TWO-DIMENSION TO THREE-DIMENSION CONVERSION USING SLICED SECTIONS

A thesis presented to
The faculty of the College of Engineering and Tech.
Ohio University

In Partial Fulfillment of the requirement for the degree
Master of Science

By
ARDESHIR SHOLAPURWALLA
AUGUST, 1991

OHIO UNIVERSITY LIBRARY
ACKNOWLEDGEMENTS

The author expresses his heartfelt gratitude to Dr. Jay Gunasekera for all the guidance and assistance during this research. The author also thanks Mr. Thiru Veerabadran, Universal Energy Systems, Inc. and Mr. Bhavin Mehta for the assistance and advice provided during the course of this research.

A special thanks to Universal Energy Systems, Inc. and to the Wright Paterson Air Force Base for the financial and technical assistance provided for this research.
# TABLE OF CONTENTS

1. **INTRODUCTION** ...........................................................................................................01  
   1.1 NEED TO MODEL .................................................................................................02  
   1.2 NEED FOR INTERACTIVE COMPUTER GRAPHICS ...........................................03

2. **GENERAL OVERVIEW** ..............................................................................................04

3. **ALGORITHM FOR 3-D MODELING OF 2-D SLICED SECTIONS**  
   3.1 CASE1: THE GEOMETRY IS A PROJECTED SOLID ............................................06  
   3.2 CASE2: THE GEOMETRY IS TOTALLY IRREGULAR ...........................................09  
      3.2.1 MAPPING OF IRREGULAR SECTIONS .......................................................13  
      3.2.2 CREATING THE FINITE ELEMENT MODEL ..............................................13

4. **3-DIMENSIONAL REPRESENTATION** ..........................................................................22  
   4.1 VIEWING GRAPHICS THROUGH MOVIE .......................................................23  
   4.2 VIEWING GRAPHICS IN THE INTERGRAPH ....................................................23  
   4.3 3-D SURFACES REPRESENTATION .................................................................24  
      4.3.1 STRAIGHT LINED GEOMETRY .................................................................24  
      4.3.2 STRAIGHT LINED GEOM. WITH TWIST ..................................................24  
      4.3.3 BEZIER SURFACE ....................................................................................25  
      4.3.4 B-SPLINE SURFACE ...............................................................................27  
      4.3.5 SOLID MODEL IN PATRAN .................................................................27

5. **MODELING OF BLADE AND ROOT IN PATRAN** ....................................................30  
   5.1 PATRAN AND ITS INTEGRATED MODULES .......................................................31
5.2 PATRAN COMPATIBILITY ......................................................... 33
5.3 PATRAN CONSTRUCTION COMMANDS ......................... 35
5.4 PATRAN COMMAND SYNTAX .............................................. 36
5.5 MESH OPTIONS ................................................................. 48

6. CASTING SIMULATION USING CAST3 ................................ 54
   6.1 SELECTION OF CAST3 AS CASTING SIMULATION PACKAGE ........................................... 54
   6.2 CAST3 ANALYSES .............................................................. 54
   6.3 SETTING UP THE MODEL IN CAST3PRE ................................ 57
   6.4 USING CAST3 TO SET UP GIVEN PROBLEM ................................................. 59
   6.5 MATERIAL AND MATERIAL PROPERTIES ........................................ 60

7. DISCUSSION OF RESULTS .................................................... 63

8. CONCLUSIONS ......................................................................... 70

9. FUTURE WORK AND RECOMMENDATIONS .......................... 72

10. REFERENCES ................................................................. 75

APPENDIX

APPENDIX 1
SAMPLE RUN : REGULAR TURBINE BLADE WITHOUT TWIST ......................... 77

APPENDIX 2
SAMPLE RUN : IRREGULAR TURBINE BLADE WITHOUT
TWIST WITH VARYING Z-DEPTHS ...................... 79

APPENDIX 3
SAMPLE RUN: IRREGULAR TURBINE BLADE WITH DIFFERENT 3-D SURFACES ...................... 81

APPENDIX 4
SAMPLE RUN: CASTING OF TURBINE BLADE GEOMETRY IN PATRAN ...................... 88
Chapter 1

INTRODUCTION

There are numerous engineering applications which require information in a 3-D format. Through the use of 3-D computer graphics, engineering information is more easily stored, accessed and modified. It is also more clearly represented and is more conducive to use in engineering design. (CAD/CAM/CAE)

Computer aided design (CAD) is at the heart of Computer Integrated Manufacturing (CIM) systems. At a CAD workstation, a designer converts design requirements into a mathematical and graphical model of the part. The resulting digital model of the part is used for further analysis. The CAD data is integrated with project planning, numerical control, programming and tool design, finally resulting in the production of a complex machine part.(1)

One of the main requirements today in the industry is the need to model the geometry of the part in such a way that sufficient information is available for reuse later. Models are useful as often the designer can study certain characteristics of an object easily based on the model than its physical counterpart. Today in the industry there is an ever growing need for analytically and topologically complete 3-D models. These models allow one to derive quickly and automatically any geometric property or attribute the object is likely to possess. In some cases one may wish to represent arbitrary shape information, that is, shapes that do not have special names or their character is not well defined.(2)
Much of this research aims to generate a technique which allows one to describe complex shapes as arrangements of simpler 2-D sections.

1.1 THE NEED TO MODEL:

Engineering analysis is undergoing rapid changes with the advent of solid modelers. Solid modelers permit rapid construction of finite element models. They also permit automatic static and dynamic analysis of mechanical parts subjected to a wide variety of loading conditions.

A model is created as it is a more convenient and economic substitute for the real object. It is often easier and more convenient to analyze a model (FEM analysis) than it is to experiment with the real object. A model serves as an important tool to transmit design information in the industry. When a model is constructed a substitute or a representation of the object is created. The object is cast in a more convenient form, which is easier to analyze.(3)

Models are used to construct a precise mathematical description of the shape of the real object. Hence the requirement is to build up a computer based model that contains all the necessary information on the part and product. The model should be unique and complete.
1.2 THE NEED FOR INTERACTIVE COMPUTER GRAPHICS:

Computer graphics involves the creation and manipulation of pictures with the aid of the computer. When the user has some control over the image it is termed as interactive computer graphics.

The main advantage of interactive computer graphics in any manufacturing process is that it allows the user to interact with the computer quickly to correct certain flaws or design errors.

With the help of computer graphics the user can also translate or rotate the 3-D model to get a better understanding of the geometry. Also the user may wish to view the object with hidden line removal or with color shading.(4)
CHAPTER 2

GENERAL OVERVIEW

The important features that distinguish a CAD system from traditional engineering computing is the use of interactive graphics techniques and the potential to reuse information. The requirement to reuse part data leads to two further distinguishing features.

a] Modelling the part geometry in such a way so that sufficient information can be easily accessed later.

b] The ability to store, transmit and retrieve part data.

The main purpose of this research is to automatically obtain a 3-D solid model from 2-D sliced sections. The procedure involves inputting the co-ordinates of the nodes of a given 2-D sliced section by hand or through a data file.

In case of a complex shape like a turbine blade, first the geometry of the turbine blade is generated, i.e. the 3-D model of the turbine blade is built up. Once the model is built, it can be made available to designers for the analysis as well as preparing N.C. programs for machining, FEM analysis etc.

The algorithm aims at creating a 3-D solid model of complex shapes. It follows two different approaches depending on whether the complex shape is a symmetric projected solid or completely irregular.

If the given shape is a symmetric projected solid, the co-ordinates of any one 2-D section will suffice. In case of symmetric
shapes, the geometry is scaled up or down to the required final product shape. Intermediate sections are then determined. If the 2-D sections are unsymmetrical, two or more sections are mapped and connectivities generated. The 2-D slices are stacked, so as to say, to generate a 3-D model.

The first part of this research involves the creation of a 3-D model, built by stacking together 2-D sliced sections. The second stage involves the use of a finite element package to validate the model. PATRAN and CAST3 have been used for this purpose.
CHAPTER 3

ALGORITHM FOR 3-D MODELLING OF 2-D SLICED SECTIONS

The information in the model of a three dimensional object can be divided into two important classes, geometry and topology. Geometry is concerned with measurements, such as the location of a point or the dimensions of an object. Topological information records how points can be aggregated to form polygons and polygons to form objects. Auxiliary information such as color of the surfaces or shading can also be recorded in the model.

As seen in flow chart 1, the user has to specify the length of the required geometry, number of nodes on each 2-D section and the number of elements of the geometry.

3.1 CASE I : The geometry is a symmetric solid projection

In case all the 2-D sections are to scale i.e. the first and the last 2-D slices of the given geometry are to scale, the user needs only to provide the X and Y co-ordinates of the initial 2-D section. The algorithm automatically calculates the centroid of the section and the X and Y co-ordinates of the nodes on each and every section. This will depend on the number of elements along the geometry and its length. The 2-D sections will be evenly spaced along the length of the geometry unless otherwise specified.
FIGURE 3.1: SHOWS A MOVIE OUTPUT FOR A TURBINE BLADE
A sample run of a 2-D section of a simple turbine blade is shown in Appendix 1-A. The MOVIE output of the blade is shown in Figure 3.1.

In case of blades or gears, there is often a twist along the length of the geometry. Transformation of object points along circular paths is called rotation. This type of rotation is specified with a rotation angle, which determines the amount of rotation for each vertex of a polygon. Figure 3.2 illustrates displacement of a point from position \((x,y)\) to a position \((x',y')\) determined by a specified rotation angle relative to the co-ordinates origin.(5) Provision is made in the algorithm if such a twist is needed. The user has to specify the angle of twist along the length of the geometry.

Equations of rotation:

As the rotation is about the origin, the distances from the origin to \(P\) and \(P'\) are equal and are labeled \(r\).

Hence, by simple trigonometry,

\[
\begin{align*}
x &= r \cos \phi, \\
y &= r \sin \phi
\end{align*}
\]

......3.1
and,

\[ x' = r \cos(\phi + \beta) = r \cos \phi \cos \beta - r \sin \phi \sin \beta \]  \quad \text{...3.2}

\[ y' = r \sin(\phi + \beta) = r \cos \phi \sin \beta + r \sin \phi \cos \beta \]  \quad \text{...3.3}

Now, substituting equation 3.1 in equation 3.2 and 3.3,

\[ x' = x \cos \beta - y \sin \beta \]  \quad \text{...3.4}

\[ y' = x \sin \beta + y \cos \beta \]  \quad \text{...3.5}

In the matrix form,

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} =
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\begin{bmatrix}
  \cos \beta & \sin \beta \\
  -\sin \beta & \cos \beta
\end{bmatrix}
\]  \quad \text{...3.6}

A sample run of a complex shape with a twist of 5° along the length of the geometry is shown in the MOVIE output in Figure 3.3.

Note: The user can key in the X and Y co-ordinates or can input it through a data file. Also the algorithm can calculate in addition to the centroid, the perimeter of any irregular shape. The volume of the 3-D model created can also be calculated if the user so desires.

3.2 CASE II : The geometry is totally irregular

As in case I, the user has to specify the length of the geometry, the number of nodes along the 2-D section and the number of sections. The user has to then input the data file which contains the X and Y co-ordinates of each of the 2-D sliced sections. If the 2-D sections are equally spaced, the algorithm will automatically provide the Z co-ordinate for each and every sliced section depending on the length of the geometry. The algorithm then proceeds to map the sections depending on the shortest distance and the minimum surface area criteria. This will be discussed in detail later.
FIGURE 3.3: SHOWS A MOVIE OUTPUT FOR A TURBINE BLADE WITH TWIST
FIGURE 3.4: MODEL WITH SUBSECTIONS OF VARYING LENGTHS
FIGURE 3.5: MODEL WITH FINER MESH IN CRITICAL REGIONS
If unequal sections are required along the length of the geometry, i.e. if the given 2-D sliced sections are not uniformly cut off, the user has to specify the Z depth of each respective section. There is also a provision for automatically incorporating sub-sections along each section in the geometry. This might help in finite element analysis of the created solid model when a finer mesh is required in some critical regions as compared to a coarser mesh in other non critical regions. Figures 3.4 and 3.5 demonstrate this quite satisfactorily.

Before the mapping of these 2-D sections begins, there is a provision in the algorithm to view the initial geometry for each of the 2-D sections so that the user is absolutely sure that the co-ordinates of each of the sections is proper as even a slight deviation might result in a gross mistake in the final 3-D model.

3.2.1 MAPPING OF IRREGULAR SECTIONS:

The algorithm provides an option of aligning the 2-D sections along their centroids or in case of a turbine blade, the stacking axes, from where all the co-ordinates are measured. The default is taken up as the origin (0,0,0). The irregular 2-D sliced sections may or may not be equally spaced. The distance between each section can be manually given. Refer Figure 3.6

3.2.2 CREATING THE FINITE ELEMENT MODEL:

The algorithm calculates the shortest distance between the grid points on any two successive sections, within the constraints of minimum surface area of a patch.(6)
FIGURE 3.6: IRREGULAR 2-D SECTIONS WHICH ARE NOT EQUALLY SPACED
As shown in the Figure 3.7, if 1,2,3 are grid points on section(a), 1',2',3' on section(b) and 1'',2'',3'' on section(c), the user has to specify the connectivity of one grid point on every 2-D section, say 1,1',1'' in the above case. That is to say, point 1 on section(a) connects point 1' on section(b), which in turn connects point 1'' on section(c). Now what needs to be determined is where point 2 will connect on section(b). Considering patch 1,2,2'1' the algorithm will calculate the shortest distance between grid point 2 and line 1',2'. It will then take up patch 1,2,3',2' and calculate the shortest distance of grid point 2 onto line 2',3'. Comparing the two distances it will put a grid point 2' on section(b). The same procedure is repeated in the section adjacent to section(b), but now the algorithm uses patch defined by 1',2'',1'',2'' and so on.
An important point to be considered is that in incorporating a grid point 2* instead of 2', an effective patch 1,2,2*,1' is being created. In some cases it might so happen that the 2-D section(b) might get distorted. Hence a provision is made in the algorithm which checks whether the geometry has been distorted beyond a permissible tolerance. This is done by simply checking the surface area of patch 1',2',2* and that of patch 1,2,2*,1'. If the ratio of the surface areas of the two patches is negligible the algorithm proceeds to the next stage. If it is not negligible, it remeshes the geometry by incorporating the necessary extra grid points along the entire geometry. In doing so it will put an additional grid point on each and every 2-D section which connects grid point 2* on section(b). Refer Figure 3.8.

![Figure 3.8](image_url)
Now the algorithm can take up another point on section(a). As described earlier the same steps will be repeated, ie. mapping of the two adjacent sections with regard to the shortest distance within the constraints of minimum surface area of a patch. The algorithm will then check the surface area of the patch formed by 2'2'3' and that of patch formed by 2'233'. If the ratio of these two patches is not within permissible tolerance, it implies that the 2-D section(b) has been distorted. Hence it will place an additional point 3' on section(a) to connect point 2 on section(b). This is shown in Figure 3.8. Similarly the algorithm will check for distortion of section(c).

Hence the algorithm checks at every stage with the initial 2-D sections for confirming that the original 2-D shape is maintained. If the shape is distorted it puts additional nodes on each 2-D section. The additional grid points are interpolated on each section. After the whole geometry is meshed, the model can be viewed in MOVIE. The total number of additional grid points per section are given at the end of the remeshing.

If in the given model it is necessary to specify that a certain grid point (1) has to connect to grid point (2) on another section and another point (5) has to connect point (9), there is a provision to set the number of loops within the algorithm. Depending upon the number of grid points the user wants to manually connect, the user has to specify the number of loops. If the number of loops is three then the user wants to connect three points on section(a) to three points on section(b). The algorithm will ask the user to input the number of
nodes within each loop. If the mesh is to be generated automatically the number of loops will be 1. Refer Figure 3.9

Flow chart 1 and 2 shows the stages in creation of a 3-D model from 2-D sliced sections.
FLOW CHART:

3-D MODELING OF 3-D SLICED SECTIONS

User input:
Length of geometry, number of nodes
in each section, number of sections.

-ES

Is the shape a solid symmetric projection?

-ES

Input Data file with coordinates.

Calculates the coordinates of the nodes in each section depending upon the number of sections required and length of the geometry.

-ES

Is twist required in the geometry?

-ES

Output angle of twist.

Creates MOVIE data file.

STOP
FLOW CHART 1

3-D MODELLING OF 3-D SLICED SECTIONS

A

NO

Input data file with coordinates

YES
Are equal lengths required for each section

NO

Mapping of sections depending upon shortest distance and minimum surface area criteria

Input number of sectional length, number of subsections in each sectional length

DIE SURFACE DEFINITION

1. Directly to MOVIE data file
2. Incorporating twist in geometry
3. Fitting B-spline curve through DFPI which can be viewed in the INTERGRAPH
4. Fitting Bezier curve/MOVIE data file

STOP
FLOW CHART 2

3-D MODELLING OF 2-D SL CED

SECTIONS

Mapping of 2-D sections

Calculations for Z-coordinates depending on the length of the geometry and the number of sections

Calculates the centroid of each section

Calculates the shortest distance between points on any two successive sections within the constraints of minimum surface area of a patch

Checks with initial 2-D sections for original shape

SHAPE OK

YES

Rectifies distorted shape by automatically generating additional nodes on the section

Output of 3-D solid model

NO

Interpolates additional nodes on each section for one-to-one connectivity (Remeshing)
CHAPTER 4

3-D DIMENSIONAL REPRESENTATION

Many three-dimensional objects can be represented by a set of plane polygon surfaces. Polygon representation provides an approximate description of the object. This polygonal-mesh representation can be displayed quickly to give a general indication of the object's structure, and this approximation can be improved by dividing object surfaces into smaller polygon faces.

Each polygon in an object can be specified to a graphics package to define the vertex co-ordinates. Parameters specifying the spatial orientations of each polygons are obtained from the vertex co-ordinate values and the equation that defines the polygon planes. The equation of a plane surface can be expressed in the form

$$Ax + By + Cz + D = 0$$

where \((x,y,z)\) is any point on the plane. The coefficients \(A, B, C\) and \(D\) are constants that can be calculated using the co-ordinate values of three noncollinear points in the plane. Denoting the co-ordinates of three vertices of a polygon as \((x_1,y_1,z_1)\), \((x_2,y_2,z_2)\) and \((x_3,y_3,z_3)\) and using Cramer's rule the solutions can be written in the explicit form as:
After running the program to generate the 3-D geometry, the user can view the geometry in MOVIE or on INTERGRAPH. Design of 3-D packages requires some consideration that are not necessary with 2-D packages. A significant difference between the two packages is that a 3-D package must include methods for modeling of surfaces of solid objects, hidden line removal and transformation of objects in space.(7)

4.1 VIEWING GRAPHICS THROUGH MOVIE:

MOVIE is a general purpose computer graphics package. MOVIE provides display of the data in line drawing or continuous tone image format. Multiple models can be viewed simultaneously. The solid model can be translated in any direction or rotated about any global or Cartesian axes. Selection of color and shading formats are also possible.

4.2 VIEWING GRAPHICS ON THE INTERGRAPH:

INTERGRAPH is a versatile system with many more additional features than MOVIE. It has a dual screen capability with which the user can view all the sides of the object at one setting. Hidden line
removal, shading and setting up of B-spline and Bezier surfaces are quite simple on the INTERGRAPH.

4.3 3-D SURFACE REPRESENTATION:

The main program provides four different surface geometries and the creation of solid model in PATRAN.

a] Straight line geometry (MOVIE output)
b] Straight line geometry with twist (MOVIE output)
c] Bezier surface (MOVIE output)
d] B-spline surface (DFPI/INTERGRAPH output)
e] Solid model creation in PATRAN

4.3.1 Straight line geometry:

In the first option a simple straight line geometry is generated. The MOVIE output file containing the X, Y and Z coordinates of the geometry with their respective connectivities is automatically created. On running MOVIE the 3-D model can be viewed on the screen.

4.3.2 Straight line geometry with twist:

The second option gives a straight line geometry as well but with a twist. This twist is provided along the length of the geometry. In case of complex turbine blades, the geometry along the length of the blade is often not regular. Hence the algorithm allows the user to provide a certain angle of twist along the length of the geometry. The graphics can be viewed through MOVIE.

The above two options generate a straight line connectivity between two successive 2-D sections. This may lead to
discontinuities along the surface. Many a times the user would like a smooth profile with a higher degree of continuity. Hence, an option for incorporating Bezier and B-spline surfaces is provided for.

By introducing a fourth parameter $u$, into the co-ordinate description of a curve, each of the three cartesian co-ordinates can be expressed in the parametric form.

$$P(u) = (x(u), y(u), z(u))$$

The parametric functions $x$, $y$ and $z$ trace out the location of the curve or surface. The parametric functions can themselves take many forms. A single curve can be approximated in several different ways.

Generally it is not possible to device a simple function that specifies the shape of an entire curve or surface. But it is not so difficult if the function applies only to a small piece of the shape. The entire shape is then defined by a series of functions pieced together. Piecewise approximations help achieve the desirable local control property by defining a piece in terms of only control points near it and of the continuity requirements of its joints. (8)

Bezier and B-spline formulations both use control points that lie off the curve but nevertheless provide remarkable effective control of the curve shape. Hence the algorithm provides the user with a Bezier and a B-spline die surface definition.

4.3.3 Bezier curves and surfaces:

The curve $P(u)$ is defined in terms of the location of $n + 1$ control points.
\[ P(u) = \sum_{n=0}^{i=0} p_i B_{i,n}(u) \quad \text{..................4.1} \]

where
\[ B_{i,n}(u) = C(n,i) u^i(1-u)^{n-i} \quad \text{..................4.2} \]

and \( C(n,i) \) is the binomial coefficient
\[ C(n,i) = \frac{n!}{i!(n-i)!} \quad \text{..................4.3} \]

Writing separately for the \( x, y \) and \( z \) parametric functions we have:

\[ x(u) = \sum_{n=0}^{i=0} x_i B_{i,n}(u) \]
\[ y(u) = \sum_{n=0}^{i=0} y_i B_{i,n}(u) \]
\[ z(u) = \sum_{n=0}^{i=0} z_i B_{i,n}(u) \quad \text{..................4.4} \]

where the three dimensional control point \( p_i \) is \([x_i, y_i, z_i]\) and \( u \) goes from 0 to 1.

Two Bezier surface patches joined together at a boundary indicated by arrows. Control points are chosen to yield first-order continuity across the boundary.
A Bezier curve can easily be extended to describe a three dimensional Bezier surface.

\[ P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} p_{i,j} B_{i,n}(u) B_{j,m}(v) \]  

Figure 4.1 shows two Bezier surfaces joined together at a boundary indicated by arrows. The main disadvantage of a Bezier curve is that it has global and not local control.(9) Hence, if a control point is changed then the complete die surface changes. As a B-spline provides local control an option for B-spline surface is also incorporated in the algorithm. Figure 4.2 shows the blade geometry fitted with Bezier curves.

Though the B-spline has the same general form as the Bezier, the value for \( u \) ranges from 0 to \( n-k+2 \), where \( k \) controls the order of continuity of the curve. The blending function is

\[ N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}} \]  

The smaller the value of \( k \), the closer the spline will follow the contour.

4.3.4 B-spline surface:

The fourth option for the 3-D surface geometry uses a DFPI routine to create a design file for the die geometry, which can be viewed on the INTERGRAPH. B-spline curves are fitted along the geometry to give a smooth profile for the turbine blade as shown in figure 4.3. The design file is created automatically, the user having only to input the number of nodes on each 2-D section of the die geometry.(10) The user can open the design file in Graphics on INTERGRAPH and view the 3-D model.
FIGURE 4.2: BLADE GEOMETRY FITTED WITH A BEZIER SURFACE
FIGURE 4.3: BLADE GEOMETRY FITTED WITH A B-SPLINE
(INTERGRAPH OUTPUT)
CHAPTER 5

MODELING OF BLADE AND ROOT IN PATRAN

Mechanical computer aided engineering (MCAE) is one of the three major components of computer aided engineering (CAE). The other two components are CAD and CAM. MCAE is the process of defining a physical model of the design in the computer, then subjecting that model to a simulated environment to determine its response.

The main advantage of MCAE is that it allows the designer to take his design from the conception stage to reality with less need of prototypes. This greatly reduces the development cycle time and reduces the cost.

PATRAN offers an extensive MCAE software interface system. It not only has the capability to perform many of the MCAE functions but also unites a wide range of existing software tools. With PATRAN the engineer can model the design, model the environment, analyze the model within the environment and interpret the results.

PATRAN was chosen to model the given problem of a turbine blade as nearly all major finite element codes, as well as, many computer aided drafting and manufacturing software packages can be linked with PATRAN.(11)
5.1 PATRAN AND ITS INTEGRATED MODULES:

PATRAN can be described as a general purpose, 3-D MCAE software package that uses interactive graphics to link engineering design analyses and result evaluation functions. The package incorporates an advanced solid modeler, extensive graphic capabilities, and a unique open ended 'gateway' architecture that facilitates access to virtually all design, analysis and manufacturing software programs.

The package consists of five tightly integrated modules, namely P/SOLID, P/FEM, P/IMAGE, P/POST, and P/PLOT.

P/SOLID: It is a geometric modeling system which incorporates both analytical solid modeling and trimmed surface modeling techniques. The two modeling methods are interwoven, allowing the designer to access both simultaneously. The P/SOLID analytical solid modeling library contains comprehensive commands to generate and manipulate entities like points, curves, surfaces, co-ordinate frames etc. It also provides framework for intermediate finite element meshing. P/SOLID's integration into PATRAN Plus makes it easy to accurately conceptualize, model and modify potential designs.

P/FEM: It prepares the model for analysis. The geometry created in P/SOLID can be directly accessed to develop a finite element mesh and also for the application of loads and boundary conditions. Meshes can be uniform across the whole model or concentrated in some critical regions. It can generate finite element meshes for 1-D, 2-D
and 3-D models. It allows replication and manipulation of the model. It also incorporates features by which the model can be optimized to ensure that the model is complete before analysis.

P/IMAGE: It encompasses the complete graphics capabilities found within PATRAN Plus. It includes graphics feedback for all commands, provides for shading and hidden line removal, and serves as a visual verification prior to executing an analysis. P/IMAGE pre-processing features include identification labels for all entities, adjustable colors, selected displays at any time, hidden line plots for geometric as well as finite element plots, combination of shaded and hidden line wire frame images on one display. P/IMAGE post processing features also include results which can be color coded and displayed, titles printed on screen automatically or manually and wireframe animation.

P/POST: It displays the results of any analysis being carried out in PATRAN. P/POST eliminates the need of stacks of printouts by making it easier for the designer to understand the results of the analysis and identify critical regions. P/POST employs a variety of means to depict results which include animation, deformed geometry plots, contour plots, fringe plots, XY plots for beam elements etc. Results can be displayed as deformed geometry superimposed on the undeformed geometry.

P/PLOT: It generates X-Y plots. This module permits the user to compare two generic sets of data and assists in evaluating the
designed model. The close integration of this module with others in PATRAN Plus environment enables the user to easily generate multiple graphs from within the PATRAN system. P/ PLOT is an extremely versatile module with commands like the number of plots needed to a screen, size of each plot, size and color of each axis, number of curves to a graph, data input formats etc. It can incorporate upto 16 simultaneous plots. Hardcopy images can also be generated.

5.2 PATRAN COMPATIBILITY:

The table below shows PATRAN compatibility with the basic components involved in MCAE analytical methods.

TABLE 5.1 : MCAE ANALYSIS METHOD

<table>
<thead>
<tr>
<th>ANALYSIS PROCEDURE</th>
<th>PATRAN COMPARISON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Defines product geometry and constructs the geometric model</td>
<td>P/SOLID provides extensive geometric modeling capabilities to construct and view the model.</td>
</tr>
<tr>
<td>2. Build a model and prepares it for analysis by generating a mesh, i.e. nodes and elements on the model.</td>
<td>Constructs an analysis model using P/FEM</td>
</tr>
</tbody>
</table>
3. Defines environmental condi. and creates a simulated envir. that includes B.C.'s, temp's., element & material properties.

4. Subjects the model to a simulated environment and determines the model response to a set of chosen conditions.

5. Evaluates the model response to optimize the design.

Can assign various B.C's to create a test environment.

PATRAN offers several optional but completely integrated application modules that provide focused analysis.

PATRAN's post processor displays complex analysis data in color coded results.
5.3 PATRAN CONSTRUCTION COMMANDS:

PATRAN construction commands follow a special syntax called the 'NOODL' rule. This helps the user remember the proper input format for the PATRAN directives. The format is as follows:

NAME, OUTPUT-LIST, OPTION, DATA, LIST1, LIST2.....

NAME : It is the command being invoked. It specifies what the user wants to create.
(Grid, line, Patch, Hyperpatch, Mesh....)

OUTPUT : It lists the item being created. These identity numbers can be assigned manually or automatically by PATRAN:
- n # number with next n numbers
- n1Tn2 numbers n1 through n2
- n1Tn2Bk numbers n1 through n2 by increments of k

OPTION : Refers to the method associated with the name field. The way in which the user wants to create the name field.

DATA : Contains the necessary information for the option being invoked.
- x,y,z co-ordinate triplet
- uses data from data list n

LIST1, LIST2: The list of existing entities from which output list is to be generated by the method specified in OPTION and as directed by DATA.
5.4 PATRAN COMMAND SYNTAX:

A summary of syntax tools that you can use in defining data within a field is as follows:

Comma : It is used to separate fields in a command.

Slash : It is used to separate items within a field.

eg: GRID, 101, , 1.2/3.8/4.2

Creates a grid point at the given co-ordinates (1.2,3.8,4.2)

Parenthesis: It is another way of separating data within a data field.

The field is enclosed in parenthesis and separated with commas.

eg: LINE, (1,2,3), STRAIGHT,,........

Creates three lines numbered 1, 2 and 3.

Through : A range of values in either the list field or data field can be specified by using Through operator.

n through m

m has to be greater than n. If not, only m will be recognized.

By : It is used in conjunction with THRU list to change the increment by which the list is created.

eg: GRID, 10T30B10, , 1/2/3/2/4/6/4/8/12

will create Grid point 10 at (1,2,3), Grid 20 at (2,4,6,) and Grid 30 at (4,8,12).

#: It is a wild card used in place of a number.

eg: HP, 1T#, PL
: An 'at' character can be used anywhere in a command line that a PATRAN entity id. is to be specified. PATRAN will then prompt for the required id. via the cursor.

PATRAN offers two kinds of modes for creating the model. The Command mode and the Menu mode. Both offer similar menu picks. The command mode is the mode of computer user interface where the user issues commands from a command line prompt. In the menu mode the menus are picked by numbers only.

Though the Menu mode is preferred by beginners, the main advantage of command mode is that allows the user to ignore the PATRAN menu structure and issue any command at any time. Changing from Menu mode to Command mode and vice versa is quite simple.

The problem taken up is that of a turbine blade with its root. For sake of convenience in modeling, the geometry can be broken up into two parts.

1] The blade
2] The root

The root in turn consists of three sections namely the land, the web and the bottom hexagonal part.

As discussed in Chapter Three, the blade portion of the geometry can be built by using the algorithm Blade. This portion will create a data file which gives co-ordinates of each and every grid points with their respective connectivities for the model.
As the solid modeler to be used is PATRAN, this data file is to be opened in PATRAN by making use of the read file option in the menu. This will give the locations of all the grid points along the whole blade length. The connectivities can be automatically generated.

To create a solid model in PATRAN individual hyperpatches have to be first generated and then blended together to create the whole blade geometry.

If 1,2,3,4,5 and 6 are some given grid points and 1,2, and 3 are lines joining grid points 1 and 4, 2 and 5 and 3 and 6 respectively, a hyperpatch can be created between these grid points and some other grid points 1',2',3',4',5', and 6' in two ways.

(a) Using the QUAD option a patch can be drawn for grid points 1245 and 1'2'4'5'. Then using 2 patch option the hyperpatch can be drawn.

\[
\text{PA, 1, QUAD, , 1/2/5/4}
\]
\[
\text{PA, 2, QUAD, , 1'/2'/5'/6'}
\]
\[
\text{HP, 1, 2P, , 1, 2}
\]

(b) Draw a patch using 2 line option and then using 2 patch option hyperpatch can be drawn.

\[
\text{PA, 1, 2L, , 1, 2}
\]
\[
\text{PA, 2, 2L, , 1', 2'}
\]
\[
\text{HP, 1, 2P, , 1, 2}
\]

Hence the solid model of the whole blade can be created.

Figure 5.1 shows two patches 1 and 2 drawn by the two line option. Figure 5.2 shows hyperpatches 1 and 2 being drawn from two patch option.
FIGURE 5.1 PATCHES DRAWN FROM 2-L OPTION
The solid model of the root can also be drawn up in PATRAN using GRID, LINE, PATCHES, HYPERPATCHES options. The root which is made up of three parts, has to be drawn separately in PATRAN. In case of the web and the hexagonal sectional, QUAD option can be used to draw the patches and then 2 patch option can be used to draw the hyperpatches. The land of the root is much more complex as it possesses groves at the bottom which connect the web portion. The land of the root is drawn in PATRAN and patches are created in such a way that when hyperpatches are created from them, any two
adjacent hyperpatches will have a common face. The simplest way to create hyperpatches for the land is to extrude patches by a certain depth in the Y direction. In order to avoid multipoint constraints, care should be taken that the hyperpatches travel uniformly and continuously from the land right up to the bottom portion of the root.

Figures 5.3(a) shows the hyperpatches on the bottom and the web of the root and Figure 5.4(a) shows the breakup of hyperpatches into several smaller ones on the land of the root to achieve this.

After the root and the blade have been modeled independently, they have to be blended together to get the entire geometric model. The blade has to be projected onto the roof or the land of the root so that the base of the blade sits on the roof. At the interface of the blade and the root, the user will have to use multipoint constraints as it is very difficult if not virtually impossible to have uniform continuity of the hyperpatches from the blade to the root.

After the whole geometry has been modeled in PATRAN, a finite element mesh has to be created for analysis of the module. This can be done with ease in PATRAN using the mesh command.

Each hyperpatch can be meshed individually or together. The mesh directive produces both nodes as well as elements from a single command. It has the ability to perform totally automated meshing using either WEDGE or HEX. elements. It can provide a isoparametric mesh or a transitional mesh as required. It has the ability to specify uniform or non uniform node spacing.
FIGURE 5.3(a): HYPERPATCHES ON THE WEB AND THE BOTTOM PART OF THE ROOT
FIGURE 5.3(b): BOTTOM OF ROOT WITH HIDDEN LINE REMOVAL
FIGURE 5.3(c): BOTTOM OF ROOT WITH HIDDEN LINE REMOVAL AND ELEMENT FILL
FIGURE 5.4(a): LAND OF ROOT BROKEN UP INTO SMALLER HYPERPATCHES
FIGURE 5.4(c): LAND OF ROOT WITH HIDDEN LINE REMOVAL AND ELEMENT FILL
5.5 MESH OPTIONS:

Mesh options available in PATRAN are Isoparametric, Number and Length.

5.5.1 Isoparametric Option:

This option allows the user to generate a finite element mesh over the entire geometric model with a single command. With this option the same mesh is produced no matter how the parametric axes are oriented.

eg: MESH, HP1T#, QUAD, ISO, 0.5

Creates a isoparametric mesh of standard QUAD elements on all hyperpatches using an element length of 0.5.

5.5.2 Number option:

It gives the user total control over the number of elements and the node spacing, along an edge of a geometric region. In case of meshing a hyperpatch, the user needs to specify the number of elements and mesh ratio for each edge of the base patch as well as number of elements in the normal direction.

eg: MESH, HP1T 50, HEX, NUM, 1/3/1/3/2

Creates a mesh with HEX. elements with 1,3,1 and 3 elements respective on the base patch and two elements along the normal.

5.5.3 Length Option:

It allows the user to generate a mesh with fairly uniform element size over the entire model. In case of meshing a hyperpatch, the element edge length and number of element along the thickness direction has to be specified.
eg: \texttt{MESH, HP1T10, WEDGE, LEN, 5.0}

Creates a mesh with wedge elements 5.0 units length along each side or edge.

For the given geometry Number option is used as it gives complete control over the number of elements and node spacing along the edge of the each geometric region. Prompted meshing can also be done, but it is time consuming and tedious as it meshes each hyperpatch at a time, requiring interaction from the user at every stage.

Similarly the root of the model can also be meshed, keeping into account the uniformity and the continuity of the mesh.

Flow chart 3 demonstrates the various steps to be followed in meshing up a blade in PATRAN.

Figure 5.5(a) shows the whole geometry meshed up in PATRAN. Figure 5.5(b) shows the whole geometry with hidden line removal and Figure 5.5(c) shows the entire geometry with hidden line removal and fill.
FLOW CHART 3

STEPS TO BE FOLLOWED TO MESH UP A BLADE IN PATRAN

1. PLOT GRID POINTS
2. DRAW LINES THROUGH THE GRID POINTS
3. DRAW PATCHES USING 2L OPTION
4. DRAW HYPERPATCHES USING 2P OPTION
5. MESH USING NUMBER OPTION
FIGURE 5.5(b): WHOLE GEOMETRY SOLID GEOMETRY IN PATRAN
FIGURE 5.5(c): WHOLE GEOMETRY WITH ELEMENT FILL IN PATRAN
CHAPTER 6

CASTING SIMULATION USING CAST3

Computer modules can predict the complex 3-D solidification of metal casting process quite accurately. Time saving in critical areas of product design and development can be achieved with the application of computer simulation. If the results are interpreted accurately, the process parameters can be tested and optimized before any production time, material or money is spent.

6.1 SELECTION OF CAST3 AS CASTING SIMULATION PACKAGE:

CAST3 was chosen for the simulation program to analyze the given problem, as it provides the most complete casting modeling package available. CAST3 simulation program is a very versatile and user friendly package which incorporates conduction, convection, radiation, fluid flow and solute transport problems. Full geometric complexity including gating and risering arrangements can be accommodated. It can even simulate the surroundings, and any type of casting material can be accurately modeled.

6.2 CAST3 ANALYSIS:

A complete CAST3 analysis requires the use of five software packages. The CAST3 simulation analysis can be summarized as shown in Flow chart 4
FLOW CHART 4

STEPS TO BE FOLLOWED FOR CASTING SIMULATION

PATRAN

CAST3PRE

CAST3DG

CAST3

CAST3POST

PATRAN
The finite element model is to be built in PATRAN. This model should contain the following necessary information:

1. x, y and z coordinates of the nodes
2. element connectivity
3. boundary condition records

CAST3PRE: Allows the user to add more specific information to the finite element model. Upon exiting CAST3PRE the analysis is completely defined.

CAST3DG: It reviews the entire model, performs extensive error checking and converts all the units into standard form. It also prepares a summary file which completely describes the model.

CAST3: It will perform the actual simulation analysis. The results of this analysis can be reported in both hardcopy and postprocessing files.

CAST3POST: The postprocessing files can be analyzed using CAST3POST.
6.3 SETTING UP THE MODEL IN CAST3PRE BEFORE USING CAST3:

After the entire model has been modeled in PATRAN a neutral file can be generated which has stored all the necessary information pertaining to the model. CAST3PRE is a screen oriented preprocessor which combines this information with the information input by the user. For the preprocessor to function correctly, CAST3PRE expects some rules to be followed while building the PATRAN model.

a) Nodes and element id's must be optimized in the PATRAN model so that they are sequential.

b) All nodes must be referenced at least once.

c) Heat boundary conditions if present should be created in PATRAN by DFEGing the appropriate faces in the model.

   DFEG, HP7T10, HEAT, 2, 2, F4

   Puts a convection heat transfer coefficient on face four of hyperpatches seven to ten. The value 2 has been employed as an identifier. The specific value of the heat transfer coefficient is assigned in CAST3PRE.

d) Constant Temperature boundary conditions if present can be created in PATRAN by DFEGing the appropriate face in the model.

   DFEG, HP2, TEMP, 3, 3, F2

   Puts a fixed temperature on face two of hyperpatch two.

e) Volume heat sources can be generated in the same way.

   DFEG, HP1T10, HEAT, 4, 4

f) Each three-dimensional region must be assigned a property ID in PATRAN.
PFEG, HP3, HEX, 1
Assigns a property ID. 1 in PATRAN to hyperpatch three.

g) Coincident interfaces will be so generated automatically by CAST3PRE if indicated by the user. Hence when dissimilar materials share a common face, a coincident interface is automatically inserted at that location. As mentioned in (f), each interface has to be identified by PFEG ID. numbers on either side. All PFEG regions can be assigned their associated material properties.

h) When nodes of one region lie on the face of elements of another region, a multi-point constraint has to be generated. Even if the two regions have the same material properties, they have to be defined by different property ID. numbers.

i) Multi-point constraints cannot be generated across a coincident node interface

j) The enclosure patches which are used in radiation model must totally enclose the casting.

k) If the view factor radiation model is employed, maximum numerical accuracy will be obtained if the origin of the PATRAN coordinate system is located within the casting.

l) When View Factor radiation model is employed, participating faces need not be defined using DFEG PATRAN identifier.
6.4 USING CAST3PRE TO SET UP ANALYSIS IN CAST3 FOR GIVEN PROBLEM:

CAST3PRE makes use of various screens through which information has to be input by the user. It is designed to present the user with only the question that needs to be answered in the analysis. This minimizes the number of screens the user has to travel through.

The user has the option of choosing the input units and tolerances. CAST3PRE gives the element breakup (depending on the type of elements used in meshing), the minimum and maximum X,Y and Z co-ordinates, count of coincident nodes, coincident interfaces etc. The control parameters can be steady state or transient. The problem to be analyzed might be linear or non-linear.

Contained within the CASTPRE database are material types listed in alphabetical order. CAST3PRE provides the user with two options. The user can either assign a material type which is already present in the database or can add a new material property to the database. A separate material entry is required for each unique material recorded. Hence if the user wants to assign two different regions like the root and the blade in the case of the present study, by the same material properties, two different material records are needed to be input.

A temperature dependent specific heat can be incorporated into the model. This can be done in two ways. The user can either input a specific heat and temperature curve combination or give an enthalpy curve by itself. If both are assigned to the material, the specific heat will be calculated from the enthalpy curve. For a
material undergoing change of phase it is better to use enthalpy curve method. (12)

6.5 MATERIAL PROPERTIES:

MATERIAL: IN-100 (Nickel based alloy)
COMPONENT: Turbine blade with root
ALLOY:
- Thermal Conductivity = 17.3 W/m-K
- Density = 7.75 gm/cc
- Latent heat of fusion = 70 cal/gm
- Specific Heat = 480 J/kg-K
- Pouring Temperature = 2804 °F
- Ambient Temperature = 80 °F
- Time to pour = 5 seconds

<table>
<thead>
<tr>
<th>Temperature in °F</th>
<th>Conductivity Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.00</td>
</tr>
<tr>
<td>1200</td>
<td>1.09248</td>
</tr>
<tr>
<td>1400</td>
<td>1.2543</td>
</tr>
<tr>
<td>1600</td>
<td>1.45665</td>
</tr>
<tr>
<td>1800</td>
<td>1.66474</td>
</tr>
</tbody>
</table>
### TABLE 6.2 SPECIFIC HEAT

<table>
<thead>
<tr>
<th>Temperature in °F</th>
<th>Specific Heat Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.00</td>
</tr>
<tr>
<td>1200</td>
<td>1.04166</td>
</tr>
<tr>
<td>1400</td>
<td>1.09375</td>
</tr>
<tr>
<td>1600</td>
<td>1.13542</td>
</tr>
<tr>
<td>1800</td>
<td>1.21875</td>
</tr>
</tbody>
</table>

### TABLE 6.3 ENTHALPY

<table>
<thead>
<tr>
<th>Temperature in °F</th>
<th>Enthalpy in J/g</th>
</tr>
</thead>
<tbody>
<tr>
<td>1708</td>
<td>978</td>
</tr>
<tr>
<td>1708.2</td>
<td>1826.2</td>
</tr>
<tr>
<td>1720</td>
<td>2755.8</td>
</tr>
<tr>
<td>1777.4</td>
<td>3727.4</td>
</tr>
<tr>
<td>1779.4</td>
<td>4693.8</td>
</tr>
<tr>
<td>1786.6</td>
<td>5668.8</td>
</tr>
<tr>
<td>1829.6</td>
<td>6657.8</td>
</tr>
<tr>
<td>1962.9</td>
<td>7732.1</td>
</tr>
<tr>
<td>1975.3</td>
<td>8912.1</td>
</tr>
</tbody>
</table>
A natural cooling simulation was run using CAST3 to study the solidification process of the given complex geometry. A natural cooling simulation was undertaken to study the variation in cooling process of the blade and root part of the geometry using IN-100 as the alloy. This was to demonstrate that the created geometry would act like one piece casting though it has been created and meshed up separately.
CHAPTER 7

DISCUSSION OF RESULTS

The entire geometry of the blade which was built in PATRAN has 2996 nodes and 1948 elements. The blade and root portion of the geometry as mentioned earlier was built and meshed separately. They were then blended together as whole model was to be created as a one piece casting.

A natural cooling simulation run for material IN-100 showed the following effects:

The bottom hexagonal portion of the root showed high temperature regions. This implies that the bottom of the blade geometry would be the last part in the casting model to solidify. This can be attributed to the fact that the region is bulky as compared to the other regions.

The land which lies between the web and the blade acted like a fin and showed regions which had the lowest temperature in the geometry. This is due to the fact that it is the thinnest part of the geometry.

The blade showed an uniform cooling pattern, with the hottest region lying in the interior, the semi-hot region lying along the
leading edge or the thicker part of the aerofoil 2-D section and the coolest part along the thin trailing edge.

The region along the interphase of the root and the blade geometry also shows healthy cooling patterns. Though created separately, there seems to be no change in the cooling pattern along the boundary due to the fact that multipoint constraints were provided, which gives an effect similar to the blade being merged into the root.

The cooling pattern in general looks gradual and uniform from the bottom of the root to the tip of the aerofoil section. Figure 7.1 to 7.3 show the temperature contours in whole geometry at different time steps. Figures 7.4 and 7.5 show a plot of the isochrons which are often helpful to illustrate the progression of solidification in the model.

As seen in Figures 7.4 and 7.5 the whole model will take 101414 seconds to solidify. The blade solidifies after 69874 seconds. The last region to solidify is the base of the root. Normally for such a model directional solidification is applied. Chills are placed along the bulky base of the root to increase the rate of solidification in that region. This is done as nonuniform solidification may lead to defects like porosity, shrinkage etc., leading to failure at a critical juncture.
FIGURE 7.1: TEMPERATURE CONTOURS AT TIME STEP 300
FIGURE 7.2: TEMPERATURE CONTOURS AT TIME STEP 400
FIGURE 7.5: CONTOUR PLOT OF ISOCRONS SHOWING PROGRESSION OF SOLIDIFICATION
CHAPTER 8

CONCLUSION

The main objective of this research was to have a better understanding of Computer Aided Engineering tools. Effort has been made to combine two basic features of a CAE system, namely, design and analysis.

For irregular shapes such as a turbine blade, building a 3-D geometry is quite complex. Hence, in the industry the data is stored in form of 2-D sliced sections of the geometry.

In this research a method of creating a finite element model for a complex geometry using 2-D sections has been discussed. The 3-D geometry was built by mapping together 2-D sliced sections. These sliced sections were mapped together depending upon their position and orientation along the stacking axis and the length of the geometry. Modeling of symmetric projected shapes, as well as totally irregular shapes has been demonstrated in this research. This algorithm has been made compatible with MOVIE and the INTERGRAPH, so that the user can view the graphics on the screen.

The second part of this research shows a casting simulation carried out in CAST3, after the model of the blade and root is built in PATRAN. A PATRAN interphase has been incorporated in the algorithm which automatically builds the solid model of the blade in PATRAN. Normally, it would have taken a long time for the model to
be built manually in PATRAN. An option for automatically meshing the model has also been provided in the algorithm.

The results shown in Chapter 7 proves that the whole geometry can be modeled as a single piece casting though the root and the blade have to be created and meshed separately.

This research demonstrates that a healthy blend can be achieved between design and analysis within the framework of CAE. The designer can make use of the tools to create a model that is not only analytically complete, but also time and money saving as compared to experimenting on the actual object.
CHAPTER 9

SUGGESTIONS FOR FUTURE WORK

Many other formulations for the modelling and the display of curves and surfaces can be developed in addition to Bezier and B-spline methods. They include quadric surfaces, natural splines, Coons surfaces and so on. For successful application in CAD, a thorough understanding of the mathematical properties of shape representations is essential. This understanding will help in the design of interactive systems that permit the designer to refine the model to his satisfaction.

Currently the X, Y and Z coordinates of each 2-D section can either be fed in manually or read in through a data file. This might be time consuming and tedious if there are a considerable number of two-dimensional sliced sections. Another option available might be to digitize the drawing.

As demonstrated in the earlier chapters the given problem was first meshed in PATRAN and a casting simulation was run in CAST3 to get the finite element analyses. A different approach can be to forge the blade geometry instead of casting it. The basic stages in the forging a turbine blade involves obtaining the hollow die geometry, billet geometry and the forging parameters. The hollow die geometry can be created using the Blade algorithm.
After the die geometry is obtained, the billet geometry and the forging parameters can be determined.

Note: In practice the blade geometry will be much more complex. Preliminary analysis has to be done regarding die cavity fill, minimization of flash losses, material flow properties and so on. A very important criteria is the position of the flash. The orientation of the blade defines where the flash will be. But if the blade is totally asymmetric and twisted as a complex turbine blade, the determination of the position becomes quite difficult. To ensure uniform fill for the die, the upper and lower die geometry, flash position and directional material flow properties are very important. The various steps can be seen schematically in Flow chart 5.(13)
FLOW CHART 5

3-D MODELLING OF 2-D SLICED SECTIONS

User input
Length of the geometry
number of nodes
number of sections

Mesh up the Billet
in IRM
Obtain the X,Y,Z
coorinates

Input
1. Data file
2. Interactive

Transfer to
neutral file

Hollow Die
Geometry

Transform neutral
file to ALPID
data file

Create ALPID
data file

RUN ALPID
CHAPTER 10

REFERENCES


APPENDIX 1

SAMPLE RUN: REGULAR TURBINE BLADE WITHOUT TWIST
APPENDIX 1

3-D MODELLING OF COMPLEX SHAPES USING 2-D SLICED SECTIONS

by Ardeshir Sholapurwalla

Under the guidance and supervision of:
Dr. J.S. Gunasekera
Department of Mechanical Engineering
Ohio University
Athens, Ohio 45701

INPUT THE LENGTH OF THE GEOMETRY

=>
100.0000

INPUT THE NUMBER OF NODES IN EACH SECTION

=>
46

INPUT THE NUMBER OF ELEM ALONG GEOMETRY

=>
3

IS THE GEOMETRY A SYMMETRIC SOLID PROJECTION, (Y/N)

=>
Y
SPECIFY WHETHER YOU WISH TO INPUT DATA INTERACTIVELY
OR READ FROM A DATA FILE

I : FOR INTERACTIVE DATA INPUT
D : FOR DATA INPUT FROM DATAFILE

ENTER OPTION ACCORDINGLY >>>>

D

NAME OF DATA FILE WITH CO-ORD

CAST.DAT

XCENTER YCENTER

2.87 2.05

INPUT THE SCALING DOWN FACTOR (IF IT IS TO BE REDUCED TO
HALF SCALE INPUT 2)

1

DO YOU WANT TO VIEW INITIAL GEOMETRY

N

DO YOU WANT TWIST IN THE BLADE? (Y/N)

N

CC.CC MOVIE FILE CREATED

FORTRAN STOP
SAMPLE RUN: REGULAR TURBINE BLADE WITHOUT TWIST WITH VARYING Z-DEPTHS
APPENDIX 2

**********************************************
*
*
*
3-D MODELLING OF COMPLEX SHAPES
*
*
*
USING 2-D SLICED SECTIONS
*
*
*
by Ardeshir Sholapurwalla
*
*
*
Under the guidance and supervision of :
*
Dr. J.S. Gunasekera
*
Department of Mechanical Engineering
*
Ohio University
*
Athens, Ohio 45701
*
*
**********************************************

INPUT THE LENGTH OF THE GEOMETRY

=======>
100.0000

INPUT THE NUMBER OF NODES IN EACH SECTION

=======>
46

INPUT THE NUMBER OF ELEM ALONG GEOMETRY

========>
3

IS THE GEOMETRY A SYMMETRIC SOLID PROJECTION, (Y/N)

========>
N
NAME OF DATA FILE WITH CO-ORD

=====>
CAST.DAT

ARE UNEQUAL SECTIONS REQUIRED IN THE GEOMETRY

=====>
Y

NUMBER OF SECTIONAL LENGTHS
(MAXIMUM NUMBER OF SECTIONAL LENGTHS IS 7)

=====>
4

VALUES OF SECTIONAL LENGTHS
(INPUT "0.0" IF NUMBER OF SECTIONAL LENGTHS < 7)

=====>
10
20
30
40
0.0
0.0
0.0

NUMBER OF SUB-SECTIONS IN EACH LENGTH
(IF NO SUB-SECTIONS REQUIRED INPUT "0" FOR EACH SECT. LENGTH)

=====>
0
0
0
0
0
0
0
0
1.0000
1.0000
1.0000
1.0000
1.0000

CC.CC MOVIE FILE CREATED

FORTRAN STOP
APPENDIX 3

SAMPLE RUN: IRREGULAR TURBINE BLADE WITH DIFFERENT 3-D SURFACES
APPENDIX 3

3-D MODELLING OF COMPLEX SHAPES
USING 2-D SLICED SECTIONS

by Ardeshir Sholapurwalla

Under the guidance and supervision of:
Dr. J.S. Gunasekera
Department of Mechanical Engineering
Ohio University
Athens, Ohio 45701

INPUT THE LENGTH OF THE GEOMETRY

100.0000

INPUT THE NUMBER OF NODES IN EACH SECTION

46

INPUT THE NUMBER OF ELEM ALONG GEOMETRY

3

IS THE GEOMETRY A SYMMETRIC SOLID PROJECTION, (Y/N)

N
NAME OF DATA FILE WITH CO-ORD
=====>
CAST.DAT

ARE UNEQUAL SECTIONS REQUIRED IN THE GEOMETRY
=====>
N

DO YOU WANT TO GIVE STACKING POINT AS THE CENTROID OF EACH SECTION?
NOTE: DEFAULT IS THE ORIGIN
=====>
N

DO YOU WANT TO CALCULATE VOLUME OF GEOMETRY
=====>
N

DO YOU WANT TO VIEW THE INITIAL GEOMETRY
=====>
N

NUMBER OF LOOPS NEEDED
=====>
1

IS THE SHAPE A SIMPLE SHAPE
=====>
N
<table>
<thead>
<tr>
<th>SURFACE AREA</th>
<th>X-COORD</th>
<th>Y-COORD</th>
<th>SHORTEST DISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.359</td>
<td>19.7870</td>
<td>16.1030</td>
<td>0.3729</td>
</tr>
<tr>
<td>0.403</td>
<td>19.2900</td>
<td>15.7100</td>
<td>0.5347</td>
</tr>
<tr>
<td>0.069</td>
<td>18.2980</td>
<td>15.5670</td>
<td>0.2439</td>
</tr>
<tr>
<td>23.853</td>
<td>19.1950</td>
<td>16.3656</td>
<td>0.4937</td>
</tr>
<tr>
<td>11.453</td>
<td>18.5440</td>
<td>16.0000</td>
<td>0.0347</td>
</tr>
<tr>
<td>13.773</td>
<td>17.4990</td>
<td>15.6620</td>
<td>0.3327</td>
</tr>
<tr>
<td>64.475</td>
<td>17.5000</td>
<td>14.4500</td>
<td>0.6046</td>
</tr>
<tr>
<td>61.388</td>
<td>17.1370</td>
<td>14.7700</td>
<td>0.4839</td>
</tr>
<tr>
<td>38.017</td>
<td>16.7400</td>
<td>15.0000</td>
<td>0.4588</td>
</tr>
<tr>
<td>89.487</td>
<td>16.0000</td>
<td>12.6500</td>
<td>0.5186</td>
</tr>
<tr>
<td>71.498</td>
<td>15.6500</td>
<td>13.0800</td>
<td>0.5544</td>
</tr>
<tr>
<td>54.039</td>
<td>15.3700</td>
<td>13.3750</td>
<td>0.4067</td>
</tr>
<tr>
<td>98.997</td>
<td>14.0000</td>
<td>10.4600</td>
<td>0.5532</td>
</tr>
<tr>
<td>86.484</td>
<td>14.0000</td>
<td>11.2700</td>
<td>0.5482</td>
</tr>
<tr>
<td>67.050</td>
<td>14.0000</td>
<td>11.7100</td>
<td>0.2973</td>
</tr>
<tr>
<td>94.178</td>
<td>12.0000</td>
<td>8.4500</td>
<td>0.7511</td>
</tr>
<tr>
<td>94.686</td>
<td>12.0000</td>
<td>9.2300</td>
<td>0.5644</td>
</tr>
<tr>
<td>97.915</td>
<td>12.0000</td>
<td>9.4500</td>
<td>0.2200</td>
</tr>
<tr>
<td>90.162</td>
<td>10.0000</td>
<td>6.5700</td>
<td>0.4884</td>
</tr>
<tr>
<td>92.900</td>
<td>10.0000</td>
<td>7.2300</td>
<td>0.4667</td>
</tr>
<tr>
<td>95.716</td>
<td>10.0000</td>
<td>7.3300</td>
<td>0.1000</td>
</tr>
<tr>
<td>85.232</td>
<td>8.0000</td>
<td>4.8600</td>
<td>0.3766</td>
</tr>
<tr>
<td>90.418</td>
<td>8.0000</td>
<td>5.2800</td>
<td>0.3920</td>
</tr>
<tr>
<td>93.696</td>
<td>8.0000</td>
<td>5.3300</td>
<td>0.0361</td>
</tr>
<tr>
<td>82.104</td>
<td>6.0000</td>
<td>3.3200</td>
<td>0.1769</td>
</tr>
<tr>
<td>86.255</td>
<td>6.0000</td>
<td>3.5400</td>
<td>0.2200</td>
</tr>
<tr>
<td>88.806</td>
<td>6.0000</td>
<td>3.5500</td>
<td>0.0100</td>
</tr>
<tr>
<td>77.549</td>
<td>4.0000</td>
<td>2.0300</td>
<td>0.0100</td>
</tr>
<tr>
<td>81.333</td>
<td>4.0000</td>
<td>2.0400</td>
<td>0.0100</td>
</tr>
<tr>
<td>83.941</td>
<td>4.0000</td>
<td>1.9900</td>
<td>0.0500</td>
</tr>
<tr>
<td>71.679</td>
<td>2.0800</td>
<td>1.0700</td>
<td>0.1700</td>
</tr>
<tr>
<td>73.765</td>
<td>2.0500</td>
<td>0.8600</td>
<td>0.2121</td>
</tr>
<tr>
<td>77.326</td>
<td>2.0300</td>
<td>0.6900</td>
<td>0.1712</td>
</tr>
<tr>
<td>67.252</td>
<td>0.1200</td>
<td>0.4000</td>
<td>0.3795</td>
</tr>
<tr>
<td>66.696</td>
<td>0.2300</td>
<td>0.0600</td>
<td>0.3574</td>
</tr>
<tr>
<td>61.386</td>
<td>0.4900</td>
<td>-0.1000</td>
<td>0.3053</td>
</tr>
<tr>
<td>66.701</td>
<td>-1.9500</td>
<td>-0.0400</td>
<td>0.5921</td>
</tr>
<tr>
<td>66.599</td>
<td>-1.8000</td>
<td>-0.5800</td>
<td>0.5604</td>
</tr>
<tr>
<td>65.383</td>
<td>-1.6800</td>
<td>-0.9800</td>
<td>0.4176</td>
</tr>
<tr>
<td>66.758</td>
<td>-4.0000</td>
<td>-0.1800</td>
<td>0.7700</td>
</tr>
<tr>
<td>65.799</td>
<td>-3.8700</td>
<td>-0.9300</td>
<td>0.7612</td>
</tr>
<tr>
<td>63.416</td>
<td>-3.6600</td>
<td>-1.4900</td>
<td>0.5981</td>
</tr>
<tr>
<td>67.362</td>
<td>-6.0000</td>
<td>-0.0200</td>
<td>0.9800</td>
</tr>
<tr>
<td>66.796</td>
<td>-6.0000</td>
<td>-0.9300</td>
<td>0.9100</td>
</tr>
<tr>
<td>68.931</td>
<td>-6.0000</td>
<td>-1.7100</td>
<td>0.7800</td>
</tr>
<tr>
<td>69.565</td>
<td>-8.0000</td>
<td>0.5100</td>
<td>1.0657</td>
</tr>
<tr>
<td>68.326</td>
<td>-8.0000</td>
<td>-0.5900</td>
<td>1.0917</td>
</tr>
<tr>
<td>67.288</td>
<td>-8.0000</td>
<td>-1.5400</td>
<td>0.9476</td>
</tr>
<tr>
<td>72.578</td>
<td>-10.0000</td>
<td>1.2900</td>
<td>1.1646</td>
</tr>
<tr>
<td>70.717</td>
<td>-10.0000</td>
<td>0.0300</td>
<td>1.2054</td>
</tr>
</tbody>
</table>
SURFACE DEFINITION
TYPE 1 FOR CREATING MOVIE DATA FILE
2 INCORPORATING TWIST IN SECTION
3 FITTING A BSPLINE CURVE/DFPI
4 FITTING A BEZIER CURVE/MOVIE OUTPUT FILE

DO YOU WANT TO CALCULATE VOLUME OF GEOMETRY

N
NAME OF MOVIE DATA FILE INTO WHICH CO-ORD ARE TO BE WRITTEN


CC.1

MOVIE FILE CREATED

DO YOU WANT TO CREATE A DIFFERENT SURFACE FOR THE SAME GEOMETRY


Y

SURFACE DEFINITION
TYPE  1 FOR CREATING MOVIE DATA FILE
      2  INCORPORATING TWIST IN SECTION
      3  FITTING A BSPLINE CURVE/DFPI
      4  FITTING A BEZIER CURVE/MOVIE OUTPUT FILE


4

DO YOU WANT TWIST IN THE SECTION, (Y/N)


N

DO YOU WANT TO CALCULATE VOLUME OF GEOMETRY


N

NUMBER OF NODES INCREASED PER SECTION = 0

HIT RETURN TO CONTINUE <RET>
NUMBER OF CONTROL POINTS?

=>

4

NODES AND INCREASE IN NUMBER OF NODES PER SECTION

=>

46

0

NUMBER OF CONTROL POINTS?

=>

4

DESIRED NUMBER OF INCREMENTS

NOTE: 1/5 GIVES 6 POINTS ALONG THE LENGTH OF THE GEOMETRY

# HAS TO BE MORE THAN THE NUMBER OF 2-D SECTIONS

=>

0.1

NAME OF DATA FILE INTO WHICH CO-ORD ARE TO BE WRITTEN

=>

CC.2

MOVIE FILE CREATED

DO YOU WANT TO CREATE A DIFFERENT SURFACE FOR THE SAME GEOMETRY

<=

N

FORTRAN STOP
APPENDIX 4

SAMPLE RUN: SOLID MODEL OF TURBINE BLADE GEOMETRY IN PATRAN
RUN CHK1
PROGRAM TO INPUT X,Y,Z CO-ORDINATES INTO PATRAN

NAME OF THE DATA FILE FROM WHICH DATA IS TO BE READ
====>
CAST.DAT

NAME OF DATA FILE INTO WHICH YOU WANT PATRAN COMPATIBLE DATA TO BE WRITTEN
====>
ADI.DAT

LENGTHS ALONG THE Z-DIRECTION
====>
1
2
3
4
5

NUMBER OF NODES PER SECTION
====>
46

NUMBER OF 2-D SECTIONS ALONG THE LENGTH OF THE GEOMETRY
====>
DO YOU WANT TO MESH THE GEOMETRY?

=> Y

DO YOU WANT TO CREATE MESH BY ISO. OR NUM. OPTION?

   TYPE 1 FOR ISOPARAMETRIC OPTION
   2 FOR NUMBER OPTION

=> 2

ELEMENT TYPE SPECIFICATION

   TYPE 1 FOR WEDGE
   2 FOR HEX

=> 2

CREATING MESH FOR CENTER HYPERPATCH

ENTER THE NUMBER OF ELEMENTS ALONG EDGE 1

=> 3

ENTER THE NUMBER OF ELEMENTS ALONG EDGE 2

=> 1

ENTER THE NUMBER OF ELEMENTS ALONG EDGE 3

=> 3
ENTER THE NUMBER OF ELEMENTS ALONG EDGE 4
=====>
1
ENTER THE NUMBER OF ELEMENTS ALONG NORMAL
=====>
2

CREATING MESH FOR CORNER HYPERPATCHES

ENTER THE NUMBER OF ELEMENTS ALONG EDGE 1
=====>
1
ENTER THE NUMBER OF ELEMENTS ALONG EDGE 2
=====>
3
ENTER THE NUMBER OF ELEMENTS ALONG EDGE 3
=====>
1
ENTER THE NUMBER OF ELEMENTS ALONG EDGE 4
=====>
3
ENTER THE NUMBER OF ELEMENTS ALONG NORMAL
=====>
2

FORTRAN STOP