Tetrahedra to Hexahedra Conversion for Finite Element Analysis

by

Alejandra Carmona García, Production Engineer

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario

Canada

© Copyright 2002

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-79692-2

Canada

The undersigned hereby recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis,

Tetrahedra to Hexahedra Conversion for Finite Element Analysis

submitted by

Alejandra Carmona García, Production Engineer

in partial fulfilment of the requirements for
the degree of Master of Applied Science

Thesis Supervisor

Chair, Dept. of Mechanical and Aerospace Engineering

Carleton University
June 2002

Abstract

Hexahedral elements have proven to provide superior results to tetrahedral elements in certain types of complex finite element analysis (FEA). However, the development of a robust and complete hexahedral mesh generation scheme has been problematic. A need for tetrahedra-to-hexahedra (TTH) conversion software is identified, particularly for specialty codes not associated with typical CAD/CAM packages.

The development and testing of TTH software is presented. It uses the splitting method, a simple operation of dividing each tetrahedron into four hexahedra. Several mesh optimization routines are implemented at various stages of the program to improve the overall quality of the mesh.

As a whole, TTH is seen to have the capability of handling a variety of geometries incorporating features typical of FEA analysis. Although TTH cannot completely eliminate poorly formed elements the mesh optimization routines are effective at improving the overall quality of the mesh.

Acknowledgements

I would like to express my appreciation and gratitude to the people who have supported me throughout this project:

To my advisor Professor Moyra McDill, for her patience and support during these years. For her valuable teachings and countless encouragement to participate more actively in various aspects of the engineering field.

To my friends for their generous advice, and my most special gratitude to Jerome for his unconditional assistance.

To my parents who have been always supportive through the distance.

Table of Contents

Abstract	iii
Acknowledgements	iv
Table of Contents.....	v
List of Tables	vii
List of Figures	vii
List of Appendices.....	viii
Nomenclature	ix
1 Introduction.....	1
2 Background	5
2.1 Finite Element Analysis Process.....	5
2.2 Meshing	7
2.2.1 Mesh Generation.....	9
2.2.1.1 Mesh Generators Classifications	10
2.2.1.2 Tetrahedral vs. Hexahedral Elements	12
2.3 Mesh Improvement	13
3 Literature Review	18
3.1 Mesh Generators vs. CAD/CAM Packages.....	19
3.2 Direct and Indirect Element Generation Schemes	21
3.2.1 Direct Element Generation Scheme	21
3.2.1.1 Mapping	22
3.2.1.2 Sweeping and Mesh Extrusion	25
3.2.1.3 Grid-based	26
3.2.1.4 Plastering	27
3.2.1.5 Whisker Weaving.....	28
3.2.2 Indirect Element Generation Scheme	29
3.2.2.1 H – Morph.....	30
3.2.2.2 Splitting (TTH).....	31
3.3 Mesh Quality Assessments	32
3.4 Mesh Smoothing	34
3.4.1 Opti-Smoothing.....	35
3.4.2 Weighted Area.....	36
3.4.3 Laplacian.....	36
3.4.4 Isoparametric	37
3.4.5 Equipotential	40
3.4.6 Untangling and Geometric-Based Smoothing	40
4 Element Splitting and Quality Assessment.....	41
4.1 Programming Language	41
4.2 Data-Gathering	44
4.3 Splitting.....	44
4.3.1 Data Structures	46
4.4 Tetrahedral Centroid	49
4.5 Tetrahedral Quality Assessment	49
4.6 Hexahedral Quality Assessment	51
5 Smoothing	54

5.1 Hexahedron Quality Classifications	54
5.2 Geometry-Based Technique	55
5.2.1 AR Fix	56
5.2.2 Skew Rotate	59
5.3 Isoparametric Technique	62
5.3.1 Element Verification	64
6 Testing and Results	67
6.1 Test Cases	67
6.2 Mesh Quality	69
6.2.1 Improvements to Existing Tetrahedra (Pre-Splitting)	71
6.2.2 Improvements to Hexahedra (Post-Splitting)	73
6.2.2.1 Skew Rotate	74
6.2.2.2 AR Fix	75
6.2.2.3 Isoparametric	78
6.3 TTH Meshes	80
6.4 Multi-Platform Capabilities	82
7 Discussion	86
8 Conclusions and Future Work	88
8.1 Conclusions	88
8.2 Future Work	89
9 References	90

List of Tables

Table 3-1 Auto generating capabilities of FEM programs.....	20
Table 3-2 List of mesh assessment functions and relevant information	33
Table 6-1 Test cases mesh densities	69
Table 6-2 Computer specifications.....	83
Table 6-3 Runtime tests	84

List of Figures

Figure 2-1 Examples of structured meshes	8
Figure 2-2 Examples of unstructured meshes.....	8
Figure 2-3 Mapped mesh	10
Figure 2-4 Free meshing.....	11
Figure 2-5 FEM process	17
Figure 3-1 Volume map	23
Figure 3-2 Sweeping technique	26
Figure 3-3 Plastering meshing algorithm.....	28
Figure 3-4 STC of a block	29
Figure 3-5 Tetrahedron to hexahedron conversion	32
Figure 3-6 Laplacian method.....	37
Figure 3-7 Isoparametric smoothing method.....	39
Figure 3-8 Comparison between Laplacian and isoparametric smoothing methods	39
Figure 4-1 Tetrahedra-to-hexahedra (TTH) process.....	43
Figure 4-2 Linear tetrahedron with additional nodes.....	45
Figure 4-3 Typical splitting.....	45
Figure 4-4 Computational coordinates of a tetrahedron	47
Figure 4-5 8-Node, isoparametric volume element	51
Figure 4-6 Computational coordinate axis in global space	53
Figure 5-1 AR Fix procedure	57
Figure 5-2 AR Fix application	58
Figure 5-3 Skew Rotate procedure	60
Figure 5-4 Skew Rotate application	61
Figure 5-5 Connecting nodes for isoparametric calculations	63
Figure 6-1 Wedge	68
Figure 6-2 Cylinder.....	68
Figure 6-3 Throw	68
Figure 6-4 Stationary platen	69
Figure 6-5 Wedge TTH mesh	70
Figure 6-6 Cylinder TTH mesh	70
Figure 6-7 Mesh quality comparison	72
Figure 6-8 Interior node relocation.....	73
Figure 6-9 Percentage of elements in the mesh vs. skew angles	75
Figure 6-10 Number of elements (%) vs. aspect ratio of hexahedra	76
Figure 6-11 Comparison of the element aspect ratios after first smoothing techniques	78

Figure 6-12 Number of elements vs. skew angles after smoothing procedures	80
Figure 6-13 Throw TTH mesh	81
Figure 6-14 Stationary platen mesh after TTH.....	81
Figure 6-15 Smoothing process on individual elements.....	82

List of Appendices

Appendix 1 Node translation direction with the different skew angles changes.....	95
Appendix 2 Development of the 3-D isoparametric equation	96
Appendix 3 TetCentroid Function	99
Appendix 4 TTH user manual.....	100

Nomenclature

a, b, c	Nodal positions of the corner nodes in a 2-D parallelogram
a_r	Position of the connected adjacent node in the 2-D r^{th} neighbour element
b_r	Position of the opposite corner node in the 2-D r^{th} neighbouring element
c_r	Position of the connected adjacent node in the 2-D r^{th} neighbour element
$\{f\}$	Vector of nodal forces
i	Summation counter or subscript
$[J]$	Jacobian matrix
$[k]$	Stiffness matrix
NE	Number of neighbour elements connected to a node
N_i	Interpolation functions
$\{q\}$	Vector of nodal displacements
r	Summation counter or subscript
S_i	Length of any edge i
S_{rms}	Root mean square (S_i) ($i=1...6$)
V	Tetrahedron volume
w	Weighting factor in isoparametric smoothing
x	Coordinates of the nodal point to be smoothed

X	Coordinates in global space
x'	obtained x-coordinate value after smoothing
x_A, x_B, x_C	Adjacent node to the intended smoothing node
x_D	Opposite corner node to the element smoothed node
x_i	x-coordinate of the nodal point
Y	Coordinates in global space
y_i	y-coordinate of the nodal point
Z	Coordinates in global space
z_i	z-coordinate of the nodal point
γ	Tetrahedral aspect ratio
γ_N	Normalized tetrahedra aspect ratio
η	Coordinates in the computational domain
$\theta_{\xi\eta}$	Skew angle in global space formed between $\xi - \eta$ axis
$\theta_{\eta\zeta}$	Skew angle in global space formed between $\eta - \zeta$ axis
$\theta_{\zeta\xi}$	Skew angle in global space formed between $\zeta - \xi$ axis
ξ	Coordinates in the computational domain
ζ	Coordinates in the computational domain

1 Introduction

The finite element method (FEM) is perhaps the most popular numerical technique for the analysis of engineering designs with complicated geometry and/or non-homogeneous material properties. It uses a computer model to approximate the mechanical behaviour of components under various working environments. Future development or refinements of products are often driven by the results obtained from finite element analysis (FEA). New product designs often employ FEA to verify performance and specifications prior to manufacturing or construction. This analysis is also used to qualify new service conditions for the product or structure when modifications are made.

Typically FEA uses two-dimensional or three-dimensional models. Two-dimensional (2-D) analysis tends to be less accurate due to its simplicity, but relatively low computational requirements are needed. Three-dimensional (3-D) models, on the other hand, produce more accurate results, but require a higher computational effort.

A completely defined FEA model requires geometric data, topological information and boundary conditions. Boundary conditions govern the loads and constraints, such as temperatures, pressures, forces, moments, or accelerations, that each component or system of components experiences in its working environment. Geometrical data

represent the position and shape of the geometries, while the topological data specify the relations between objects [1].

Geometric data and topological data are typically embedded in the description of the mesh. Consequently the accuracy of the analysis is highly dependent on the quality and characteristics of the given mesh. Mesh generation techniques are continuously being refined and improved to ensure that high quality meshes with good characteristics are consistently achieved. Currently much of the research in this area is focused in the development of automated 3-D hexahedral mesh generation algorithms, which at present are still not as robust as the software available for the generation of meshes of tetrahedral elements [2]. Tetrahedral mesh generation is easier than hexahedral mesh generation, especially for complex geometries that contain holes and interior voids [3].

While tetrahedral elements are generally accepted for most analyses, there are instances for which hexahedral elements are preferred. For example, in simulations of manufacturing processes such as welding and flame cutting where the simulation is both time dependent and has large displacements [4], linear hexahedral elements perform better in capturing the temperature gradient than linear tetrahedral elements. Hexahedral elements are also preferred for dynamic analysis because tetrahedral elements are often too stiff [5]. For these applications specialty codes such as TIAMAT¹

¹ TIAMAT is a multipurpose code for the analysis of 3-D thermal-elasto-plastic problems developed at Carleton University.

[6], SIMPLE² [7], ABAQUS³ [8] and DYNA⁴ [9] have been developed to handle meshes which include hexahedral elements. Commercial CAD packages (I-DEAS⁵ [9] and Pro/ENGINEER⁶ [9]) often allow users to export the geometric information in a format compatible with commonly used specialty codes such as the ones named above. The challenge is that these codes are designed to populate the geometry with tetrahedral elements, but require significant user effort to generate hexahedral meshes.

Hexahedral meshes in most commercial codes are often generated using the mapped meshing method. Mapped meshing of hexahedral elements is a labour intensive process. It requires the user to identify key areas and, typically, the distribution of hexahedral elements within these regions [5]. Free meshing requires minimal user involvement, but it is usually reserved for tetrahedral elements. Clearly, there is a place for an algorithm which can convert a free-meshed tetrahedral mesh to a hexahedral mesh with limited user involvement. This algorithm should exploit the existing topology and geometry with no loss in geometric detail.

This thesis presents the design, development and testing of an algorithm which converts free-meshed tetrahedral elements produced by two commercial codes, I-DEAS and Pro/Mechanica, into hexahedral elements. This 3-D converter has been created as an

² SIMPLE is a multipurpose code for the analysis of 3-D thermal-elasto-plastic 2-D and 3-D problems, developed the Technical University of Luleå

³ ABAQUS is a general-purpose non-linear FEA program.

⁴ DYNA is an explicit FEA code typically used for non-linear dynamic problems.

⁵ I-DEAS is a multipurpose CAD/CAM/CAE software with integrated FEM modules.

⁶ Pro/ENGINEER is a multipurpose CAD/CAM/CAE software with integrated FEM modules (Pro/Mechanica).

independent hexahedral mesh generator for specialty codes such as TIAMAT, SIMPLE, ABAQUS and DYNA, for modelling complex phenomena such as welding and impact.

Chapter 2 of this thesis presents background information related to FEA. In Chapter 3, previous literature related to the conversion of tetrahedral elements to hexahedral elements as well as the methodology used in mesh smoothing are presented. Chapters 4 and 5 describe the specific method adopted in this thesis for tetrahedral division and mesh smoothing respectively. Chapter 6 provides results and discussions of various tests performed to convert the data from the commercial codes used in this thesis. Thesis discussion, and thesis conclusions and recommendations for further work are provided in Chapters 7 and 8 respectively.

2 Background

It is useful to review the related background information on FEA and to highlight some procedures required in the pre-processing stage, which is the main interest of this project. A general discussion of FEA processes will be introduced first, followed by a presentation on mesh generation techniques, element types, and various mesh improvement methods.

2.1 Finite Element Analysis Process

There are three key components in any FEA: a pre-processor, processor, and a post-processor. The pre-processor provides the interface between the user and the processor. It comprises of a variety of utilities for geometric definition, manipulation, material definition, element properties, generation of meshes, boundary condition constructing schemes and data file compilation required by the solver.

In FEA, pre-processing is a critical step because it has a substantial impact on the accuracy of the solution and, subsequently, on the cost of the analysis. It is common for the pre-processing stage to consume as much as 80 percent of the time involved in the finite element process [10].

A processor or solver is an integral part of the FEM process. Its purpose is to calculate

the reaction or response of the system based on the stiffness matrix and the input parameters. The stiffness matrix $[k]$ relates the vector of element nodal forces $\{f\}$ to the vector of nodal displacements $\{q\}$, as shown [11]:

$$[k]\{q\} = \{f\} \quad [2-1]$$

FEA solvers are generally categorized into implicit or explicit solvers, each of which has its own specialty and limitations, that should be used based on the problem type. Implicit solvers are ideal for steady state problems where linear equations are relatively simple to form and can be solved with few difficulties. Their efficiency deteriorates when used in non-linear problems, where equilibrium of the linear equations has to be acquired iteratively. The nature of implicit solvers allows users to analyze these problems with larger time steps. The concerns with this combination are the convergence speed and the possibilities of diverging results, which might lead to unstable and unusable solutions [12].

Explicit solvers ensure convergence by means of direct time integration, which avoids the problem caused by the iterative nature in implicit methods. Non-linear and time-dependent investigations such as crash-worthiness and sheetmetal-forming processes favour the explicit method due to its stability. However, the time steps required are generally small to ensure a reliable solution, and in some cases, parameters are artificially modified to improve the efficiency of the analysis [12].

The post-processor takes the data from the solver and presents it to the user in a more

representative fashion, e.g. scalars can be plotted as colour contours or fringes, vectors as arrows with magnitude and direction. An important feature of a post-processor is the total user control of the graphical outputs.

At this point, steps are often taken to improve the quality of the analysis by refining the element types, meshes and/or boundary constraints.

2.2 Meshing

The efficiency and accuracy of any analysis are directly related to the size and shape of the elements within the mesh. This is why much attention is placed on the development of meshing. There are two main classifications of meshes, structured and unstructured.

A structured mesh is a grid that has a regular lattice arrangement for all its elements [13]. Figure 2-1 is an example of a 2-D structured mesh. Each node in the grid is at an intersection of two or three curves in 2-D or 3-D space respectively. Structured meshes are used extensively by the finite difference community due to their ease of management in the element topology database. Depending on the dimensions of the mesh, all grid points require only two or three parameters to be completely defined. The cumbersome relational information between elements is less significant. This type of grid has the convenience of fast identification of neighbouring cells, which is ideal for a finite element approach [14]. The drawback of this approach is the rigidity of the lattice arrangements, which makes modelling of irregular and complex geometry difficult.

Figure 2-1 Examples of structured meshes

Unstructured meshes on the other hand are ideal for modelling complicated objects especially when minimum angle and area conditions exist.

Figure 2-2 shows some examples of a 2-D representation of unstructured meshes. Unstructured meshes require extensive topological information to be useful in finite element simulations. The information necessary to facilitate an analysis includes connection between elements, edges, and coordinates of the nodes. Each cell in an unstructured grid and its connections to adjacent cells are defined separately, which means all the variables have to be categorized and stored. These complex data structures require specialized computational algorithms, and are computationally intensive for complex objects.

Figure 2-2 Examples of unstructured meshes

2.2.1 Mesh Generation

Inexpensive high performance computers have revolutionized engineering, especially in the FEA field, which has been greatly improved by the advances made in automatic mesh generation.

Automatic mesh generation is a common feature in most FEM packages. It is an algorithm designed to generate meshes exclusively from computer-aided design (CAD) models. The aim is to reduce the time and labour required by the tedious nature of mesh generation. Its drawback is its insensitiveness to specific mechanical data (i.e., elements selection, boundary conditions (BC), mechanical behaviour), which affect the accuracy of the simulations [15].

The selection of elements used by mesh generators depends on the problem, domain, and elements available in the specific FEA package. In general, 2-D (triangular and/or quadrilateral)⁷ and 3-D (tetrahedral and/or hexahedral) elements can be chosen [16]. This project focuses on the 3-D aspects of FEM, where development opportunities are still available and needed. 3-D models are necessary when complex geometries have a low aspect ratio and cannot be idealized as planar models or when applied loads or material response occur in three dimensions. These models provide more realistic representation of the conditions and behaviour of the solid objects in their working environment [5].

⁷ Axisymmetric elements are a special class of 2-D elements.

2.2.1.1 Mesh Generators Classifications

Rank et al. [17] using the classification terminology of Ho-Le [1] classified most of the mesh generators into two groups: macro-element techniques and free meshing techniques.

The macro-element technique, commonly referred to as mapped meshing, is used in regions most frequently described by eight-node isoparametric elements. Elements are meshed automatically into a more or less brick-shaped element pattern as shown in Figure 2-3. The main advantage of mapped meshing is the capacity to easily generate completely hexahedral meshes. Brick-shaped elements are preferred because they demonstrate improved accuracy over the corresponding tetrahedral elements [18].

Figure 2-3 Mapped mesh [19]

The difficulty with mapped meshing is in generating strongly graded meshes on arbitrary domains. The generation of a graded mesh using this method is possible, but considerable time is required to decompose a general geometry into simpler regions and

apply the necessary embedded external constraints [17]. Figure 2-3 is a practical example of graded meshing; notice the fine mesh around the centre hole (counter-bore) and sparse grid in the surrounding areas. The transition from fine to coarse mesh requires irregular midedge node elements [4] or externally applied constraints [9][20], and extra attention to ensure the validity of the mesh. Despite this drawback, most commercial mesh generation programs available today rely on mapped methods to generate hexahedral meshes. [21].

Free meshing has fewer geometric restrictions, which makes it more flexible than mapped meshing. It can be defined on any shapes, surfaces, curves, edges, sections or complicated volumes [3]. The main disadvantage of the free meshing technique is that the geometry is populated with tetrahedral elements. An example of free meshing is shown in Figure 2-4.

Figure 2-4 Free meshing

In general, mapped meshing provides more user control than free meshing but is more time consuming [10]. Mapped meshing requires the user to specify or force a particular

mesh pattern. The mapped meshing requirements for a complicated geometry could be labour intensive. Both techniques have been adopted in mesh generating routines in almost every commercially available FEM packages.

2.2.1.2 Tetrahedral vs. Hexahedral Elements

The majority of mesh generators developed today are for tetrahedral meshing. As of the current moment, the author's research was not able to identify a commercial program that is capable of varied and robust automatic hexahedral meshing [22]. Intense efforts in both industrial and academic communities are being made to develop a robust hexahedral generation scheme. Although all commercial FEM pre-processors are capable of some form of automatic generation of hexahedral elements there are strict limitations. For instance, the sweeping technique available in CUBIT⁸ can only handle a limited class of geometries [23].

The drawback related to the algorithmic complexity of hexahedral generation is small when compared to its advantages. The advantage of linear hexahedra is their ability to deform in a lower strain energy state, which often makes them more accurate [18], [24], [25], [17], [21]. It has been shown that linear hexahedra can out-perform quadratic tetrahedral elements in plastic deformation situations [18]. Thus, a fully automatic discretization of 3-D geometries into fully hexahedral meshes remains an open issue and the subject of current research.

The lack of automatic hexahedral mesh packages that can provide a high quality solid

⁸ CUBIT is a software for 2-D and 3-D finite element mesh generation.

brick has made the use of linear tetrahedra more popular. 3-D domains can easily be represented into a collection of tetrahedral elements due to fewer geometrical constraints, but the same geometry might not be always decomposable into brick elements [5].

Linear tetrahedra are commonly referred to as constant-strain elements. They tend to behave in an overly stiff manner [5]. Under displacement-dominated conditions, the error in using tetrahedral elements is much larger than using hexahedral elements [26]. Therefore, the mesh density can be lower if hexahedral elements are used. In addition, more tetrahedra than hexahedra are needed to capture the behaviour of rapid change in a gradient [24].

In summary, hexahedral elements are often preferred over tetrahedral elements [18]. Expert analysts state that the additional time and effort required to prepare models with hexahedral elements is justifiable for improving the quality of the simulation results [27].

2.3 Mesh Improvement

Areas that might be problematic to any untested structure are often unknown at the start of most analyses. It would not be efficient to apply a fine mesh throughout the entire geometry for initial examination, because the high computational cost is not justifiable. In general, an experienced analyst would approach the problem with a coarser mesh to understand the mechanical behaviour of the structure. Refinements would then be made locally or globally based on the needs. Mesh enhancing techniques

such as conversion of elements, refinement, smoothing and coarsening are frequently used to obtain an optimal mesh [16]. An optimal mesh is defined by Kittur et al. [28] as a mesh in which all elements contribute equally to the error energy norm. For Kittur, the criteria for a 'good mesh' are based on the strain energy, displacement, and stress values at selected critical points of a structure.

Tetrahedral elements are perhaps the most widely utilized elements in 3-D analysis. It has been reiterated that tetrahedral elements have fewer constraints in meshing arbitrary geometries and are available in almost all free meshing codes [29][30][31]. Tetrahedral elements are often sufficient in capturing the result for preliminary analysis, but for detailed analysis they might become inadequate. A simple remedy is to increase the element order from linear to a higher order such as quadratic. Higher order elements have better characteristics in capturing sharp changes in gradient, but the computational cost for higher order elements can be substantial. Therefore, conversion to higher order elements or more efficient and accurate elements such as hexahedral elements is one of the approaches for mesh optimization. Element conversions are often performed globally to improve the overall accuracy of the mesh. Although local element substitution is possible, irregular elements are often needed to provide a proper transition zone. Irregular elements are not common feature in FEM packages, which make local element substitution difficult.

Another method for mesh optimization is mesh refinement, which requires more user interaction. There are three techniques available for mesh refinement: h-refinement, p-refinement, and hp-refinement. The goal for mesh refinement is to achieve the desired

mesh density distribution by adjusting the mesh characteristic dimension and the element shape order. In h -refinement the characteristic dimension, h , is decreased in areas where fine meshes are required. P -refinement involves increasing the order of the polynomial shape functions in the regions with high error, but the number of elements in the region remains constant. HP – refinement means that both the number of elements and the order of the shape functions are increased to reduce the error in a particular neighbourhood. Mesh refinements are generally performed in localized regions of the mesh. Its intention is to improve the accuracy in the area of interest, without incurring a heavy penalty on computational efficiency.

The methods described thus far have been aimed at mesh refinement of a larger scale. Errors associated with the analysis are not necessarily contributed solely from the mesh. The amount of distortion in individual elements also contributes to the problem. For instance, an ideal hexahedral element is a cube, which has perfect aspect ratio and has no skew so all angles are 90° . It is often difficult, if not impossible, to define a problem using an exclusively cubic hexahedral. Smoothing techniques are used in an attempt to regularize distorted elements. The basic idea is to improve the element by shifting its nodes. Smoothing is usually performed iteratively and globally.

Another class of mesh improvement technique is re-zoning; it is mainly used in problems that involve time dependent localized phenomena. Manufacturing processes such as laser line heating and sheet forming generate localized phenomena because they induce deformation in a localized region. For simulation purposes, re-zoning mesh manipulation techniques are often employed to analyze such processes. It is a technique typically

developed to generate a fine grid near the source, where most of the changes are taking place. Sparse grids are meshed in the remaining area, where minor changes occur. The meshes are updated at every time step or at specific time steps to coincide with the source. The information from the previous time step is used to compile the initial data for the new time step. The advantage of such mesh manipulation techniques is that it captures the dynamic characteristics of such problems, and the computational costs are kept to a minimum [32].

While mesh refinement is well understood, mesh-coarsening can also be used since maintaining a fully refined mesh is expensive. For example, in h-coarsening many small elements are replaced with fewer large elements. The assembly technique proposed by McDill and Oddy [33] is one example of the h-refinement and h-coarsening solutions to reduce the computational costs.

As it was previously stated, this thesis presents the design ideology, development method and testing process of an algorithm that converts free meshed tetrahedra into hexahedral elements. The main interest of the thesis is to provide an acceptable hexahedral mesh, which provides results of sufficient accuracy at a reasonable cost. The algorithm is designed to work in tandem with existing FEM pre-processors. Figure 2-5 shows how the FEM process in general relates to this specific project. The outline in bold shows the areas of research considered in the project.

Figure 2-5 FEM process

3 Literature Review

At the early stage of the automatic mesh generation development, researchers were focused on efficiently discretizing 2-D domains into triangular elements. Then, quadrilateral meshes showed better performance than triangular elements, which persuaded the research community to develop techniques that focused on quadrilateral generation [34]. The evolution has been similar in 3-D mesh generation, where tetrahedral elements are often found to be inferior to hexahedral elements [35]. Current research on 3-D applications is mainly concentrated on the discretization of the domain into hexahedral elements. The challenge is the difficulty of placing well-shaped hexahedral elements into a 3-D domain. Although some novel ideas have contributed to considerable progress in the area, none of them is at the stage of maturity that is comparable to free meshing of tetrahedral elements. The main hurdle is the robustness of the original concepts for automatic generation of hexahedral elements for arbitrary volumes. The generation of hexahedral elements continues to be a current topic of study [35].

This chapter details some of the meshing capabilities of the software available in today's market. Also, details of methods used by element generation schemes for hexahedral elements are presented. In addition, topics related to mesh quality improvement are also included. These are essential for minimizing the distortion of the hexahedral

elements which, in turn, improves the quality of the analyses.

3.1 Mesh Generators vs. CAD/CAM Packages

CAD/CAM packages are routinely designed with a wide variety of extra capabilities to allow the product development process to proceed through different design phases on a single platform. The comprehensive approach of the packages allows a product to be developed in a more efficient and organized manner. The principal contribution of integrated CAD/CAM packages has been the elimination of the exchange of data between various phases of the design process. Although there is a wide range of applications available in CAD/CAM packages, they are not necessarily adequate for all circumstances. For instance, the mesh generation in CAD/CAM integrated modules is usually limited to tetrahedral meshes and hexahedral meshes in primitive geometries [36]. The demand for hexahedral elements prompted the continual development of specialized programs for mesh generation.

Specialized mesh generation codes are often incorporated in the pre-processor stage of FEA programs. The mesh generation codes contain multiple tools for a wider range of meshable geometries. Specialty codes in general are expensive and not easy to use, but are capable of producing suitable meshes to a variety of FEA applications such as thermal and elastic analysis.

A diverse variety of mesh generation software has been offered in the market; e.g., MSC, MARC, HyperMesh, FEMAP, ANSYS, COSMOS, and recently CUBIT. CAD/CAM

packages have their specialized FEA utilities such as SDRC from I-DEAS, Pro/MESH from Pro/ENGINEER, and Elfini from CATIA. In general, the majority of mesh generators and CAD/CAM FEA modules have similar tools for meshing and element selections; i.e., line, shell and solid auto meshing. A brief comparison of current solid meshing capabilities of some of the mentioned codes is shown in Table 3-1. FEM programs in general offer more than just meshing tools and element libraries. They also include element quality validation routines to ensure element accuracy. Mesh smoothing and refining functions are common in these packages for mesh quality improvement.

Table 3-1 Auto generating capabilities of FEM programs [5]

Mesh	HyperMesh	FEMAP	Pro/MESH	SDRC/I-DEAS	MSC/PATRAN	COSMOS/MgeoStar	ANSYS/PrepPost	Pro/MECHANICA	MARC
Automesh, solids									
Tet	Y	Y	Y	Y	Y	Y	Y	Y	Y
Tet/wedge ⁹	N	N	N	N	N	N	N	Y	N/A
Hex	N	N	N	N	N	N	N	N	Y
Hex/wedge	N	N	N	N	N	N	N	N	N/A

Conventions:

Y : Entity of the table indicates that the product has this capability

N : The product does not have the capability.

N/A : Information not available

Currently there are few packages that have the capability for auto meshing hexahedral elements. For example, Pro/Mesh, which is an integrated meshing program for the popular commercial CAD/CAM software Pro/ENGINEER, is not capable of generating a

⁹ Wedge Element: A triangular protrusion with a node at each of its six vertices.

pure hexahedral mesh. Another example, CUBIT, which is a mesh generator for the CFD program FLUENT, is capable of generating a fully hexahedral mesh but for only limited classes of geometries.

3.2 Direct and Indirect Element Generation Schemes

The majority of the automatic meshing algorithms for quadrilateral/hexahedral elements developed today are of direct generation class. The direct generation scheme requires the elements to be generated in a domain without any pre-existing mesh. Indirect generation on the other hand, utilizes a pre-existing tetrahedral meshed domain, and systematically transforms the tetrahedral elements into hexahedral elements. Indirect schemes are able to take advantage of local topological information. A background tetrahedral mesh is more effective in achieving a rapid gradation effect over irregular and complicated domains. The implementation of a robust direct hexahedral generation scheme is currently inhibited by the complexity of the issue, which will be further illustrated in the following section.

3.2.1 Direct Element Generation Scheme

Some of the most common mesh generating algorithms used for creation of hexahedral meshes are mapping and sweeping methods. Elements generated either individually or in patches employing mapping techniques are of better quality and contain fewer irregular nodes than the elements generated by other techniques [14][34]. The processes of manual decomposition of the geometries into regular primitive shapes are

critical to mapped meshing [37]. Such techniques require the users to have extensive experience to be able to decompose the geometry effectively, and even that would not guarantee the process to be efficient, simple or quick.

3.2.1.1 Mapping

By definition a mappable volume must have six logical sides and eight logical vertices; each side can consist of one or several geometric surfaces [23]. Figure 3-1 (a) shows a typical six-sided volume that is considered optimal for mapped meshing. This scheme meshes a volume with a structured mesh of hexahedral elements based on the above requirements. Elements in simple cases are straightforward to create especially for domains with blocky features. Even shapes with rounded edges, which are often used in today's designs, can be easily handled using this method. Since the vertices can be placed on curves, as illustrated in Figure 3-1(b), this allows volumes of less than eight vertices to be mapped [8]. Figure 3-1(b) shows a typical five-sided volume that is common in structures.

The greatest disadvantage in using the mapping process is that the mesh generator cannot handle anything beyond simply connected convex domains. Domains that contain concave or internal openings must be broken down into simple connected convex regions. The process of converting a complex region into simply connected convex regions for abstract geometries is extremely tedious and time consuming. It often results in a large number of small partitioned regions that require fine meshes. There are several reasons why extremely fine meshes may be counterproductive. First,

the fine meshes might grossly exceed the requirement to obtain satisfactory results in that particular region, and in turn consume more valuable computational time than normally required. Other considerations are the limits imposed by the FEM codes; for example, many codes limit the number of nodes and/or elements to be analyzed.

Figure 3-1 Volume map (a) six sided volume and (b) five sided volume [23]

The second drawback of having a fine mesh in multiple small regions is the problem of grading. Grading in such a rigid conditioned environment is often difficult due to the problems arising from the transition between fine and coarse grids. Small partitioned areas restrict the element size available for the transition. Since these processes are usually performed manually, mapped meshing for complex concaved domain with voids is a long and tedious process.

Submap

Submapping is a meshing tool developed to extend the capabilities of mapping technique [38]. As mentioned previously, mapping techniques for 3-D applications are

limited to simple geometries with six distinctive surfaces. Complicated domains require manual division of the volume into mappable geometries, which is a time consuming and tiresome practice. Submapping is aimed at automating these unfriendly processes by using a decomposition method.

A decomposition algorithm breaks the surface into mappable regions first by using the technique of automated interval assignments. Automated interval assignment is a procedure that finds the compatible number of mesh intervals on each edge. It uses geometric features recognition routine to classify the edges into six directional groups. A set of equations is derived using these directional groups to determine the interval assignments that will be used by the surface mesh generator.

In the second step, surface mesh generators build structured grids on surfaces of the domain. The locations of surface partitions are based on the automated geometric decomposition. Surface meshes are then propagated to form volume meshes.

The advantage of the submap is its automated nature of handling the routine tasks common to mapped meshing, and the capabilities of producing structured meshes for volumes with more than six logical sides. Its disadvantage is that it suffers from the same limitation as mapped meshing, which requires the volume to be blocky. In addition, the automated interval assignments have difficulties in detecting and identifying hole features [23].

Medial Surface

The medial surface approach is part of the mapping family. It is a direct extension of the medial axis method for quadrilateral meshing [39]. It involves decomposition of the volume by medial surfaces which are the surfaces generated from the midpoint of a maximal sphere as it is rolled through the volume. It generates mappable regions, which can be filled with hexahedra. The medial surface technique is not reliable for all geometries and presents robustness difficulties [23].

3.2.1.2 Sweeping and Mesh Extrusion

Sweeping and mesh extrusion [40] belong to the same class of structured hexahedral mesh generation, which is referred to as 2.5-D geometry mesh generation method. As its name depicts, this method requires the volume to be 2.5-D. 2.5-D geometries are special class of CAD objects that have similar cross-sections between the source and target faces, and a path called the linking surface connects both faces to form a 3-D object. Many CAD features; e.g., cylinders, elbows, and pipes, are in this class of geometry.

During the first stage, the source and target surfaces are pre-meshed using 2-D mesh generation methods. The sweeping algorithm will make a quick comparison between the meshes to ensure their compatibilities. Once their compatibilities are established the source face sweeps across the geometry along the linking surface until it reaches the target face, generating hexahedral elements at the intervals prescribed by the user on the linking surface as shown in Figure 3-2. The difference between sweeping and mesh

extrusion is that mesh extrusion is unidirectional only.

These methods are reliable and extremely efficient, but are only applicable to a specific class of geometry. Lai et al. [27] used the basic of sweeping and extrusion methods and enhanced them by allowing the generation of fully hexahedral finite element meshes from multiple source surfaces to multiple target surfaces. Lai's method is known as Multisweep. These topics are among the most active research in FEA mesh generation.

Figure 3-2 Sweeping technique [23]

3.2.1.3 Grid-based

The grid-based method was introduced by Schneiders to generate fully hexahedral meshes for an arbitrary volume [41]. The program proceeds through two different stages: interior and exterior mesh generation. The initial step is to produce a straightforward fully structured hexahedral element mesh at the interior of the volume, for which the user controls the element size. The goal of this stage is to generate a highly structured hexahedral grid at the interior that extends to half of the user-specified element length of the boundary. The second stage is a complicated process that

connects the node at the outer layer of the interior volume mesh to the proper node of the exterior surface mesh. The final product is a topologically correct fully hexahedral meshed volume.

The shortcoming of the method is the dependency on the complicated relationship between the surface mesh and the outer layer of the interior volume mesh. For instance, an incompatible surface mesh will result in mesh generation failure. In addition to the disadvantage of incompatible meshes, this method is highly sensitive to the orientation of the interior grid.

3.2.1.4 Plastering

Plastering is the 3-D version of the 2-D paving algorithm, which was initially introduced by Blacker [42]. In the 3-D version, seed quadrilateral meshes are generated at the boundaries; individual quadrilaterals are then projected towards the centre of the volume to form hexahedral elements. Figure 3-3 is an illustration of the plastering algorithm. The major drawback of plastering is that complex interior voids may result, and in certain cases filling the geometry completely with hexahedral elements is not possible. As the algorithm advances, intersecting faces have to be deleted. The algorithm determines the connection of pre-existing nodes, joins faces and/or relocates elements when necessary. The program performs well for blocky structures where the surface mesh will form a valid boundary for an interior hexahedral mesh, which does not contain any irregular nodes. The plastering algorithm has not yet been proven to be reliable in all circumstances [2].

Figure 3-3 Plastering meshing algorithm [23]

3.2.1.5 Whisker Weaving

Whisker weaving is an unconventional approach to hexahedral generation that was first presented by Tautges and Blacker[43]. The first stage of whisker weaving is common to many direct generation methods, which is the generation of surface quadrilateral elements. The second stage for typical hexahedral generation methods is to construct the hexahedra, and subsequently resolve the connectivity problems as they arise. In whisker weaving, the connectivity between elements is established first using the Spatial Twist Continuum (STC) scheme. Hexahedra are generated at a later stage.

STC is a geometric scheme that reduces the dimension of 3-D geometries into planes, edges and points. A simple illustration of STC representation superimposed on to a rectangular block is shown in Figure 3-4.

The information is captured through a series of interdependent whisker sheets. Whisker sheets provide important structural information on connectivity and position. The development of whisker sheets is based on local search of neighbouring faces and edges. For instance, an indication for the formation of a hexahedron is three distinct

neighbouring faces sharing three pair-wise edges. The whisker sheets are updated by the formation of the new element. The next sets of candidates are searched locally. This process continues until the whisker sheets are completely filled. Once the sheets are populated with hexahedra connectivity and location data, the hexahedra are constructed in physical space as prescribed on the sheets.

The process is efficient since all searches are performed locally and not globally. The quality of the mesh and mesh size for this approach is directly related to the coarseness of the surface mesh. A finer surface mesh will result in a higher final quality mesh and vice versa. The method is still under development, and the published program is currently restricted to several hundred elements [23]. However, the developers of whisker weaving are optimistic about its future application.

Figure 3-4 STC of a block [23]

3.2.2 Indirect Element Generation Scheme

The indirect method relies on an initial set of triangular or tetrahedral elements in the

domain. These schemes are able to take advantage of local topology information provided by the initial mesh. A background tetrahedral mesh is more effective in achieving a rapid gradation effect over irregular and complicated domains [34] [44].

The advantage of this kind of algorithm is the considerable benefit in avoiding the significant process of intersection calculations and closure resolution inherent to direct methods [14].

Two indirect methods will be examined and discussed in the following sections, they are Owen's H-Morph and a tetrahedral dicing method referred to in this paper as TTH.

3.2.2.1 H – Morph

Owen [44] presents H-Morph, an indirect element generation scheme approach that produces a mesh dominated by hexahedral elements. H-Morph uses the advance front technique [14], which produces hexahedral elements from the outer boundaries and proceeds inward to the interior.

The original tetrahedral mesh is systematically transformed and combined into a topologically appropriate formation of well-shaped hexahedra. The initial front consists of a set of prescribed quadrilateral surface grids, which result from combining the surface triangular elements. Individual fronts are processed by recovering each of the six quadrilateral faces of a hexahedron from the tetrahedral mesh.

The method results in a reasonable quality collection of hexahedral elements, and a mesh size that is comparable to the original one. The rigid nature of hexahedral

element's geometry in some cases does not allow proper hexahedra to be formed at the interior of the domain. The result is hexahedral elements covering the exterior portion of the domain, while the interior domain remains populated with tetrahedral elements.

3.2.2.2 Splitting (TTH)

Splitting is a straightforward extension of the 2-D splitting method [17]. In 2-D, every triangle can be split into three quadrilaterals by connecting the points on the edges of the triangle with an interior point. In 3-D cases, the middle edges, the centroid of each surface and the interior point, are the required nodes to divide the tetrahedral elements into four hexahedra. Each surface of the tetrahedron is divided as in the 2-D model and the central interior point is first assumed to be at the centroid as shown in Figure 3-5.

The TTH refers to splitting a tetrahedra into four hexahedra. It will be used to describe the tetrahedra-to-hexahedra conversion module of the software in this thesis. The simplicity of this method and its suitability for complex meshes were key factors for the selection of this approach, for the creation of an algorithm capable of developing a fully hexahedral mesh. A more detailed description of the processes and methodologies involved will be provided in the following chapter.

Although the splitting method has numerous advantages it does have some drawbacks, and they have to be tackled in order to obtain a desirable mesh. The initial tetrahedral elements are transformed quickly to hexahedral elements, but the excessive subdivision and refinement of the mesh may lead to poor quality hexahedra. The traditional thinking that meshes, which are a product of a tetrahedral division are of insufficient

quality for FEA, has been changing due to recent studies that have shown otherwise for selected analysis types [30].

Figure 3-5 Tetrahedron to hexahedron conversion [23]

3.3 Mesh Quality Assessments

Mesh quality enhancements are often required to improve the quality of the mesh when it has been produced by indirect schemes. It is known that mesh quality affects the accuracy of an analysis, and clearly the element distortion should be minimized [45]. A hexahedral element is said to be distorted when the shape of the element in the physical (X, Y, Z) domain deviates from the one in computational (ξ, η, ζ) space [45].

An ideal hexahedron such as the one mentioned in computational space is described as a unit cube, where all edges are of equal length and perpendicular to each other. It is certainly not realistic to expect all conformal FEM mesh of arbitrary geometry be composed of only unit cubes. The degree of distortion in hexahedral meshes is often measured by several basic quantities such as aspect ratios and skew angles. Aspect

ratios describe the stretching of an element, and skew angles describe the degree of twisting in an element. The Oddy metric [45], condition number [46], and the determinant of Jacobian [47] are examples of some commonly used quantifiers for mesh quality assessment. Table 3-2 is a brief list of some of the functions for assessing mesh quality, included are their units, acceptable and possible ranges.

Table 3-2 List of mesh assessment functions and relevant information [45] [47]

<i>Function Name</i>	<i>Units</i>	<i>Full Range</i>	<i>Acceptable Range</i>
Aspect Ratio	---	1 to ∞	1 to 4
Skew	Degrees	0° to 180°	45° < range < 135°
det [Jacobian]	Volume	- ∞ to + ∞	(+) Positive
Oddy Metric	D*	0 to + ∞	≤ 10

* D, Distortion metric is a function of the terms of the Jacobian to the fourth power.

In this project the aspect ratio and skew angles were selected to quantify the quality of the element. They are evaluated in this program at the centroid of each element by decomposing the Jacobian matrix. These fundamental quantities describe all the first – order mesh qualities (length, areas and angles) of interest [47]. A mesh with all-positive determinants, is usually the minimum acceptable quality mesh [47]. The aspect ratio in this project tracks the ratio between the physical space distances from the centroid to the element boundaries in three local space directions (ξ , η , ζ). Similarly, the skew angles are the angles between the three axes of the local coordinates in physical space evaluated at the centroid.

Mesh enhancements such as smoothing improve the accuracy and stability of the

numerical solution, and reduce the number of elements required to capture the underlying physical phenomenon [48]. The effectiveness of mesh smoothing techniques relies on a mesh quality assessment, where both detecting and locating a problem are important.

3.4 Mesh Smoothing

Mesh smoothing is used to improve the quality of meshes that are distorted beyond the allowable limits stated in the previous section. In general, smoothing techniques involve an iterative change of the location of grid points in the mesh without altering the overall mesh topology. Smoothing techniques can generally be classified as averaging methods, optimization-based methods, physically based methods and midnode placement [49].

Averaging methods are the most common techniques. They reposition nodes based on a weighted average of the geometric characteristics of the surrounding nodes and elements. Laplacian smoothing is the most common averaging method. Optimization-based methods measure the quality of the elements surrounding a node and attempt to optimize by computing the local gradient of the element quality with respect to the node location. Physically based methods [50], reposition nodes derived from a simulated physically based attraction or repulsion force. Both optimization and physically based methods are commonly combined with the Laplacian technique to reduce the excessive computational time required. On the other hand, the midnode placement method repositions not the corner nodes but rather the midnodes on quadratic elements to

improve the element quality. This technique is not relevant for this project since the elements used here are linear.

Brief description of the equipotential, opti-smoothing, weighted area, Laplacian methods and isoparametric method, an alternative to the Laplacian method, are presented in the following sections. Emphasis is placed on the Laplacian and isoparametric methods as these techniques were selected for implementation in the thesis.

3.4.1 Opti-Smoothing

Opti-smoothing is a global smoothing method. The goal is to minimize the global average distortion metric using optimization procedures. The global average distortion metric is the summation of the distortion metric [45] of all elements divided by the number of elements. The average distortion metric characterizes the mesh quality, and it can be minimized through various optimization techniques. The optimization technique given as an example in Hyun and Lindgren [49] is a conjugate gradient method, which is able to handle large number of variables in the optimization.

The opti-smoothing method can be extended to include a global maximum distortion metric. The combination of both local and global parameters gives the condition of the worst quality element plus the overall quality of the mesh. This method occasionally causes a negative Jacobian matrix, and optimization has to be restarted from a previous known good mesh. It is also a time consuming method [51] due to its global smoothing approach.

3.4.2 Weighted Area

In this method a node is pulled towards the centre of each of its surrounding elements. Each of its neighbouring elements contributes the direction along which the node translates. The magnitude of the movement is proportional to the areas. The method is quick and can be implemented into any routine with little difficulty. The quality of the smoothing is not always reliable as demonstrated in Hyun and Lindgren [49].

3.4.3 Laplacian

Laplacian smoothing is perhaps the most common smoothing technique used today. It is simple to incorporate into any program and is widely applicable. There are two variations of this method, but the overall idea is to place the node in the relative centre among its neighbours. The 'near' variation of Laplacian smoothing is to average the location of adjacent nodes, for example nodes 1 to 4, around node 0 in Figure 3-6.

Figure 3-6 is a typical 2-D mesh with node 0 as the intended node to be moved by smoothing. The 'all' variance of Laplacian method is the average of all surrounding nodes, for example, nodes 1 through 8 in Figure 3-6.

The advantages of the Laplacian method have been highlighted in the previous paragraph, but there are some difficulties with this method. For example, the length-weighted Laplacian formulation has difficulty with highly concave regions and requires iterations to find the position of the node points [23]. Secondly, it suffers from the same drawback as weighted area method, which is its unreliability stemming from the

lack of consideration of element distortion.

Figure 3-6 Laplacian method [49]

3.4.4 Isoparametric

Herrmann [52] presents the isoparametric method as an alternative to the general Laplacian scheme for 2-D bodies. It is shown as an improvement over the Laplacian scheme, which he considers too insensitive to capture the information contained in the grid spacing and boundary curvature defined by specified boundary node locations.

In the isoparametric technique the position of the node to smooth is set to create elements to be as close to parallelograms as possible. Consider the element in Figure 3-7 [26], x is the point to be smoothed. Its new position x' allows the element to become a parallelogram. To obtain x' value, the following equation should be applied:

$$x' - c = a - b$$

, or

$$x' = a + c - b$$

[3-1]

The new position is given by the average contribution of all elements neighbouring x .

Therefore, the new position is:

$$x' = \frac{1}{NE} \sum_{r=1}^n (a_r + c_r - b_r) \quad [3-2]$$

Where NE is the number of the neighbour elements, a_r , c_r are the position of the connected adjacent nodes in the r -th neighbouring element, and b_r is the position of the node opposite to the working node.

The method is effective in cases where the meshes are structured, because it maintains the general shape of the element. Hyun and Lindgren [49] showed a comparison of Laplacian and isoparametric methods applied to the 2-D case of a concave mesh as shown in Figure 3-8. The isoparametric method Figure 3-8(c) outperforms the Laplacian method Figure 3-8(b). However, in the case of unstructured meshes, where the geometries are more arbitrarily formed, the isoparametric is much less useful. Its efficiency dropped drastically as demonstrated in Hyun and Lindgren [49].

To address this issue, Herrmann [52] proposed a grid generation parameter w , which varies from 0 to 1. A value of $w = 1$ represents a fully isoparametric generation scheme, whereas a value of $w = 0$ yields a fully Laplacian scheme. Intermediate values of w produce mixtures of the two approaches. Equation [3-2] becomes:

$$x' = \frac{1}{n(2-w)} \sum_{r=1}^n (a_r + c_r - wb_r) \quad [3-3]$$

In general, the isoparametric method has been seen to be better for structured grids, while the Laplacian methods are better with unstructured grids. The new method is designed to cover the complete range from unstructured to structured grids by optimizing the w parameter. Herrmann [52] found that the most significant effect for most meshes is when w is in the range of 0.7 to 1.0.

Figure 3-7 Isoparametric smoothing method

Figure 3-8 Comparison between Laplacian and isoparametric smoothing methods

- (a) Original mesh (b) Laplacian smoothing showing difficulty with curvatures
(c) Isoparametric smoothing improved performance [49]

3.4.5 Equipotential

The equipotential approach is similar to the Laplacian scheme and the weighted area method, in that they all smooth the mesh by adjusting nodal location. In the equipotential method, the objective is to adjust the node based on equalizing the volume of the surrounding elements. The attractive attribute of this method is its ability to “pull in” badly shaped elements [23]. It is also a computationally expensive scheme, because it performs iterations of every single element in the entire domain.

3.4.6 Untangling and Geometric-Based Smoothing

The untangling method assumes that the mesh has valid connectivity but the node point positions are such that some of the elements are inverted and would have negative Jacobians. Meshes can be untangled using optimization methods, which are typically used since traditional smoothing methods don’t attack the problem directly and can be time consuming [53]. In this thesis, a geometry-based technique similar to untangling will be used for distorted elements. In particular, the aspect ratios of the elements and the skew angles are adjusted by moving nodal positions as appropriate. While the goal of mesh untangling is to eliminate inverted elements from a given mesh, the goal of the geometry-based technique is to improve the mesh quality since the initial mesh from a CAD/CAM or specialty pre-processor will not have any inverted elements, but may have distorted elements. Strategic combinations of the approach with isoparametric smoothing are also included to reduce the computational cost, without affecting the efficiency of the method.

4 Element Splitting and Quality Assessment

This chapter contains detailed information on the procedure used for the creation of a hexahedral element mesh. As was explained in the previous chapters, an existing tetrahedral element mesh is converted to a hexahedra mesh through element division, which is a straightforward extension of the 2-D splitting method [17].

The tetrahedra-to-hexahedra (TTH) software is programmed in C language, and is composed of four main modules: data-gathering, splitting, quality assessment and smoothing. The data-gathering module extracts mesh information from a variety of CAD pre-processor input files, such as Pro/ENGINEER and I-DEAS. The splitting module transforms each tetrahedral element into four hexahedra, which is the main focus of this project. Quality assessment evaluates the quality of initial tetrahedra and the resulting hexahedra. The smoothing module uses two different algorithms to improve the mesh quality. This chapter introduces the principles and approaches taken in the first three modules. The last module, smoothing, is discussed in chapter 5.

4.1 Programming Language

The goal for TTH is to be efficient and practical. C is a programming tool that can

deliver these qualities. In order for the program to be considered a practical extension of FEM mesh generation, it must be able to handle reasonably large mesh sizes. This does not pose as a significant problem to C, because it has the capability to operate on large multi-dimensional arrays. Arrays are also an effective means for computational analysis, which is discussed in data structures section. In addition, C is equipped with dynamic memory allocation. It allows TTH to allocate memory, based on demand. Dynamic memory allocation improves the program efficiency by making more memory available for other functions or programs. Static allocation is less efficient because it reserves specified memory space regardless of the demand.

One incentive for using C is the pre-existing library of functions that make use of the abstract data type referred to as a K-D Tree. K-D tree is a data structure used by McDill [4] for associative searching. It is extremely valuable in the splitting process. The powerful search and comparison algorithm reduced computational cost considerably as opposed to other techniques [4]. The program development greatly benefited from the versatile debugging environment for the C language in Windows NT. It substantially reduced the development cost by indicating precisely the faulty areas and the possible cause of the problems. Several other advantages such as cross-platform compilation are some of the considerations during the programming language selection process.

Figure 4-1 shows a block diagram of the TTH process, which begins with the data-gathering module, followed by conversion of the data into a specified format from the acquired FEA input file style, classification of the data, splitting of the tetrahedra into hexahedra, assessment of the elements, assessment of the position of the parent

tetrahedron's centroid and movement of it to produce better hexahedra. Finally a smoothing module which adjusts both the aspect ratio and skew of the hexahedra is applied. The results are output to the required format to be used by the specialty FEA codes.

Figure 4-1 Tetrahedra-to-hexahedra (TTH) process

4.2 Data-Gathering

The input and output formats of the program are constructed with the goal for ease of adaptability and with interchangeability in mind. It is anticipated that many users of specialty FEA codes such as, SIMPLE [7], ABAQUS, DYNA, TIAMAT [6] and others will find this software useful. The writer has chosen the TIAMAT format as the primary interface due to its straightforward structure, and in conjunction with the original intention to build a module for the seamless integrated hexahedral generation in TIAMAT. Nevertheless, adoption of other formats has proven to be an easy task. Several routines have been successfully written to extract mesh data from popular commercial programs that have limited hexahedral generation capabilities. The two input formats that TTH supports at this time are I-DEAS universal and the ANSYS standard input format produced by Pro/ENGINEER. These are converted so that the output of TTH is, in fact, in the style required for the geometry and topology of the FEA input file.

4.3 Splitting

All popular FEM integrated CAD/CAM software and specialized FEM codes have some form of meshing capabilities, but mesh generation in the 3-D domain is commonly limited to tetrahedral elements. TTH is designed to extend their capabilities by transforming a fully tetrahedral mesh to a fully hexahedral mesh. The splitting module systematically transforms each linear tetrahedral element into four hexahedra. The data-gathering module extracts the linear tetrahedral mesh information, and the data

are submitted in a specified format to the splitting module.

The approach used by the splitting module is to place an additional node on each midedge and an interior node (midbody) at the centre of each tetrahedron. Four other nodes are also inserted on each midface to facilitate the splitting technique. Figure 4-2 is an example of a tetrahedral element with the additional nodes inserted. The next step is to link the midface nodes with their three adjacent midedge and midbody nodes. In the final stage the fully connected tetrahedron is split into its four hexahedra. Figure 4-3 shows typical transformations from a tetrahedron to hexahedra using TTH.

Figure 4-2 Linear tetrahedron with additional nodes
Vertex nodes (1-4), Midedge nodes (5-10), Midface nodes (11-14) and Midbody node (15)

Figure 4-3 Typical splitting (Showing 4 hexahedra created from a tetrahedron)

The runtime of the algorithm depends on various factors such as computational power of the computer and data structure used. The former factor depends solely on the type of computer system available to the user, and was not considered during the code development. Data structures on the other hand, have a direct influence on the program performance. Therefore, using an efficient data structure was a priority. The considerations and developments of such structures are discussed in the next section.

4.3.1 Data Structures

The principal data structure used for the geometry is the array `ElNodes [1, MaxEl] [1, 15]`. It is dimensioned to 15 nodes and `MaxEl`, the number of tetrahedral elements in the mesh. `ElNodes` keeps the node list for the tetrahedra in terms of global node numbers. This data structure is important for the storage of the initial data of the tetrahedral mesh, and to store the subsequent generation of nodes at the midedges, midfaces and the midbody. Positions 1 to 4 hold vertex nodes provided by the package with the original tetrahedral mesh such as I-DEAS or Pro/ENGINEER, while positions 5 to 10 hold the midedge nodes. Positions 11 to 14 hold the midsurface nodes, and the interior node is in position 15. The midbody node position is directly related to the quality of the resulting hexahedral elements. The technique for adjusting its position is described in the following section.

Each node has three coordinates. Three vectors `xcoord[]`, `ycoord[]` and `zcoord []` hold the X, Y, and Z coordinates respectively for all nodes in global space and are dimensioned to the maximum number of nodes.

Once the initial data in the previous sections are stored, TTH begins the process of splitting tetrahedral edges to create midedge nodes. Nodal coordinates in physical space for midedge nodes, midsurfaces and tetrahedral global centroid are determined using the tetrahedral basis functions in the computational space coordinates (ξ, η, ζ) .

The interpolation functions in this project are expressed as:

$$\begin{aligned} N_1(\xi, \eta, \zeta) &= 1 - \xi - \eta - \zeta \\ N_2(\xi, \eta, \zeta) &= \xi \\ N_3(\xi, \eta, \zeta) &= \eta \\ N_4(\xi, \eta, \zeta) &= \zeta \end{aligned} \quad [4-1]$$

Figure 4-4 shows the computational (ξ, η, ζ) coordinates of nodes 1, 2, 3 and 4.

Figure 4-4 Computational coordinates of a tetrahedron [54]

One concern during the development of this phase was to address the problem of repeated nodes, which occur as nodes added to one element must also be inserted in its neighbours. Two different approaches were considered. In the first approach the relations between neighbouring linear tetrahedra prior to splitting are defined. The

information is then used to insert new node information to the intended and adjacent tetrahedra. The complexity of the mesh connectivity grows as the size of the mesh increases due to the unstructured nature of free meshing. The overlapping of nodes on edges made this approach difficult to implement.

The second approach is simpler and was implemented into the program. Each tetrahedron is processed on an individual basis rather than globally as in the previous approach. The newly created nodes for each tetrahedron are examined, unique nodes are inserted into the node list and duplicated nodes are replaced with the correct node number. This approach demands an efficient node comparison routine in order to be successful. Fortunately, K-D trees [55] are data structures used to facilitate effective data retrieval. They store information in a binary tree format to be retrieved by associative searches. Data can be inserted and extracted from the K-D tree [55].

The K-D tree functions were extracted from existing codes [4] and were used in a 'black box' mode. The use of those functions has been a key factor in developing well formed topological meshes; i.e., meshes without either extra or duplicated nodes. Once the process is completed the initial tetrahedral mesh is held in a data structure (K-D tree), which now treats the tetrahedra as 15-node elements.

A well-organized data structure improves the efficiency of the program, but other measures are also taken in TTH to optimize the runtime. One such measure is to minimize mesh optimization by improving the inherent quality of hexahedra. The routine in the next section is designed specifically for this purpose.

4.4 Tetrahedral Centroid

The original arrangement in the splitting module is to place the nodes at the midedges, midfaces, and midbody of the tetrahedron. These locations do not necessarily form the best hexahedra. On an individual basis, the new nodes in a tetrahedron can be adjusted to obtain the best possible hexahedra. Globally this is a complicated scenario, where midface and midedge nodes are shared between adjacent tetrahedra. The optimal location for one tetrahedron is often not suitable for the other node-sharing neighbours. The best locations for shared nodes have to be determined iteratively amongst the elements. The problem dealing with shared nodes is time consuming and was avoided.

The midbody node however, is not shared and is unique for each tetrahedron. Its position can be adjusted without affecting neighbouring tetrahedra. The result is improved quality in the resulting hexahedra with minimal increase in computational cost. The relocation of the midbody node is the fundamental idea in TetCentroid module.

4.5 Tetrahedral Quality Assessment

Although I-DEAS and Pro/ENGINEER have advanced meshing algorithms that produce good quality meshes, it is often not practical nor possible to obtain an unstructured free meshed geometry populated with only high quality tetrahedra. It was then decided to adjust the centroid position in lower quality tetrahedra to produce better shaped hexahedra. A tetrahedral quality assessment function was needed to identify distorted elements.

Parthasarathy [56] recommended using the aspect ratio (γ) to assess the quality of tetrahedra. The aspect ratio can be used with a wide variety of tetrahedra and it is also computationally inexpensive to determine.

The equations for determining aspect ratio (γ) are:

$$\gamma = \frac{S_{rms}^3}{V} \quad [4-2]$$

where

$$S_{rms} = \sqrt{\frac{1}{6} \sum_{i=1}^6 S_i^2} \quad \text{for } i = 1 \text{ to } 6 \text{ edges} \quad [4-3]$$

and

$$V = \frac{1}{6} * \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} \quad [4-4]$$

Where x, y and z refer to the physical coordinates of the four vertex nodes.

The normalized aspect ratio γ_N is then calculated based on the aspect ratio of a perfect equilateral tetrahedron [56]; i.e., 8.47967 as shown in equation 4-5.

$$\gamma_N = \frac{\gamma}{8.47967} \quad [4-5]$$

γ_N can then be used to compare the quality of a variety of tetrahedra. Limits placed on the shifting of the tetrahedral centroid are discussed in chapter 6.

4.6 Hexahedral Quality Assessment

Various methods have been designed to measure the quality of hexahedra. As indicated previously, aspect ratios and skew angles have been selected for the purposes of assessing the hexahedra quality. Clearly an ideal hexahedron has three skew angles of 90° and the aspect ratios of one. Figure 4-5 shows an 8-node isoparametric element in local space.

Figure 4-5 8-Node, isoparametric volume element [57]

The eight isoparametric equations for a hexahedron are¹⁰:

$$\begin{aligned}
 N_1 &= \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta) & N_5 &= \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta) \\
 N_2 &= \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta) & N_6 &= \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta) \\
 N_3 &= \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta) & N_7 &= \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta) \\
 N_4 &= \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta) & N_8 &= \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)
 \end{aligned}$$

[4-6]

¹⁰ This is the node numbering scheme used in TIAMAT.

The first step in calculating the quality of the hexahedra is to evaluate the set of shape functions for the hexahedron. The Jacobian matrix is formed from the derivative of the above equations and evaluated, at centroid, in this case¹¹.

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} y_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} z_i \\ \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} y_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} z_i \\ \sum_{i=1}^8 \frac{\partial N_i}{\partial \zeta} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \zeta} y_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \zeta} z_i \end{bmatrix} \quad [4-1]$$

The Jacobian, which relates physical and local space, can be decomposed to provide the three principal aspect ratios $\left(\frac{\xi}{\eta}, \frac{\eta}{\zeta}, \frac{\zeta}{\xi} \right)$ and the three principal skew angles $(\theta_{\xi\eta}, \theta_{\eta\zeta}, \theta_{\zeta\xi})$. Information regarding the distance and direction for the 3 vectors in Figure 4-6 is extracted from the Jacobian matrix. Aspect ratio and skew angles can be obtained by manipulating the three vectors as described in Kerlick and Klopfer [58] and extracted from McDill [4].

¹¹ The definition of the Jacobian is important. In some references the form of the Jacobian used is the transpose; i.e., J^T of the form seen in equation [4-1]. Either form may be used provided the approach is consistent.

Figure 4-6 Computational coordinate axis in global space

5 Smoothing

It is commonly known that the use of well-shaped hexahedra is prudent for an accurate analysis. The best quality hexahedral element has a cubic-like geometry matching the computational geometry. However, in order for a hexahedral mesh to model complicated geometries accurately, element deformation is often necessary. Nevertheless highly deformed elements are undesirable to an analysis.

An in-depth discussion regarding the classification of the element quality is presented in this chapter. Smoothing is a standard feature in mesh generation programs. Distorted elements are repaired using smoothing routines, although sometimes at the expense of their better-shaped neighbours. Various examples of smoothing techniques have been presented in the Literature Review (Chapter 3). Two distinct types of smoothing routines are implemented in the TTH software. These techniques are the geometry-based technique and the isoparametric technique. Each routine focuses on a specific aspect of smoothing.

5.1 Hexahedron Quality Classifications

A hexahedron has a wide range of potential geometries from long and narrow to right-angled and skewed. These shapes are all geometrically valid, but few are well suited to FEM analysis. The commonly accepted metric values of a good quality hexahedron are

aspect ratios that do not exceed 3.0 and skew angles that do not deviate more than 45° from the right angle [45]. These figures serve as a guideline to determine the validity of hexahedra in the TTH algorithm. The program uses a more conservative aspect ratio of 2.5 as the smoothing criteria, because smoothing only improves element quality but does not necessarily make the element acceptable. Thus a better overall mesh is achieved with a slight increase in computational cost.

The measuring standards for aspect ratios and skew angles were stated earlier, but are reiterated here for clarity. Aspect ratios refer to the three length-to-width ratios of a hexahedron. The skew angles refer to the angles between the same three axes used in the calculation of the aspect ratios. Based on the above limits and quality-measuring units, an acceptable hexahedron is defined as one which has a maximum aspect ratio of less than 2.5 and skew angles between 45° and 135° .

5.2 Geometry-Based Technique

It was realized early in the development of the TTH that not all free meshed tetrahedra are of good quality. Their hexahedral children are often badly formed and clustered together. In addition, the effectiveness of conventional smoothing techniques such as the Laplacian and isoparametric methods depends largely on the surrounding elements. The combination of these factors result in poor performance of these smoothing methods. The selected alternative is the geometry-based technique. In the geometry-based approach the elements are individually modified so that the hexahedron is adjusted without accounting for its surroundings. The result is a dramatic improvement

to the element after the operations. Two procedures are included in TTH software. The first (AR Fix) is designed to fix the aspect ratios, and the second (Skew Rotate) improves the skew angles.

5.2.1 AR Fix

AR Fix was designed to improve elements with large aspect ratios. It corrects the aspect ratio by reducing the distance between the opposite faces that are furthest apart, and extends the length between the faces that are closest together. During the development of the routine, it was found that the algorithm's efficiency strongly depends on the direction in which the faces are moved. The optimal direction reduces the disturbances on the other element qualities. In Figure 5-1 the scheme of the AR Fix procedure is presented.

The strategy used in AR Fix is to move the faces in parallel with the computational axis as needed. There are two immediate concerns: the first deals with the selection of the proper axis for the face translation. The second addresses the rotation of the computational axis in the physical space after the centroid relocation. The relationship between the axes in the physical space and the computational space is not "one-to-one"; i.e., they do not necessarily coincide with each other. In addition, the computational axis in the physical space may rotate if the centroid of the hexahedron is repositioned. For example, if a face of a hexahedron is translated linearly in a specific direction in the computational space, it may be travelling in a curvilinear manner in the physical space. The exact shape of the path is determined iteratively in TTH. The

routine allows a face to translate in a specific direction at small intervals. A new direction is calculated at the termination of each interval. These processes are repeated continuously until the 20th iteration¹² or until the hexahedron quality is acceptable, whichever condition is reached first.

Figure 5-1 AR Fix procedure

¹² Experience has shown that 20 iterations is suitable

The selection of the direction of translation is straightforward. The Jacobian, evaluated at the centroid, is decomposed. The routine first determines the largest out-of-range aspect ratio. The two lengths in physical space associated with that aspect ratio are measured. The opposite faces connected to the longest length are translated, parallel to that length, towards each other. The next step involves the faces connected to the shortest length, and they are translated in a similar fashion but away from each other. Figure 5-2 (a) shows an example of a hexahedron with a high aspect ratio between the global (Y, X) , local (ξ, η) lengths. The physical dimensions are shown. The local axes are included for clarity. Different orientations between the global coordinate axes and the computational space are typical. Parts (b) and (c) of Figure 5-2 show the directions applied in TTH to correct for the high aspect ratio existing in this particular hexahedron. Part (b) shows AR Fix reducing the overall length between the global (Y, X) and local (ξ, η) faces. Illustration (c) depicts AR Fix increasing the spacing between the global (Y, Z) and local (ξ, ζ) faces.

Figure 5-2 AR Fix application

(a) Distorted hexahedron with high aspect ratio (global $\frac{Y}{X}$, local $\frac{\xi}{\eta}$). (b) AR Fix performed in the direction of (global Y , local ξ) and (c) AR Fix performed at (global X , local η)

The routine works best for interior elements, where all eight nodes are available for translation in the global space. This method is ineffective for boundary elements, especially for hexahedra with large number of exterior faces. Element faces are checked using a function based on existing code. The hexahedra with exterior surfaces are difficult for smoothing, because their surface nodes are restricted from translation. The exterior faces comprise the mesh topology, and are prohibited from moving in TTH. The AR Fix routine omits such nodes and translates only the interior nodes. This may result in warpage of a face; i.e., non-coplanar face nodes. Limiting the number of iterations in TTH prevents the excessive face warpage.

5.2.2 Skew Rotate

Skew Rotate is the geometry-based smoothing routine for fixing the skew angles. Skew Rotate is similar to AR Fix. In fact, they share the same element classification techniques and translation routines. The only difference is that the element faces are translated in different directions. The Skew Rotate and the AR Fix routines operate independently in TTH, but it is feasible to combine them into a single routine. The reasoning for keeping them separate is to allow the developer easy access for examination and fine-tuning of the routines.

Figure 5-3 shows the Skew Rotate procedure, which is similar to AR Fix. Skew Rotate first finds the unacceptable skew angles in the hexahedron. It then determines if the skew angle is higher or lower than the acceptable limits. The directions of translation for the faces are controlled by both the skew angles and their magnitude. A table

showing the dimensions of the translation of the 8 nodes to adjust the skew angles is included in Appendix 1. The ultimate goal for Skew Rotate is to increase or decrease the specified skew angle. It is designed to accomplish this, without producing face warpage while at the same time maintaining the magnitude of the other skew angles. The approach is to twist the element along a specified plane. If the nodes are translated on a parallel plane, at the desired opposite directions, the face warpage in the element is negligible.

Figure 5-3 Skew Rotate procedure

The skew angles are determined by decomposing the Jacobian at the element centroid. The remaining skew angles are not affected, because the changes occurred on a specific plane. Figure 5-4 (a) is an example of an element in global space with excessively large skew angle between the global (X, Y) and local (ξ , η) faces. Figure 5-4 (b) shows the rotation of the $\eta - \zeta$ faces, while Figure 5-4 (c) demonstrated the rotation of the $\xi - \zeta$ faces. The $\xi - \eta$ faces are left undisturbed in this operation to preserve the remaining of the skew angles. The operations in Figure 5-4 (b) and (c) can individually accomplish the task of correcting the specified skew angle, but it is more efficient to apply them in combinations.

Figure 5-4 Skew Rotate application

(a) Element with excessive skew angle between (global Y-X, local $\xi - \eta$) axes. Rotation of the (b) (global X-Z, local $\eta - \zeta$) faces and (c) (global Y-Z, local $\xi - \zeta$) faces.

5.3 Isoparametric Technique

The isoparametric technique and the Laplacian method utilize the principle of average node position. The former has the additional benefit of accounting for the boundary curvature and node point spacing for locating interior nodes [52]. TTH uses a hybrid form of these techniques to smooth the nodes; it is simply referred to as the isoparametric method in this project. It is a direct extension of the initial 2-D isoparametric equation presented by Herrmann [52]. A full derivation of the 3-D method is presented in Appendix 2.

The following equation for the 3-D isoparametric method is the derivation of Herrmann's original equation to accommodate 3-D elements.

$$x' = \frac{1}{3 * NE} * \left(\sum_1^{NE} w * (x_A + x_B + x_C) + 3 * (1 - w)x_D \right) \quad [5-1]$$

Where the nodes x_A , x_B and x_C are adjacent to the intended node for smoothing (x'). On the other hand, x_D represents the element node that is in the opposite corner to x' . A simple illustration of a single brick with the relevant nodes labelled, is shown in Figure 5-5. The term w in [5-1] represents the weight function, a value of $w=0$ is equivalent to the Laplacian method while $w=1.0$ is equivalent to the isoparametric. Intermediate values are a combination of both smoothing techniques. The chosen value for w depends on the geometry and characteristics of the mesh [52]. For example, an intermediate weight value can be more effective for node smoothing than using a pure Laplacian or isoparametric technique. Although Herrmann did not develop a 3-D

scheme, he recommended a value of $w \approx 0.95$ as well as the use of a direct solution scheme instead of iteration, in order to reduce the computational costs. In this case experience has shown $w = 0.7$, provides a better balance and overall performance than other values.

Figure 5-5 Connecting nodes for isoparametric calculations
i is the node to be moved; Adjacent nodes: A, B and C; D is the opposite node.

The isoparametric routine is used to complement the geometry-based techniques. Unlike the geometry-based method that smoothes in the mesh an element at a time, the isoparametric method smoothes the mesh on a node-by-node basis. The result of the isoparametric routine depends on its surrounding elements, and on their surrounding nodes.

The process begins essentially in the same manner as the geometry-based routines with one exception. The connectivity information between the nodes and the elements has to be constructed prior to the smoothing process. Once the information is completed, the routine seeks out elements with a skew angle which exceed the previously stated limits. The isoparametric smoothing is performed on a pair of opposite nodes in the poorly formed element. The first choice for node selection is the first and the seventh local node pair of the element. If either one or both of the nodes are exterior nodes, the program will select the next opposite node pair in numerical order. If complete pairs are not available, the routine is designed to use the first available node in numerical order. The intention for using an opposite node set is that once the procedure is performed, all faces of the element are adjusted and the quality of the element is enhanced.

5.3.1 Element Verification

If the mesh optimization routines are allowed to iterate continuously, it might alter the element to the point of creating inverted elements. For example, an initially poor element, once smoothed, may have nodes inverted; i.e., a negative volume. Such scenarios are prevented in TTH by the implementation of the element verification routine. The element verification routine follows each smoothing procedure, and is used prior to the insertion of any new node coordinates.

The detection of invalid elements is a simple task if the hexahedra are ideally shaped and oriented along the global coordinates. The reality is that the elements are not often

ideally shaped and their local axis orientations seldom coincide with the global coordinates. The unstructured nature of the mesh also causes difficulties. A single node can be shared amongst many elements, TTH allows for 20 elements to connect at a single node¹³. These pose problems for any approach that operates solely in the global space. The TTH routine relies mainly on local coordinates. The routine determines hexahedral shape functions for all the elements that shared the nodes and the equations are based on the original node positions. An existing modifying module that converts global coordinates to local coordinates is incorporated into the routine. It is well known that a hexahedron in local space has coordinate limits of (-1, -1, -1) and (1, 1, 1) in its three coordinate directions. If the local coordinate of an arbitrary point exceeds any of these limits, it must reside outside of the hexahedron. The same principle applies in the element verification routine, where the computational space position of the intended node is obtained for each of the neighbouring elements. If the movement is valid, the node will reside in at least one of the neighbouring elements. If the node position is situated outside of all the neighbouring elements, the element is said to be invalid. The invalid node location is discarded and the original node location is reinstated.

The local coordinate with respect to the original hexahedron is tested with FINDRST function. It iteratively determines the local coordinate (ξ, η, ζ) position from a given global coordinate in (X, Y, Z) space. It is based on the chain rule and the Jacobian as follows:

¹³ Preliminary testing showed that a limit of 20 was appropriate.

$$\begin{bmatrix} d\xi \\ d\eta \\ d\zeta \end{bmatrix} = [J^{-1}]^T * \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad [5-2]$$

The shape functions that form the Jacobian have to include the original node location, because the hexahedra formed by the original node location are used as the baseline. The new node location is considered valid when one of the local coordinates sets is within the limits of a hexahedron in the computational space (-1, -1, -1) to (1, 1, 1).

6 Testing and Results

Tetrahedra-to-hexahedra (TTH) is a practical utility for FEA, but to this point, no information regarding its performance has been presented. This chapter addresses these issues by presenting and discussing various aspects of TTH testing and operations. Attention is paid to the pre-splitting treatment of the tetrahedra and post-splitting treatment of the TTH generated hexahedra. The final section of this chapter is devoted to the evaluation of TTH capabilities under different operating systems (OS).

6.1 Test Cases

Throughout the early development of TTH many simple cases; i.e., less than perhaps 20 tetrahedral elements, were used to refine the software. Later four test cases incorporating a variety of geometric features and element density, were used to more rigorously test the software. These cases are a wedge, cylinder, throw and stationary platen.

The wedge, shown in Figure 6-1 has flat sides. The 'cylinder' shown in Figure 6-2 has curved sides. Figure 6-3 shows the throw [59] which has curved protrusions and flat sections. Finally, in Figure 6-4 an actual part used by the plastics manufacturing industry, a stationary platen [19] is tested. This model has several complex internal features. In all cases, various free meshed tetrahedral grids of different element densities are tested.

44 Tetrahedra
Figure 6-1 Wedge

4050 Tetrahedra
Figure 6-2 Cylinder

102 Tetrahedra

1293 Tetrahedra

Figure 6-3 Throw

Table 6-1 shows the mesh densities of the chosen test cases.

Table 6-1 Test cases mesh densities

Part Name		Initial (Tetrahedra)		Final (Hexahedra)	
		No. of Elements	No. of Nodes	No. of Elements	No. of Nodes
Wedge		44	25	176	267
Cylinder		4,050	970	16,200	19,336
Throw	(Coarse)	102	48	408	595
	(Fine)	1,293	353	5,172	6,387
Stationary Platen	(Coarse)	356	127	1,424	1,924
	(Fine)	8,179	1,927	32,716	38,710
	(Extra Fine)	49,559	10,171	198,236	225,906

6.2 Mesh Quality

During this portion of the testing phase, it was first verified that TTH was correctly transforming tetrahedral elements to hexahedra. For example, Figure 6-5 shows how the wedge with 44 tetrahedral elements was remeshed to 176 hexahedral elements. Similarly Figure 6-6 shows the cylinder case. The next point of concern was to address the quality of the resulting meshes. The following sections study the effectiveness of the mesh quality improvement techniques introduced in 5.2 and 5.3 implemented in

TTH. First, the initial tetrahedral elements are considered with respect to moving the interior node to improve the resulting hexahedra. Second, the techniques applied to the resulting hexahedra are examined.

44 Tetrahedra

176 Hexahedra

Figure 6-5 Wedge TTH mesh

4050 Tetrahedra

16476 Hexahedra

Figure 6-6 Cylinder TTH mesh

6.2.1 Improvements to Existing Tetrahedra (Pre-Splitting)

During the development of TTH, the TetCentroid function (see Appendix 3) was created to modify the interior node position placed originally at the centroid of the volume. A series of tests were carried out to determine the appropriate aspect ratio of the tetrahedron at which to begin a shift of the interior node, as it is important that the four hexahedral children be as well formed as possible.

TetCentroid operates first by evaluating the quality of the tetrahedral elements. Figure 6-7 compares the aspect ratio of the tetrahedral elements to the two extreme (maximum and minimum) skew angles of its four hexahedral children based on the throw test case (fine mesh). A comparison of the hexahedral aspect ratios and the normalized¹⁴ tetrahedral aspect is superimposed onto the same figure. It is clear that the hexahedral skew angles exceed the limits (less than 45° and bigger than 135°) when the aspect ratio of the original tetrahedron exceeds 1.5. Although not shown, similar results were noted for other test cases. As a result the TetCentroid routine will shift the midbody node toward the furthest vertex for all tetrahedra with an aspect ratio of 1.5 or more.

The decision of how much to shift the interior node must also be considered. To quantify the size of the shift both the aspect ratio and skew angles of the resulting hexahedra were considered. Testing quickly showed the improvements to skew angles

¹⁴ Normalized tetrahedral aspect ratio = $\gamma_N = \frac{\gamma}{8.47967}$ equation [4-5]

were more significant. Figure 6-8 based on the stationary platen, wedge and throw test, presents the percentage of skewed angle elements vs. the percentage of movement as a function of length for different test cases. The typical trend is a continual improvement of the element quality as the magnitude of the movement increases, but the rate of improvement declines significantly after the node has moved about 15% of the distance. This limit also minimizes the number of elements which might experience face warpage as a result of excessive movement.

The improvement of the skewed elements in a mesh can be as much as 37% as is shown in Figure 6-8. These values are highly dependent of the type of the mesh since meshes with badly shaped elements may be very difficult to correct significantly.

Figure 6-7 Mesh quality comparison¹⁵

¹⁵ The comparison between the aspect ratio of the tetrahedral mesh and the skew angles and aspect ratios of their hexahedral children, is shown for the Throw fine mesh.

Figure 6-8 Interior node relocation

6.2.2 Improvements to Hexahedra (Post-Splitting)

The mesh quality after the splitting operation varies. In general, for complex geometries with internal and external features, a finer original tetrahedral mesh generates a better overall quality in the hexahedral mesh than do coarser tetrahedral meshes. For example, in the three mesh densities that are used with the stationary platen geometry, the extra fine tetrahedral mesh produced a hexahedral mesh with 99% of its elements within the acceptable range; i.e., aspect ratios less than 2.5 and skew angles between 45° and 135° . From a pre-processing perspective, there are obvious advantages in having fine meshes. The biggest benefit is the reduction of time

consumed in the smoothing operations, and the improved orthogonality of the resulting mesh. These are due to the lesser number of elements requiring smoothing operations. However, the mesh is impractically fine for iterative and incremental FEA associated with complex thermo-elasto-plastic problems. Three operations are considered for the hexahedral elements. First, a correction is applied to correct skewed elements. Second, aspect ratios are corrected. Finally, isoparametric smoothing is applied.

6.2.2.1 Skew Rotate

The Skew Rotate routine is performed to individual elements. It is designed to correct the out-of-range skewed angle elements. The TTH code employs two sequences of Skew Rotate during the smoothing operation; each sequence is a combination of a forward sweep and a backward sweep of the routine. The highest level of improvement occurs after the first initial forward (F) sweep through the mesh, while the backward (B) sweep still improves the mesh but to a lesser degree. The backward sweep is aimed at rotating the larger or smaller extreme angles closer towards the nominal.

Figure 6-9 is a plot of skew angles for the hexahedra in the throw test case with the coarse mesh. As discussed earlier skewed angles should not be less than 45° or over 135° . The skewed angles are shifted closer towards the limits after the backward (B) sweep. The number of acceptable elements increases by only 3% - 5%, as compared to the 15% increase after the initial forward (F) Skew Rotate. The two peaks at 45° and 135° are the result of the improvements made by Skew Rotate, where out-of-range angles are shifted towards the acceptable range.

Figure 6-9 Percentage of elements in the mesh vs. skew angles

6.2.2.2 AR Fix

The aspect ratios for hexahedra are less difficult to fix than the skew angles for the TTH program. The aspect ratios after the splitting procedure are more localized in comparison to the skew angles. The majority of the aspect ratios are distributed close to the acceptable range (aspect ratio less than 2.5).

Figure 6-10 shows the worst case encountered in the test trials, for the wedge. This particular case has 31% of its post-split elements out of the aspect ratio limits, but the

worst aspect ratio is less than 5.5. The arrangement of AR Fix after Skew Rotate is largely due to this factor. A study on TTH has shown that a slight increase in the number of out-of-range aspect ratio elements of about 3% - 5% after the Skew Rotate function is typical. The slight increase is easily corrected by applying AR Fix, after the forward and backward sweeps of Skew Rotate.

Figure 6-10 Number of elements (%) vs. aspect ratio of hexahedra

There are certain drawbacks in placing AR Fix after Skew Rotate. It is difficult to correct both qualities simultaneously; fixing skew angles of one element might have an adverse effect on the quality of its adjacent elements. Prior to the implementation of the

procedures, a study on the effect on the smoothing order, i.e., Skew Rotate after AR fix and vice versa, has shown that the current configuration is the better of the two. In the current set-up, the test cases showed that there is only a 5% - 10% increase in the number of poor aspect ratio elements after the Skew Rotate procedures. This is easily offset by the average of 20% improvement of the number of acceptable elements after the AR Fix procedure.

Similar to the Skew Rotate routine, AR Fix is implemented twice in TTH and it is placed after the Skew Rotate routines. The best feature about AR Fix is not necessarily the amount of the reduction of bad quality elements. The importance of this feature is the improvement made to the high aspect ratio elements. Figure 6-11 demonstrated this effect in the coarse mesh version of stationary platen; approximately 7% of the elements have aspect ratios that exceed four after the backward sweep of the first sequence for Skew Rotate. After the AR Fix procedure, only 2.8% of the elements have aspect ratios higher than four.

In a few cases, some originally deteriorated aspect ratio elements were worse after the AR Fix procedures were applied. However the conditions can be corrected through the final procedure of isoparametric smoothing also shown in Figure 6-11. This deterioration is caused by the offset of the elements being corrected by AR Fix, where the surrounding elements are squeezed or stretched. The condition is only temporary and in all cases the quality of the final meshes benefits from TTH smoothing procedures.

Figure 6-11 Comparison of the element aspect ratios after first smoothing techniques

6.2.2.3 Isoparametric

The isoparametric smoothing technique is a well-proven smoothing algorithm [49]. It improves the mesh quality by increasing the orthogonality of the elements. The improvement is achieved through simple procedures of averaging node positions, which are evaluated based on the neighbouring elements. Usually the isoparametric method works well if the majority of the neighbouring elements are well formed, but it is ineffective for the cases where the surroundings are dominated by poor quality

elements. Unfortunately, the latter case tends to occur in meshes with complex geometries. Even though these cases are first processed by the Skew Rotate and AR Fix routines prior to the isoparametric procedures, there are side effects. In some scenarios, the qualities of the neighbours of the smoothed element are worsened. This case is shown in Figure 6-12, which presents the result for the first smoothing sequence of the wedge test case. The three curves in the figure represent the original mesh, the post-Skew Rotate mesh and post-isoparametric smoothing. The minimum and maximum angles for the original cases are approximately 15° and 165° respectively. These extreme angles expanded to 10° and 175° after the completion of the first sequence of Skew Rotate, which includes a forward and backward sweep of the mesh. In spite of this, the overall quality of the mesh significantly improved as shown by the reduction of the number of elements outside the limits. The result for post-isoparametric smoothing is the reduction of the extreme angles to a more acceptable skew angles of 30° and 160° , but accompanied by a slight increase in the number of poor quality elements.

In general the isoparametric technique for TTH does not reduce the number of poor quality elements. However, it is necessary for the improvement of mesh orthogonality. An improved mesh orthogonality will pull-in the global maximum and minimum skew angles that were adversely affected during the geometry based smoothing routines.

Figure 6-12 Number of elements vs. skew angles after smoothing procedures

6.3 TTH Meshes

The combined pre-splitting treatment of tetrahedral elements (interior node shift) followed by the post-splitting treatments of Skew Rotate, AR Fix and isoparametric smoothing applied to the hexahedral elements, generate good meshes for FEA.

TTH converts complex tetrahedral meshes to hexahedral meshes as shown in Figure 6-13 for the throw test case (5172 hexahedral elements) and Figure 6-14 the stationary platen.

Figure 6-13 Throw TTH mesh

Figure 6-14 Stationary platen mesh after TTH

The combined smoothing functions correct the overall mesh quality. Some examples of individual element smoothing from the throw fine mesh are shown in Figure 6-15. In this example a maximum variation of 10° in the skew angles correction was obtained.

Figure 6-15 Smoothing process on individual elements

6.4 Multi-Platform Capabilities

Today most FEA programs are available in multiple platforms, but some FEA codes remain platform specific for various reasons; i.e., Cubit is designed to work under UNIX and Linux, but it does not support WinNT. In order for FEA codes to benefit from TTH, it is necessary to program it for the mainstream Operating System (OS) platforms. The TTH program was tested in WinNT, Unix and Linux platforms. Table 6-2 presents OS information of the platforms compatible with TTH along with the corresponding

specifications of the computers on which it was tested. Computer specifications in conjunction with the OS play a major role in determining the allowable mesh size and the runtime for the program.

Table 6-2 Computer specifications

	LINUX	UNIX	Windows NT
Processor	Intel P3	R 10000 cpu	Intel P3
OS	Red Hat version 7.1	Irix 6.5.10	Windows NT 4.0 service pack 6
Clock speed [MHz]	Dual processor 800	175	450
RAM [MB]	512	128	128
File System	exF2	XFS	NTFS

The runtime has always been a primary focus throughout the development cycle of the TTH program. The design parameter of TTH calls for a quick mesh transformation add-on program.

Table 6-3 is a summary of some of the TTH runtime test meshes, performed on the platforms stated above. Overall, the results are satisfactory with majority of the runtimes below a minute. The only exception is the biggest mesh in the test set, which had an initial mesh size of almost 50,000 elements and final mesh size of 200,000 elements. Linux has the best performance as it is demonstrated with a runtime of less than a minute in majority of the cases; even the transformation of the biggest mesh takes only 3.5 minutes. The TTH faster runtime on Linux, can be attributed to the fastest processor of the machines tested. Extra memory and data structures of the OS

also contribute to the efficient runtime. The Unix machine, hampered by its older processor, was the slowest of the three machines that were tested.

Table 6-3 Runtime tests

Part Name		Initial No. of Elements	Initial No. of Nodes	Final No. of Nodes	PLATFORM (hr.: min.: sec.)		
					NT	UNIX	LINUX
Wedge		44	25	267	0:00:04	0:00:10	0:00:03
Cylinder		4,050	970	19,336	0:02:36	0:02:06	0:00:19
Throw	(Coarse)	102	48	595	0:00:08	0:00:13	0:00:04
	(Fine)	1,293	353	6,387	0:00:47	0:00:42	0:00:08
Stationary Platen	(Coarse)	356	127	1,924	0:00:23	0:00:37	0:00:10
	(Fine)	8,179	1,927	38,710	N/A	0:04:24	0:00:50
	(Extra Fine)	49,559	10,171	225,906	N/A	0:24:21	0:03:17

The slower processing speed in Unix does not affect the machine capability of handling large mesh sizes. A series of tests showed the mesh size limits for Unix and Linux machine as 250,000 for both hexahedral elements and nodes. On the other hand, WinNT for that specific type of machine can only handle a maximum final mesh size of 30,000 nodes and 40,000 elements. The dramatic difference between the systems is due to the use of virtual memory¹⁶.

TTH while versatile, cannot dynamically accommodate different machine capabilities. For example it cannot automatically increase the allowable mesh size to take full advantage of a machine with greater computational power. Nor can it scale down the allowable mesh size for systems with lower capabilities. This problem can lead to a premature run failure or unintended overwriting of memory. It is highly recommended that TTH run on computers with the equivalent specifications or better. Alternatively,

¹⁶ Virtual memory, combination of physical memory and memory disk, the latter is also called swap file.

the code can be altered by individual researchers to suit their desired machines.

Appendix 4 is an user manual for the TTH algorithm. It gives a full listing of the functions and files used in the program. A brief description of each function is also enclosed for review. The most important aspect of the manual is a step-by-step instructions that step the reader through all the necessary stages, which include conversion to TTH specified format, mesh transformation, and hexahedral mesh visualization.

7 Discussion

Testing of TTH code demonstrated its usefulness for a quick transformation of tetrahedral element meshes into hexahedra. Although the smoothing procedures cannot completely eliminate the poor quality elements, they improve the overall mesh quality by correcting the aspect ratios and skew angles of poorly shaped elements.

Traditionally the splitting method has not been very popular perhaps because it increases the number of degrees of freedom. However, the quality of the initial tetrahedral meshes can be improved. The first stage of improvement is the insertion of an interior node. This node is placed in a point that optimizes the quality of the hexahedral children of the badly shaped tetrahedra. Testing showed a shift of 15% of the length between the centroid of the element and its furthest vertex, provided a good compromise between face warpage and element quality. A further increase in node displacement results in extensive warpage, but a lesser adjustment reduces the impact on the quality improvement of the elements.

Once the tetrahedra are split into hexahedra, geometry-based smoothing techniques further correct the quality of the mesh. Since aspect ratios were shown to be easier to rectify, the first parameter to be corrected was the skew angle. Application of the isoparametric technique completes the smoothing process by improving the element

orthogonality and pulling in the extreme global skew angles. Results showed that the overall mesh quality and badly shaped elements could be efficiently improved obtaining meshes suitable for FEA studies. These types of meshes are highly refined and by nature, more accurate than the initial tetrahedral meshes. As a whole, the program has the capability of handling a wide array of geometries with a minimal associated computational cost.

8 Conclusions and Future Work

8.1 Conclusions

1. A tetrahedra-to-hexahedra (TTH) conversion software was written in C and tested on several hardware and platforms.

It converts tetrahedral meshes generated by standard CAD/CAM packages such as Pro/ENGINEER and I-DEAS to hexahedral meshes suitable for use by specialty FEA codes such as DYNA and TIAMAT.

2. TTH has two principal areas in which mesh quality is controlled. In the pre-splitting phase, the quality is controlled by adjusting the portion of the interior nodes. In the post-splitting phase the aspect ratio and skew angles of the hexahedra are adjusted. An isoparametric averaging technique is also included to improve the orthogonality of the meshes.

3. Testing using a variety of meshes showed that TTH successfully converts tetrahedral meshes to good quality hexahedral meshes.

8.2 Future Work

1. This thesis work has focused on the conversion of tetrahedral element meshes into hexahedra. Meshes produced by TTH are highly refined, and the large number of elements increment the time consumed in calculation processes. An adaptive coarsening system such as [33] could be implemented in order to reduce computational time and to obtain cost effective results.
2. Further studies could be done repositioning midedge and midface nodes from the tetrahedral elements to be split. Looking for an optimum position for theses nodes, would help to generate better-shaped hexahedral elements.
3. It is common to find objects made with more than one type of material; e.g. bones and filler welded structures. In order to obtain accurate FEA results in these types of objects, it is required that TTH have the capacity of multi-material analysis. This improvement to the code, could be made to enable these types of meshes in TTH.
4. Testing of the free meshed tetrahedral and TTH hexahedral meshes using specialty FEA packages.

9 References

- 1 K. Ho-Le, Finite Element Mesh Generation Methods: A Review and Classification. Computer Aided Design **1**, No. 20, pp. 27-38, 1988.
- 2 Owen, S.J., A Survey of Unstructured Mesh Generation Technology. Proc. 7th Int. Meshing Roundtable, Sandia National Lab, pp. 239-267, October 1998.
- 3 I-DEAS User's Guide, Simulation: Finite Element Modelling. Structural Dynamics Research Corporation, 1999.
- 4 McDill, J.M.J., Oddy, A.S. and Goldak, J.A., An Adaptive Mesh-Management Algorithm for Three-Dimensional Finite Element Analysis. Transactions of the CSME; **15**, pp. 57-70, 1991.
- 5 Adams, V. and Askenazi, A., Building Better Products with Finite Element Analysis. ONWORD PRESS, Santa Fe, USA, 1999.
- 6 McDill, M.J., Package for Pre-Processing and Post-Processing for Tiamat. Carleton University; Ottawa, 1996.
- 7 SIMPLE, A multipurpose code for 2-D and 3-D thermal-elasto-plastic FEA. Technical University of Luleå; Luleå, Sweden.
- 8 ABAQUS User's Manual, Version 6.2. Hibbitt, Karlsson & Sorensen Inc. Pawtucket, RI, USA; 2001.
- 9 [http:// plastics.about.com/cs/swfiniteelement/](http://plastics.about.com/cs/swfiniteelement/)
- 10 Dewhirst, D.L. and Ford Motor Company, ASM Handbook Materials Selection and Design Finite Element Analysis; **20**, ASM International, Materials Park, OH; pp. 176-182, 1997.
- 11 Society of Automotive Engineers (SAE), Fatigue Design Handbook 3rd edition AE-22, 1997.
- 12 Wang, S.P., Choudhry, S. and Wertheimer, T.B., Comparison Between the Static Implicit and Dynamic Explicit Methods for FEM Simulation of Sheet Forming Process. MARC Analysis Research Corporation, Palo Alto, AC, USA. [http:// www.marc.com/ support/Library/numiform1.pdf](http://www.marc.com/support/Library/numiform1.pdf).

- 13 Canann, S.A., Saigal, S. and Owen, S.J., *Int. J. Numer. Meth. Engng*; **49**, Issue 1-2, 2000.
- 14 Owen, S.J., Staten, M.L., Canann, S.A. and Saigal, S., Quad-Morphing: Advancing Front Quad Meshing using Triangle Transformations. *Proc. 7th Int. Meshing Roundtable*, Sandia National Lab; pp. 409-428, October 1998.
- 15 Fine, L., Remondini, L. and Leon, J.C., Geometry Adaptation of Polyhedral Models Based on Mechanical Criterion. *Advances in Finite Element Procedures and Techniques*. Civil-Comp Press, Edinburgh; pp. 159-167, 1998.
- 16 Lee, K., *Principles of CAD/CAM/CAE Systems*. Chapter 8: Finite-Element Modeling and Analysis; Ed. Addison Wesley, Boston, pp. 914-254, 1999.
- 17 Rank, E., Schweingruber, M. and Sommer, M., Adaptive Mesh Generation and Transformation of Triangular to Quadrilateral Meshes. *Communications in Numerical Methods in Engineering*; **9**, pp. 121-129, 1993.
- 18 Benzley, S. E., Perry, E., Merkley, K., Clark, B. and Sjaardama, G.D., A Comparison of All Hexagonal and All Tetrahedral Finite Element Meshes for Elastic and Elasto-plastic Analysis. *Proc. 14th Int. Meshing Roundtable*, Albuquerque, USA, 1995.
- 19 Papke, N., Fourth Year Project: Final Report. Preliminary Finite Element Stress Analysis of a Husky Injection Molding Machine Stationary Platen. Department of Mechanical and Aerospace Engineering; Carleton University, Ottawa, 1998.
- 20 LS-DYNA, Theoretical Manual. Livermore Software Technology Corp; Livermore, California, USA. May 1998.
- 21 Johnston, B.P., Sullivan Jr., J.M. and Kwasnik, A., Automatic Conversion of Triangular Finite Element Meshes to Quadrilateral Elements. *Int. J. Numer. Meth. Engng*; **31**, pp. 67-84, 1991.
- 22 Halpern, M., Industrial Requirements and Practices in Finite Element Meshing: A Survey of Trends. *Proc. 6th Int. Meshing Roundtable*, Sandia National Lab; pp. 399-411, October 1997.
- 23 <http://endo.sandia.gov/cubit/help-version7/cubithelp.html>.
- 24 Cifuentes, A.O and Kalbag, A., A Performance Study of Tetrahedral and Hexahedral Elements in 3-D Finite Element Structural Analysis, *Finite Elements in Analysis and Design*; **12**, pp. 313-318, 1992.

-
- 25 Lindgren, L.E., Haggblad, H.A., McDill, J.M.J., Oddy, A.S., Automatic Remeshing for Three - Dimensional Finite Element Simulation of Welding. *Comput. Methods Appl. Mech. Engng*; **147**, pp. 401-409, 1997.
- 26 Sing, L.T, Meshing and Substructuring of 3-D Stress Analysis Models. Ph.D. Thesis, Department of Computer Science Faculty of Science, The Queen's University of Belfast, Belfast, 1994.
- 27 Lai, M., Benzley, S. and White, D. Automated Hexahedral Mesh Generation by Generalized Multiple Sources to Multiple Target Sweeping. *Int. J. Numer. Meth. Engng*; **49**, pp. 261-275, 2000.
- 28 Kittur, M.G and Huston R.L, Finite Element Mesh Refinement Criteria for Stress Analysis. *Computer & Structures*; **34**, pp. 251-255, 1990.
- 29 Murdoch, P., Benzley, S., Blacker, T. and Mitchell, S.A., The Spatial Twist Continuum: A Connectivity Based Method for Representing and Constructing All-Hexahedral Finite Element Meshes. *Finite Elements in Analysis and Design*; **28**, pp. 137-149, 1997.
- 30 Leland, R.W., Melander, D.J., Mitchell, S.A. and Tautges, T.J., The Geode Algorithm: Combining Hex/Tet Plastering, Dicing and Transition Elements for Automatic, All-Hex Mesh Generation. *Proc. 7th Int. Meshing Roundtable*, Sandia National Lab; pp. 515-521, October 1998.
- 31 Sheffer, A., Etzion, M., Rappoport A. and Bercovier, M., Hexahedral Mesh Generation using the Embedded Voronoi Graph. *Proc. 7th Int. Meshing Roundtable*; pp. 347-364, October 1998.
- 32 Yu, G., Masubuchi, K., Maekawa, T., and Patrikalakis, N., A Finite Element Model for Metal Forming by Laser Heating. *Proceedings of the 10th International Conference on Computer Applications in Shipbuilding*. K. Johansson and C. Chrysosostomidis editors, MIT, Cambridge, MA. June 1999.
- 33 McDill, J.M.J and Oddy, A.S., Arbitrary Coarsening for Adaptive Mesh-Management in Three-Dimensional Automatic Finite Element Analysis. *Math Modelling and Sci. Computing*; **2**, pp. 1072-1077, 1993.
- 34 C.K Lee and S.H Lo, A New Scheme for the Generation of a Graded Quadrilateral Mesh. *Computers & Structures*; **52**, pp. 847-857, 1994.
- 35 Benzley, S.E., Perry, E., Merkley, K. and Clark, B., A Comparison of All Hexagonal and All Tetrahedral Finite Element Meshes for Elastic and Elasto-plastic Analysis. *Proc. 4th Int. Meshing Roundtable*, Sandia National Lab; pp. 179-191, October 1995.
- 36 <http://www.thermoanalytics.com/support/meshing/mesh-review.html>.

-
- 37 White, D.R., Lai, M., Benzley, S.E. and Sjaardema, G.D. Automated Hexahedral Mesh Generation by Virtual Decomposition. Proc. 4th Int. Meshing Roundtable, Sandia National Lab; pp. 165-176, October 1995.
- 38 Whiteley, M., White, D., Benzley, S. and Blacker, T., Two and Three Quarter Dimensional Meshing Facilitators. Engineering with Computers, Springer-Verlag; **12**, pp. 155-167, December 1996.
- 39 Armstrong, C.G., Robinson, D.J., McKeag, R.M., Li, T.S., Bridgett, S.J., Donaghy, R.J. and McGleenan, C.A, Medials for Meshing and More. Proc. 4th Int. Meshing Roundtable, Sandia National Lab; pp. 277-288, October 1995.
- 40 Knupp, P.M., Next-Generation Sweep Tool: A Method for Generating All-Hex Meshes on Two and One-Half Dimensional Geometries. Proc. 7th Int. Meshing Roundtable, Sandia National Lab; pp. 505-513, October 1998.
- 41 Schneiders, R., A Grid-Based Algorithm for the Generation of Hexahedral Element Meshes. Engineering with Computers; **12**, pp. 168-177, 1996.
- 42 Blacker, T. D., Paving: A New Approach to Automated Quadrilateral Mesh Generation. Int. J. Numer. Meth. Engng, John Wiley; **32**, pp. 811-847, 1991.
- 43 Tautges, T.J., Blacker, T., Mitchell, S.A., The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes. Int. J. Numer. Meth. Engng, John Wiley; **39**, pp. 3327-3349, 1996.
- 44 S.J. Owen and Saigal, S., H-Morph: An Indirect Approach to Advancing Front Hex Meshing. Int. J. Numer. Meth. Engng; **49**, pp. 289-312, 2000.
- 45 Oddy, A., Goldak, J., McDill, M. and Bibby, M., A Distortion Metric for Isoparametric Finite Elements. Transaction of the CSME; **12**, pp. 213-217, 1988.
- 46 Knupp, P.M., Matrix Norms & the Condition Number: A General Framework to Improve Mesh Quality Via Node-Movement. Proc. 8th Int. Meshing Roundtable, South Lake Tahoe, CA, U.S.A; pp. 13-22, October 1999.
- 47 Knupp, P.M., Achieving Finite Element Mesh Quality Via Optimization of the Jacobian Matrix Norm and Associated Quantities. Part I – A Framework for Surface Mesh Optimization. Int. J. Numer. Meth. Engng; **48**, pp. 401-420, 2000.
- 48 [http:// ksm.fsv.cvut.cz/~dr/papers/Thesis?node7.html](http://ksm.fsv.cvut.cz/~dr/papers/Thesis?node7.html).
- 49 Hyun, S. and Lindgren, L.E., Smoothing and Adaptive Remeshing Schemes for Graded Elements: Mesh Smoothing Techniques for Graded Elements. NUMIFORM'98, Netherlands, June 1998.

-
- 50 Shimada, K., Liao, J. and Itoh, T., Quadrilateral Meshing with Directionally Control through the Packing of Square Cells. Proc. 7th Int. Meshing Roundtable, Sandia National Lab; pp. 61-76, October 1998.
- 51 Kalhori, V., Modelling and Simulation of Mechanical Cutting. Doctoral thesis Luleå University of Technology; Luleå, Sweden, October 2001.
- 52 Herrmann, L.R, Laplacian-Isoparametric Grid Generation Scheme. Journal of the engineering mechanics division. Proceedings of the American Society of Civil Engineers; **102**, pp. 749-756, October 1976.
- 53 Knupp, P.M., Hexahedral Mesh Untangling and Algebraic Mesh Quality Metrics. Proc. 9th Int. Meshing Roundtable, Sandia National Lab; pp. 173-183, October 2000.
- 54 <http://www.iue.tuwien.ac.at/publications/PhD%20Theses/radi/node90.html>.
- 55 Bentley, J.L, Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM; **18**, pp. 509-517, 1975.
- 56 Parthasarathy V. N. et al, A Comparison of Tetrahedron Quality Measures. Finite Elements in Analysis and Design; **15**, pp. 255-261, 1993.
- 57 Lepi, S.M., Practical Guide to Finite Elements: A Solid Mechanics Approach. Marcel Dekker Inc., London, 1998.
- 58 Kerlick, G.D. and Klopfer, G.H., Assessing the Quality of Curvilinear Coordinate Meshes by Decomposing the Jacobian Matrix. Numerical Grid Generation, J.F. Thompson, ed., Elsevier-North Holland, New York, 1982. Appl. Math. Comput, **10/11**, pp. 787-807, 1982.
- 59 Owen, S.J., Constrained Triangulation: Application to Hex-Dominant Mesh Generation. Proc. 8th Int. Meshing Roundtable; pp. 31-41, October 1999.

Appendix 1 Node translation direction with the different skew angles changes

Action	Action Performed to Faces (a face needs 2 terms)	Node Translation Direction							
		Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
Increase S1	ξ	$-\eta$	η	η	$-\eta$	$-\eta$	η	η	$-\eta$
	η	$-\xi$	$-\xi$	ξ	ξ	$-\xi$	$-\xi$	ξ	ξ
Decrease S1	ξ	η	$-\eta$	$-\eta$	η	η	$-\eta$	$-\eta$	η
	η	ξ	ξ	$-\xi$	$-\xi$	ξ	ξ	$-\xi$	$-\xi$
Increase S2	η	$-\zeta$	$-\zeta$	ζ	ζ	$-\zeta$	$-\zeta$	ζ	ζ
	ζ	$-\eta$	$-\eta$	$-\eta$	$-\eta$	η	η	η	η
Decrease S2	η	ζ	ζ	$-\zeta$	$-\zeta$	ζ	ζ	$-\zeta$	$-\zeta$
	ζ	η	η	η	η	$-\eta$	$-\eta$	$-\eta$	$-\eta$
Increase S3	ζ	$-\xi$	$-\xi$	$-\xi$	$-\xi$	ξ	ξ	ξ	ξ
	ξ	$-\zeta$	ζ	ζ	$-\zeta$	$-\zeta$	ζ	ζ	$-\zeta$
Decrease S3	ζ	ξ	ξ	ξ	ξ	$-\xi$	$-\xi$	$-\xi$	$-\xi$
	ξ	ζ	$-\zeta$	$-\zeta$	ζ	ζ	$-\zeta$	$-\zeta$	ζ
S1 = Skew angle between local ξ - η S2 = Skew angle between local η - ζ S3 = Skew angle between local ζ - ξ									

Appendix 2 Development of the 3-D isoparametric equation

Above figure is a 26 nodes hexahedron with node numbers.

Following is a full expansion of the basis functions for the above 26 nodes hexahedron

$$x_i = \frac{1}{8}(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8) \quad (A)$$

$$-\frac{1}{4}(x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20}) \quad (B)$$

$$+\frac{1}{2}(x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26}) \quad (C)$$

The equation is divided into 3 components as shown above.

Consider (c) first. We know :

$$\frac{1}{6}(x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26}) = \frac{1}{3NE} \sum_1^{NE} (x_a + x_b + x_c)$$

where x_a , x_b and x_c are the 3 connecting nodes as the following figure illustrated.

The figure above is an illustration of the linear hexahedron with connecting and opposite nodes labelled

So,

$$\frac{1}{2}(x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26}) = \frac{3}{3NE} \sum_1^{NE} (x_a + x_b + x_c)$$

Consider (B)

$$-\frac{1}{4}(x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{20}) = \frac{1}{NE} \sum_1^{NE} (x_a + x_b + x_c)$$

Consider (A)

$$\frac{1}{8}(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8) = \frac{1}{NE} \sum_1^{NE} (x_d) \quad \text{Where } x_d \text{ is unconnected}$$

Combining them,

$$x_i = \frac{1}{NE} \sum_1^{NE} (x_d) - \frac{1}{NE} \sum_1^{NE} (x_a + x_b + x_c) + \frac{1}{NE} \sum_1^{NE} (x_a + x_b + x_c)$$

The final equation is,

$$x_i = \frac{1}{NE} \sum_1^{NE} (x_d)$$

Example of variation of Isoparametric equations:

Fully isoparametric version

$$x_i = \frac{1}{NE} \sum_1^{NE} (x_d)$$

Fully Laplacian

$$x_i = \frac{1}{3NE} \sum_1^{NE} (x_a + x_b + x_c)$$

Mixed Formulation

$w = 1$ Fully isoparametric approach

$w = 0$ Fully Laplacian approach

$$x_i = \frac{1}{3NE} \left(\sum_1^{NE} w(x_a + x_b + x_c) + 3(1 - w)x_d \right)$$

Appendix 3 TetCentroid Function

Appendix 4 TTH user manual

Documentation and User Instructions for TTH Converter Code

Written by:

Alejandra Carmona G.

This document was written with the assistance of Mr. Andrew Rader

**Carleton University
2002**

LIST OF FILES AND FUNCTIONS

4X4det.c ThreeXThree FourXFour VolumeTetra AR.h CALCELXYZ CALCDPDR Vectorprod AR SKEWWELEMENT SKEWS NEWSKEWS BadskewsElementDisplay AR_TET.h ARTET GrphSkews SFHTET HexLocal2GlobalCoord SkewRotate GrphARHex SRcoormove ARFix.h ARFixCases ARFix ARTet2HexStat.h ARTet2HexStat c_connect.h CONECT SMOOTHNODE NDPLACE NDNAYELEM c_JACOB3D.h JACOB3D INVJ c_SFHBR.h SFHBR	extra.h RepeatNdDetector ElPerNDDisplay NumberOfBadElements onehex findrst_c.c POINTS3D_C FINDRST_C fulgar_c.c FACECHK FULGAR GrphARTET.h GrphARTET pt2.h (CellTable; Pttables explanations and limits) Pt22.h : PTGet_ ptnew_ ptput_ Functions contained in the file but not in use: PTCheck PTMakeKey PTEqual PTEqualPoint PTLessPoint PTGreaterPoint PTLessEqualPoint PTIsIn PTIntPut PTInq_ PTGetR PTGetC PTSave ptrest_ Tetcentroid.c TETCENTROID
---	--

FUNCTIONS BY TYPE

Mesh Division Functions	Smoothing Functions	Testing Functions
AR ARTET CALCDPDR CALCELXYZ INVJ JACOB3D Pt_get Pt_new Pt_put Pt_table SFHBR SFHTET SKEWS Tetcentroid vectorprod	ARFix CONECT FACECHK findrst_c fulgar ndnayelem NDPLACE NEWSKEWS POINTS3D_C SkewRotate SMOOTHNODE	BadskewsElementDisplay ElPerNDDisplay GraphARHex GraphARTET GrphSkews NumberOfBadElements onehex SRcoormove RepeatNdDetector SKEWSELEMENT

FUNCTION DESCRIPTION**AR**

Finds the aspect ratios of the element using Kerlick and Klopfer equations.

ARFix

Smoothing function to repair the hexahedra aspect ratios.

ARTET

Calculates the aspect ratio of tetrahedra.

BadskewsElementDisplay

Creates a file listing the extreme skew angle of each element and determining if the value is out of range.

CALCDPDR

Calculates the shape functions derivatives of (ξ, η, ζ) .

CALCELXYZ

Translates the nodal coordinates of an element into specified arrays.

CONNECT

Subroutine used to find the connecting nodes from neighbour elements.

ElPerNDDisplay

Displays the nodes numbers and coordinates of an element.

FACECHK

Checks all 6 FACES of a 3-D element to see if it connects 2 elements.

findrst_c

Given a point in space and the element it is within, this function finds the local (ξ, η, ζ) coordinate within the element.

FourXFour

Calculates the determinant of a 4x4 matrix.

fulgar

Finds external faces in the 3-D mesh.

GraphARHex

Classifies in intervals the values of the hexahedra aspect ratios. The values can be exported to spreadsheet programs for further analysis.

GraphARTET

Classifies in intervals the values of the tetrahedra aspect ratios. The values can be exported to spreadsheet programs for further analysis.

GrphSkews

Classifies in intervals the values of the hexahedra skew angles. The values can be exported to spreadsheet programs for further analysis.

INVJ

Calculates the inverse of the Jacobian matrix in a hexahedral element.

JACOB3D

Calculates the Jacobian matrix of a hexahedral element.

NDNAYELEM

Returns the number of neighbouring elements for the specific node.

NDPLACE

This procedure selects the two nodes to smooth. This selection is based on the skew of the element and the position i.e. interior, exterior node. The coupling nodes should be

placed in opposite faces to each other and at least one of them should be an interior node.

NEWSKEWS

Calculates the skew angles in an element.

NumberOfBadElements

Creates a file that displays the number of elements in the mesh and the number of badly shaped elements. It also gives the number of elements with poor aspect ratio, low or high skew angles. It also displays the global extremes for both aspect ratios and skew angles.

onehex

Creates a file to visualize a hexahedral element in Pro/ENGINEER.

POINTS3D_C

Transforms global coordinates to local coordinates.

Pt_get

Looks for information in K-D tree data structure.

Pt_new

Creates a new record in K-D tree.

Pt_put

Put new record in K-D tree.

Pt_table

Creates a K-D tree.

RepeatNdDetector

Detects duplicated nodes.

SFHBR

Determines the value of the shape functions for a hexahedral element.

SFHTET

Determines the value of the shape functions for a tetrahedral element.

SkewRotate

Smoothing function that repairs the skew angles of distorted hexahedra.

SKEWS

Calculates the skew angles in an element. It stores the out of range element numbers and angles.

SKEWSELEMENT

Creates a file with the skewed elements values and their nodes coordinates.

SMOOTHNODE

Isoparametric smoothing function.

SRcoormove

Prints in a file the coordinates of an specific element during the smoothing process.

Tetcentroid

Computes the new value of the interior node for distorted tetrahedral elements.

ThreeXThree

Calculates the determinant of a 3x3 matrix.

Vectorprod

Determines the dot product between two vectors.

VolumeTetra

Calculates the volume of a tetrahedral element.

USER INSTRUCTIONS

Finite element analysis is a popular numerical technique for the analysis of engineering designs with complicated geometry and/or non-homogeneous material properties. It uses a computer model to approximate the mechanical behaviour of components under various environments and stress states. Finite element methods can be applied to two or three-dimensional models. Three-dimensional (3-D) models provide more accurate results, but at a cost of higher computational requirements. In order to apply FEA to a system, the object must be discretized into elements. In 3-D, these elements are generally tetrahedra or hexahedra. The points where elements are connected are called nodes.

Most commercial CAD/CAM packages such as Pro/ENGINEER or I-DEAS can be used for element mesh creation for FEA but they are unable to fully mesh complicated geometries with hexahedral elements. Hexahedral elements are better suited to many FEA applications than tetrahedral elements. It is thus desirable to have a computer program that facilitates the conversion of tetrahedral elements to hexahedral elements to be used in FEA applications.

This document outlines the procedure followed for using the TTH (Tetrahedra-to-Hexahedra) program with solid models generated in Pro/ENGINEER or I-DEAS. The steps in using the program are described below.

1.0) Exporting the model from Pro/ENGINEER or I-DEAS

Note that a model must already have created in Pro/ENGINEER or I-DEAS. First the model must be loaded. The steps below should then be followed to convert the Pro/ENGINEER or I-DEAS model.

1.1) Pro/ENGINEER

In Pro/ENGINEER, the user must select **applications > mechanica** from the toolbar and check the **FEM mode** box.

Again in the toolbar, the user must select **structure > model > mesh > create > solid > start**.

The mesh size can be adjusted by selecting **create** or **edit** under **mesh control** on the menu on the right side of the screen.

The model can then be exported by selecting **run**. ANSYS must be selected for exporting. This will give the Pro- ENGINEER file an ANSYS file extension (**.ans**). The user should enter a file name at the "output to file" field. The file should be placed in the directory that contains the FEA conversion program (**cproe.exe**).

1.2) I-DEAS

In I-DEAS, on the menu at the top right of the screen, the user should select **Simulation** mode instead of **Design** mode. It will take a few minutes to load. On the same menu, **Master Model** should be changed to **Meshing**.

On the top left square of the menu at the right of the screen the user should select **Define Solid Mesh** (mesh picture at the top left square).

In the Create or Select FE Model menu that appears, the model must be given a name. Then, the volume to be meshed must be selected.

The mesh can now be configured in the Define Mesh screen that appears. The mesh should be configured as desired by selecting the mesh type, element length, element family and element type. **Free** should be selected as the mesh type. **Solid** should be selected as the element family and **Linear** should be selected as the element type. The mesh can now be previewed selecting the **Preview Mesh** button at the bottom right corner. The mesh size and node numbering can be found by selecting in the Modify

Mesh Preview screen the **Estimate Solids** button. If the mesh is satisfactory select **Keep Mesh**, otherwise the **Cancel Mesh** option should be selected to redefine the mesh properties.

The mesh is exported by selecting **file > export** and selecting **I-DEAS SUF Simulation Universal File** for the file type.

A file name should be given. The selected directory must be the one that contains the FEA conversion program.

After the model is exported, the data conversion to the proper format should be done by calling **cideas.exe**.

2.0) Converting the Pro/ENGINEER or I-DEAS File to TTH Input Data Format

A converter program is required to convert the Pro/ENGINEER or I-DEAS model file to TTH input data format. There is a separate program for each file type. **cproe.c** is used to convert Pro/ENGINEER models, while **cideas.c** is used for I-DEAS models.

The program requests for the model filename. The full model file name, including the extension, i.e. **file.unv** or **file.ans**, should be entered.

The conversion produces the output text file automatically as **output.txt**. This converted file contains two types of data. The first part includes the element numbering and each element's node composition. The second part of the file contains the node numbering and their spatial coordinates (x, y, z).

3.0) Converting the Tetrahedral Elements to Hexahedral Elements (TTH)

The converted file at this point still contains data for the model with tetrahedral elements. The data should now be converted to a model with hexahedral elements. This is accomplished by running the file **TTH.c**. This program automatically searches for the **output.txt** as the default input file. The user is prompted to enter a name for the TTH output file including the extension, for example, **output2.txt**.

4.0) TTH Compilation and Execution

If a change in the main parameters is required, the steps to compile the program are shown in the following compiling table. If changes are not required, the program can be executed without the compiling procedure.

Compiling Table for different Platforms

LINUX	UNIX	Windows NT (Visual C++)
<ul style="list-style-type: none"> - Compile gcc -o <u>tth.out</u> TTH.c -lm - Execute program: <u>tth.out</u> 	<ul style="list-style-type: none"> - Compile cc -o <u>tth.out</u> TTH.c -lm - Execute Program: <u>tth.out</u> 	<ul style="list-style-type: none"> - Compile (Ctrl + F7) - Build (F7) - Execute Program (Ctrl + F5)

5.0) TAURUS Visualization

TAURUS, a post-processing visualization software, is used for mesh visualization. First, it is necessary to convert the TTH output to a file that TAURUS can recognize using the

following steps:

In UNIX the file **grpmsh.run*** must be called. **Grpmsh** translates a TIAMAT mesh into a TAURUS-compatible file [1].

The user is prompted to enter the name of the file to convert. Enter the name of the output file from the previous step including the extension, in that case **output2.txt**. The user is then prompted to enter a name for the output file after the graphic conversion without the extension, for example, **output3**.

Select "MODE 0" for geometry only.

The mesh can be now generated by calling **tau940*** and specifying the input filename, **output3**. TAURUS has quite a few utilities that allows user to manipulate the mesh.

REFERENCES

- [1] McDill, M.J., Package for Pre-Processing and Post-Processing for Tiamat. Carleton University, Ottawa, 1996.