.10: New section of join part

CHLOE IV
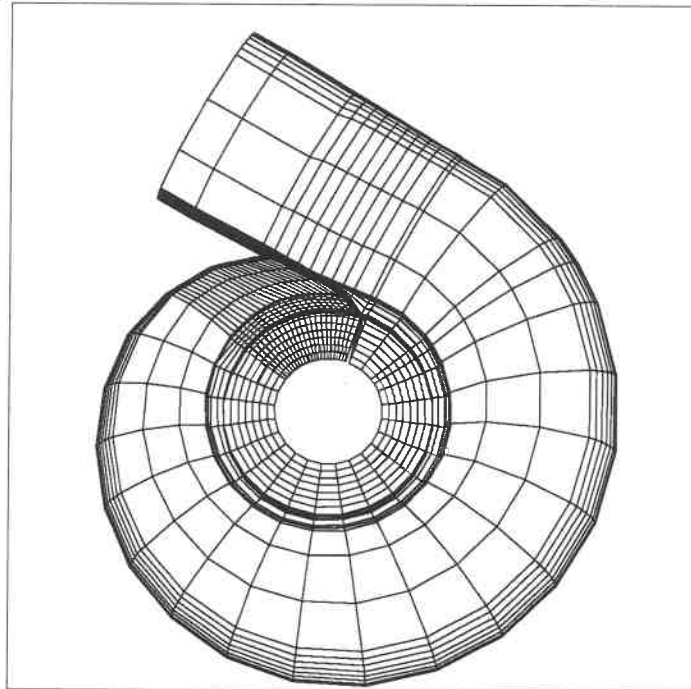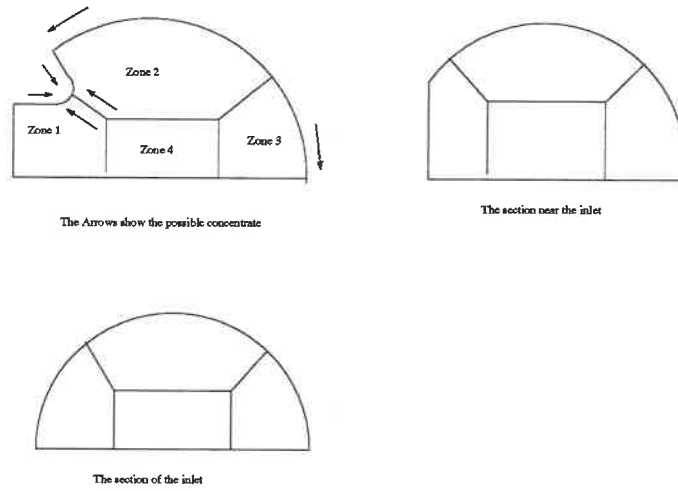
Figure 7.11: New shell after redefined section

Zone 2

Zone 1

Zone 4

Zone 3

The Arrows show the possible concentrate

The section near the inlet

The section of the inlet

Figure 7.12: Zone of section
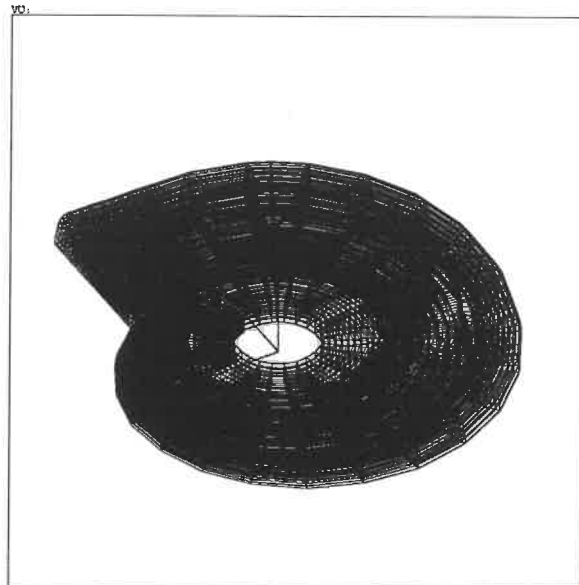

Figure 7.13: Grid of Spiral Casing

Chapter 8

Conclusion and Discussion

In this work, a computer-aided geometry and grid-generating software package for hydraulic turbine has been presented. The methods described in this research have achieved the goal of automaticaly generating grids for the computer simulation of the flow inside components of the hydraulic machine. The software includes two major parts. One is a 3D geometry language and library for curves and surface representation and manipulation. The other is an automatic grid generation tool for the hydraulic turbine components. The described software has been implemented, tested and applied to the modeling of practical geometries to illustrate the package's applicability for a variety of models ranging from Francis and Kaplan turbines to bulb turbines.

References

1. **CAMARERO R., OZELL B., GARON A., REGGIO M. and TREPANTIER J.Y.** "Grid generation with Application to turbomachinery", "Short Course", CERCA 1992

2. **DE BOOR C.**, A practical Guide to Splines, Springer-Verlag, 1978.

3. **FARIN G.**, Curves and Surfaces for Computer Aided Geometric Design, Academic Press, Inc 1988

4. **FARIN G.**, "From Conics to NURBS: A Tutorial and Survey", IEEE Computer Graphics & Applications 1993

5. **KNUPP P. and STEINBERG S.**, Fundamentals of Grid Generation, CRC Press, Inc 1993

6. **MCMAHON C. and BROWNE J.** CAD/CAM from principles to practice, Addison & Wesley, 1993

7. **MORTENSON M.E.**, Geometric Modeling, John Wiley & Sons, Inc, New York, 1985

8. **OZELL B., CAMARERO R.**, "Between The Blade: Geometric and computational modeling of turbomachinery", CIME Vol. 4, No.4, pp 28-35., 1986

9. **OZELL B, CAMARERO R.** "3-D Geometric and computational modeling of turbomachines". SYMPOSIUM, 1986

10. **PIEGL L. and TILLER W.**, "Curve and surface constructions using rational B-splines", Computer-Aided Design, volume 19, number 9, pp 485-498

11. **ROGERS D.F. and ADAMS J.A.**, Mathematical Elements for Computer Graphics, McGraw-Hill Inc, New York

12. **SARRAGA R.F.**, "Computer Modeling of Surfaces with Arbitrary Shapes", IEEE Computer Graphics & Applications, 1993

13. **THOMPSON J.F.**, Numerical Grid Generation, North-Holland, New York, 1984

14. **TILLER W.**, "Rational B-Splines for Curve and Surface Representation", IEEE CG&A, septemper 1983, pp 61-69

15. **YAMAGUCHI F.**, Curves and Surfaces for Computer-Aided Geometric Design, Springer-Verlag, 1988

Appendix A:

Geometry description of diffuser

//— SECTION 1 —

SUPPORT PLAN 1

(

POINT_ORIG (0.000000,0.501705,-0.375000)

POINT_U (1.000000,0.501705,-0.375000)

POINT_V (0.000000,1.505115,-0.375000)

)

POINT 1 (0.000000,-0.501704)

POINT 2 (0.501704,-0.501704)

POINT 3 (0.000000,0.000000)

POINT 4 (-0.501704,-0.501704)

POINT 5 (0.000000,-1.003409)

COURBE ARC_ELLIPSE 1 (// CERCLE

POINT_CENTRE 1

POINT_INITIAL 2

POINT_FINAL 3)

COURBE POLYLINE 2 (POINTS 3,3)

COURBE ARC_ELLIPSE 3 (// CERCLE

POINT_CENTRE 1

POINT_INITIAL 3

POINT_FINAL 4)

COURBE POLYLINE 4 (POINTS 4,4)

COURBE ARC_ELLIPSE 5 (// CERCLE

POINT_CENTRE 1

POINT_INITIAL 4

POINT_FINAL 5)

COURBE POLYLINE 6 (POINTS 5,5)

COURBE ARC_ELLIPSE 7 (// CERCLE

POINT_CENTRE 1

POINT_INITIAL 5

POINT_FINAL 2)

COURBE POLYLINE 8 (POINTS 2,2)

SECTION 1 // TYPE=1 (

COURBE S 1,2,3,4,5,6,7,8)

// -- SECTION 2 --

SUPPORT PLAN 2 (

POINT_ORIG (0.000000,0.547275,-0.900000)

POINT_U (1.000000,0.547275,-0.900000)

POINT_V (0.000000,1.641825,-0.900000))

POINT 6 (0.000000,-0.547275)

POINT 7 (0.547275,-0.547275)

POINT 8 (0.000000,0.000000)

POINT 9 (-0.547275,-0.547275)

POINT 10 (0.000000,-1.094550)

COURBE ARC_ELLIPSE 9 (// CERCLE

POINT_CENTRE 6

POINT_INITIAL 7

POINT_FINAL 8)

COURBE POLYLINE 10 (POINTS 8,8)

COURBE ARC_ELLIPSE 11 (// CERCLE

POINT_CENTRE 6

POINT_INITIAL 8

POINT_FINAL 9)

COURBE POLYLINE 12 (POINTS 9,9)

COURBE ARC_ELLIPSE 13 (// CERCLE

POINT_CENTRE 6

POINT_INITIAL 9

POINT_FINAL 10)

COURBE POLYLINE 14 (POINTS 10,10)

COURBE ARC_ELLIPSE 15 (// CERCLE

POINT_CENTRE 6

POINT_INITIAL 10

POINT_FINAL 7)

COURBE POLYLINE 16 (POINTS 7,7)

SECTION 2 // TYPE=1 (

COURBE S 9,10,11,12,13,14,15,16)

// — SECTION 3 —

SUPPORT PLAN 3 (

POINT_ORIG (0.000000,0.625000,-1.795455)

POINT_U (1.000000,0.625000,-1.795455)

POINT_V (0.000000,1.875000,-1.795455))

....

....

....

....

//— SECTION 17 —

SUPPORT PLAN 17 (

POINT_ORIG (0.000000,2.361818,-2.405734)

POINT_U (1.000000,2.361818,-2.405734)

POINT_V (0.000000,2.361818,-1.629594))

POINT 148 (0.848863,0.000000)

POINT 149 (-0.848863,0.000000)

POINT 150 (-0.848863,-0.776136)

POINT 151 (0.848863,-0.776136)

COURBE ARC_ELLIPSE 129 (// CERCLE

POINT_CENTRE 148

POINT_INITIAL 148

POINT_FINAL 148)

COURBE POLYLINE 130 (POINTS 148,149)

COURBE ARC_ELLIPSE 131 (// CERCLE

POINT_CENTRE 149

POINT_INITIAL 149

POINT_FINAL 149)

COURBE POLYLINE 132 (POINTS 149,150)

COURBE ARC_ELLIPSE 133 (// CERCLE

POINT_CENTRE 150

POINT_INITIAL 150

POINT_FINAL 150)

COURBE POLYLINE 134 (POINTS 150,151)

COURBE ARC_ELLIPSE 135 (// CERCLE

POINT_CENTRE 151

POINT_INITIAL 151

POINT_FINAL 151)

COURBE POLYLINE 136 (POINTS 151,148)

SECTION 17 // TYPE=6 (

COURBE S 129,130,131,132,133,134,135,136)

// — SECTION 18 —

SUPPORT PLAN 18 (

POINT_ORIG (0.000000,7.272727,-0.877273)

POINT_U (1.000000,7.272727,-0.877273)

POINT_V (0.000000,7.272727,0.881818))

POINT 152 (1.065909,0.000000)

POINT 153 (-1.065909,0.000000)

POINT 154 (-1.065909,-1.759091)

POINT 155 (1.065909,-1.759091)

COURBE ARC_ELLIPSE 137 (// CERCLE

POINT_CENTRE 152

POINT_INITIAL 152

POINT_FINAL 152)

COURBE POLYLINE 138 (POINTS 152,153)

COURBE ARC_ELLIPSE 139 (// CERCLE

POINT_CENTRE 153

POINT_INITIAL 153

POINT_FINAL 153)

COURBE POLYLINE 140 (POINTS 153,154)

COURBE ARC_ELLIPSE 141 (// CERCLE

POINT_CENTRE 154

POINT_INITIAL 154

POINT_FINAL 154)

COURBE POLYLINE 142 (POINTS 154,155)

COURBE ARC_ELLIPSE 143 (// CERCLE

POINT_CENTRE 155

POINT_INITIAL 155

POINT_FINAL 155)

COURBE POLYLINE 144 (POINTS 155,152)

SECTION 18 // TYPE=6 (

COURBE S 137,138,139,140,141,142,143,144)

ASPIRATEUR (

SECTION 1, SUPPORT 1

SECTION 2, SUPPORT 2

SECTION 3, SUPPORT 3

SECTION 4, SUPPORT 4

SECTION 5, SUPPORT 5

SECTION 6, SUPPORT 6

SECTION 7, SUPPORT 7

SECTION 8, SUPPORT 8

SECTION 9, SUPPORT 9

SECTION 10, SUPPORT 10

SECTION 11, SUPPORT 11

SECTION 12, SUPPORT 12

SECTION 13, SUPPORT 13

SECTION 14, SUPPORT 14

SECTION 15, SUPPORT 15

SECTION 16, SUPPORT 16

SECTION 17, SUPPORT 17

SECTION 18, SUPPORT 18)