AN AUTOMATIC ADAPTIVE DIRECTIONAL VARIABLE P-VERSION METHOD IN 3D CURVED DOMAINS

By

Xiaojuan Luo

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Mechanical Engineering

Approved by the
Examining Committee:

Prof. Mark S. Shephard, Thesis Adviser

Dr. Robert L. Spilker, Member

Dr. Kenneth E. Jansen, Member

Dr. Suvranu De, Member

Rensselaer Polytechnic Institute
Troy, New York

Augest 2005
(For Graduation Augest 2005)

AN AUTOMATIC ADAPTIVE DIRECTIONAL VARIABLE P-VERSION METHOD IN 3D CURVED DOMAINS

By

Xiaojuan Luo

An Abstract of a Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
Major Subject: Mechanical Engineering
The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Prof. Mark S. Shephard, Thesis Adviser
Dr. Robert L. Spilker, Member
Dr. Kenneth E. Jansen, Member
Dr. Suvranu De, Member

Rensselaer Polytechnic Institute
Troy, New York

Augest 2005
(For Graduation Augest 2005)

© Copyright 2005
by
Xiaojuan Luo
All Rights Reserved

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I am grateful to many people for their support and encouragement throughout the several years of my doctoral studies.

I thank my thesis advisor, Prof. Mark S. Shephard, for his support and guidance for my research work. I thank him for his efforts to help me get a better understanding of the p-version finite element method and provide valuable input in every aspect of the work presented in this thesis. I thank the members of my examination committee for their time and effort in reviewing the thesis.

I thank the staff and my student colleagues at the Scientific Computation Research Center at RPI that I had many discussions and gained a lot input for my thesis work.

I thank all the engineers in Simmetrix Inc. that provides a helpful working environment for me to accomplish my thesis work. In particular, I would like to thank Bob O'Bara for many helpful discussions on Bezier geometry. I also thank Engineering Software Research and Development Inc. for their generosity to let me use the p-version analysis software StressCheck for the analysis results presented in this thesis.

I thank my parents for all their support, love and taking care of my lovely daughter Maggie when I finished this thesis. I also thank my daughter's father, Chen Quan, for all his support in the nine years we shared life together.

ABSTRACT

Realization of the full benefits of p-version finite elements for general 3D curved domains requires (i) careful construction and control of the mesh layout and geometric approximation, (ii) effective three dimensional adaptive control, and (iii) efficient numerical methods for high order method. This thesis addresses the first two issues in solving linear elliptic problems in general 3D curved domains.

An automatic p-version mesh generation procedure to construct near optimal p-version meshes is presented. Starting from a surface triangulation, the process automatically isolates and meshes the singular model features and thin sections with structured cylindrical graded meshes and one layer prismatic thin section meshes respectively. A p-version mesh curving procedure with gradation and thin section mesh control curves all the mesh entities on the model boundaries. Controlled curved local mesh modifications are applied to correct the invalid mesh entities caused by curving the mesh entities. Examples demonstrated that the curved p-version meshes produced by this procedure save $15\% \sim 40\%$ fewer elements comparing to the curved meshes generated by curving without gradation and thin section control. The simulation results presented further demonstrate these meshes provide superior simulation results with less computational effort.

The performance of the p-version meshes are investigated in the p-version analysis software StressCheck and Trellis. Trellis takes advantage of a topology based high order hierarchic shape function structure to allow the assignment of different p-order for the independent coordinates of the mesh entities in the adaptive analysis. A directional correction indicator is derived from the computation of the elemental residual error estimator such that the p-orders are adaptively enriched in the appropriate directions. Linear elasticity examples demonstrate that the p-version meshes, particular in the adaptive directional p-version analysis, can achieve the same level of solution as conventional p-version method with substantially fewer degrees of freedom.

CHAPTER 1
Introduction

The error of approximation in the finite element method depends on the mesh and the polynomial order of the elements. In conventional FEA codes, the polynomial order of elements are fixed at 1 or 2, and the error is controlled by making sufficiently finer meshes with the errors of approximation reduced as h is reduced. The term h-version refers to this approach. Since the mid-1980's an alternative, known as the p-version, has been extensively researched [12, 48, 49, 66, 67]. In this approach the error is controlled by increasing the polynomial order of elements, denoted by p. The p-version method has the advantage of being able to achieve an exponential rate of convergence in the application to problems of interest that can not be accomplished by h-version method. However, realization of the full benefits p-version finite elements for general three-dimensional curved domains requires (i) careful construction and control of the mesh layout and geometry, (ii) effective three dimensional adaptive control, and (iii) efficient numerical methods for high order method. The technical components required to address these requirements have yet to receive adequate consideration.

The performance of p-version finite element method depends on the character of the exact solution over the domain. In portions of domain where the exact solution is smooth the most effective mesh is one that is as coarse as possible so long as the mesh provides a satisfactory geometric approximation to the domain. However, there are many situations in which the solution of a boundary value problem is not smooth, for example, singularities because of the domain geometry, load function and material properties etc [17, 49]. In these cases, an optimal p-version method requires a carefully designed mesh. For the singularities caused by the corners and edges of the domain, basic theory and example studies [7, 16, 17, 102, 122] demonstrated that meshes must be geometrically graded towards the singularities to attain the exponential rate of convergence.

In the p-version method, the mesh entities on the domain boundary must properly approximate the geometry model to the correct order. There is limited theoretical information available on the level geometric approximation required to maintain convergence to the correct solution for all variables when the domains are curved. A simple analysis based on the relation of approximation theory to the convergence of the error in the energy norm indicates that the energy norm will converge so long as the geometric approximation of the mesh is within one order of that used in the finite element basis [45]. In the cases of elements defined in terms of standard interpolating Lagrange polynomial this requirement is met by being sure that all nodes at mesh vertices, on mesh edges and on mesh faces on curved domain boundaries are placed on the appropriate boundary [47, 117]. If different basis functions for the finite element approximation and element geometries are chosen, one has to carefully consider how to satisfy the geometric approximation requirements for these elements by the proper improvement of the mesh edge and face shapes.

General 3-D curved solid models often have sections where one dimension is small compared to the others. Historically, the methods used to analyze thin sections involved applying deformation assumptions to the 3-D elasticity equations allowing the problem dimensionality to be reduced [105]. The application of such methods requires a reduced dimensional domain model and the identification of the thin sections to be applied model dimension reduction on those portions [105]. Handing the interconnection between two dimensional reduced elements to fully three-dimensional solid elements is another source of difficulty [1]. Since the assumptions corresponding to those deformation models are equivalent to allowing only lower order deformation modes in thickness directions of the thin sections, an alternative is to apply full three-dimensional models discretized with p-version finite elements with low polynomial order through thickness [1, 40, 55, 57, 112]. Therefore a mesh that contains elements through the thickness without through the thickness diagonals is needed.

Geometric approximation, graded mesh layout towards singularities and meshes without long diagonal through the thickness directions for thin sections for a general curved model constitute the three basic requirements to consider in the construction of a near optimal initial mesh for use with the p-version finite element method. The existed mesh generators developed for h-version method do not consider these issues adequately. The first aim of this thesis is to develop an automatic p-version mesh generation procedure that addresses these three basic requirements.

Adaptive finite element analysis procedures employ solution results to estimate the discretization error and determine how to improve the finite element discretization till the prescribed accuracy tolerance is satisfied. An estimation of the error over each element and a strategy to reduce the error by properly changing the polynomial order p over the mesh are the critic ingredients for the adaptive p-version method. The effective implementation of these methods also requires a flexible data structure and methods that support general variations of p-order over the meshes [115].

There are situations for problem of interest that the exact solution exhibits anisotropic behavior, that is, the exact solution exhibit relatively strong gradients in certain directions compared the other directions. Such situations can be caused by the problem physical properties such as the thermal and fluid boundary layers [60], orthotropic material or the model geometries, for example, one of the dimensions of the structural part is much dimensional smaller than the other [1, 57]. An effective adaptive p-version method must be capable of determining these directions and applying high polynomial order for them to capture the exact solution behavior.

Issues on how to effectively adapt the polynomial directionally to achieve the desired level of accuracy with the lowest number of degrees of freedom and computational cost are considered in this thesis. Specific attention is paid to derive the directional correction indicator to determine the directions that need finer resolution compared with other directions.

This thesis is organized with three parts. Part one discuss the basic theory of p-version finite element method and the requirements of its implementation on general 3D curved domains. Part two describes the automatic p-version mesh generation procedure that considers geometric approximation, mesh gradation layout and structured mesh for thin sections together from the beginning that are central to construct near optimal meshes for the use of p-version method. Part three discusses an adaptive directional variable p-version analysis procedure that allow assigning different polynomial orders independently on each coordinate direction. The analysis results related to the p-version method are also presented. Finally, the thesis ends with a summary of contributions and a discussion of future work.

Part I

Overview of p-Version Method and the Requirements of its Implementation on General 3D Curved Domains

CHAPTER 2

Overview of the p-Version Finite Element Method

This chapter first describes the model problem for second order linear partial differential equation on general 3D curved domains. The key components to solve the problem using p-version method is discussed. A priori estimate of the convergence rate of p-version method in energy norm based on the form of the exact solution and assuming optimal finite element meshes configurations are reviewed.

2.1 Model Problem Description

For the second order linear partial differential equations defined on domain Ω that is a bounded region in $\mathbb{R}^{n_{sd}}$ ($n_{sd} = 2,\ 3$ is the problem dimension) with piecewise smooth boundary Γ, let functions $g,\ h$ define the subjected boundary conditions as follows,

$$u \;=\; g \quad on \;\; \mathit{\Gamma}_g, \tag{2.1}$$

$$\alpha \frac{\partial u}{\partial n} \;=\; h \quad on \;\; \Gamma_h, \tag{2.2}$$

$\Gamma = \Gamma_g \cup \Gamma_h$ and $\Gamma_g \cap \Gamma_h = \emptyset$.

The abstract weak form for the second order linear partial differential equations can be stated as follows [6, 71].

Given $f,\ g,\ h$, find $u \in V$ such that for all the test functions $w \in W$,

$$a(w,\ u) = (w,\ f) + (w,\ h)_{\mathit{\Gamma}_h}, \tag{2.3}$$

V defines the trial function space as,

$$V = \{u \mid u \in H^1(\Omega),\ \ u \;=\; g \quad on\ \mathit{\Gamma}_g\}, \tag{2.4}$$

and W defines the variation space as,

$$W = \{w \mid w \in H^1(\Omega),\ \ w \;=\; 0 \quad on\ \Gamma_g\}, \tag{2.5}$$

where $a(\ ,\ )$ is the bilinear form and $(\ ,\ )$ and $(\ ,\ )_{\Gamma_h}$ are the linear forms that depend on the problems specified. There are a number of problems such as heat transfer, porous media flow, laminar flow in a channel and linear elastic solid mechanics that can be represented by the second linear elliptic equation. The examples presented in this thesis focuses on Poisson equation and linear elasticity problem.

In case of Poisson equation, the bilinear and linear forms are defined as [71, 124],

$$a(w,\ u) = \int_{\Omega} w_{,i} u_{,i}\ d\Omega, \tag{2.6}$$

$$(w,\ f) = \int_{\Omega} wf\ d\Omega, \tag{2.7}$$

$$(w,\ h)_{\mathit{\Gamma}_h} = \int_{\mathit{\Gamma}_h} wh\ d\mathit{\Gamma}. \tag{2.8}$$

For linear elasticity problem, the bilinear and linear forms are defined as [71, 124],

$$a(\boldsymbol{w},\ \boldsymbol{u}) = \int_{\Omega} w_{i,j} c_{ijkl} u_{k,l}\ d\Omega, \tag{2.9}$$

$$(\boldsymbol{w},\ \boldsymbol{f}) = \int_{\Omega} w_i f_i\ d\Omega, \tag{2.10}$$

$$(\boldsymbol{w},\ \boldsymbol{h})_{\Gamma_h} = \sum_{i=1}^{n_{sd}} \left(\int_{\Gamma_{h_i}} w h_i\ d\mathit{\Gamma} \right), \tag{2.11}$$

c_{ijkl} are the elastic coefficients.

Let V^h and W^h be the finite dimensional approximation to V and W, respectively. The Galerkin finite element formulation for the weak form in Eq. 2.3 is given as follows [71].

Given $f,\ g,\ h$, find $u^h = v^h + g^h$ such that for all $w^h \in W^h$,

$$a(w^h,\ v^h) = (w^h,\ f) + (w^h,\ h)_{\mathit{\Gamma}_h} - a(w^h,\ g^h). \tag{2.12}$$

2.2 The p-Version Finite Element Method

This section discuss the application of p-version method to solve the Galerkin finite element formulation developed in Eq. 2.12.

2.2.1 Finite Element Mesh

In the finite element method, the domain Ω is discretized with elements Ω^e, $e = 1...n_{el}$ referred to as the finite element mesh. A general topology and classification of the entities with respect to the geometric model entity that the mesh entity is on [30] is used throughout in this thesis. The nomenclature used to describe the mesh topology and classification to the geometric model are given as follows:

Ω_v	Domain associated with the model v , $v = G,\ M$ where G signifies the geometric model and M signifies the mesh model.
$\partial\Omega_v$	Boundary entities of Ω_v.
$\overline{\Omega}_v$	Closure of the domain associated with the model v, $v = G,\ M$ given by $(\Omega_v \cup \partial\Omega_v)$.
M_i^d	*ith* mesh topological entity of dimension d in the mesh model M. $d = 0,\ 1,\ 2,\ 3$ represent mesh vertex, edge, face and region respectively.
∂M_i^d	Boundary of mesh topological entity M_i^d.
$\overline{M}_i^d$	Closure of mesh topological entity defined as $(M_i^d \cup \partial M_i^d)$.
G_i^d	Geometric model topological entity i of dimension d.
Ω_i^e	Domain of a finite element associate with a mesh topological entity M_i^d.
$\overline{\Omega}_i^e$	Closure of a finite element.
$\sqsubset$	Classification symbol used to indicate the association of one or more entities from the mesh model M with the geometric model G.
$\{M^d\}$	Unordered group of mesh topological entities of dimension d.
$\{M_i^{d_i}\{M^{d_j}\}\}$	First order adjacency sets of individual mesh entity $M_i^{d_i}$ defined as the set of mesh entities of dimension d_j adjacent to mesh entity $M_i^{d_i}$.

A mesh model is represented by the basic $0-$ to $d-$dimensional topological entities [30]. In the three dimensional model ($d = 3$) these topological entities are,

$$T_M = \{\{M\{M^0\}\}, \{M\{M^1\}\}, \{M\{M^2\}\}, \{M\{M^3\}\}\}, \tag{2.13}$$

where $\{M\{M^d\}\}$, $d = 0,\ 1,\ 2,\ 3$ are the set of vertices, edges, faces and regions defining the primary topological elements of the mesh model domain.

There are two important concepts in the topology-based mesh data structure for the use of the p-version method. First, mesh classification against the geometric domain $m_i^{d_i} \sqsubset G_j^{d_j}$ that defines the unique association of a mesh entity of dimension d_i, $M_i^{d_i}$ to a geometric model entity of dimension d_j, $G_j^{d_j}$ where $d_i \leq d_j$ [30]. Multiple

$M_i^{d_i}$ can be classified on a $G_j^{d_j}$. Mesh entities are always classified with respect to the lowest order geometric model entity possible. The mesh classification on the geometric model plays a crucial role in the automated mesh generation and adaptation, analysis attribute transfer from the geometric model to the mesh, and to obtain geometric shape information needed for use in defining properly curved mesh entities. Second, adjacency describes how topological entities connect to each other. The most important set of relations are the first order adjacency relations that describe, for a given entity $M_k^{d_i}$, all of the entities, M^{d_j}, $(d_i \neq d_j)$ which are either on the closure of the entity $(d_i > d_j)$, or which it is on the closure of $(d_i < d_j)$ [30]. These sets of relations support variety operations critical to the effective application of the p-version method, including the simple assignment of the desired polynomial order with respect to each topological entity in the mesh and the topological hierarchic shape function construction.

2.2.2 Element-Level Shape Functions

Let N_a denote the high order shape functions defined on the standard reference element Ω_{st} defined in the $\boldsymbol{\xi}$-space such that the finite element approximation u^h over element Ω^e is,

$$u^h(\boldsymbol{\xi}) = \sum_{a=1}^{n_{shp}} N_a(\boldsymbol{\xi}) d_a^e, \tag{2.14}$$

where d_a^e are the unknown coefficients. n_{shp} is the number of shape functions of the element Ω^e that depends on the polynomial order of the shape functions associated with the mesh entities on the closure mesh entity of $\overline{\Omega}^e$. When applying the first-order adjacency, the closure of a finite element, $\bar{\Omega}^e$, of dimension d_e, can be specified and evaluated as,

$$\overline{\Omega}^e = \left\{M_e^{d_e}, \partial M_e^{d_e}\right\} = \left\{M_e^{d_e}, \{M_e^{d_e}\{M_j^{d_e-1}\}\}, ..., \{M_e^{d_e}\{M_j^0\}\}.\right\} \tag{2.15}$$

The selection of shape functions for the use of p-version method has received extensively research in [40, 115, 117, 118, 124] and attention was focused on defining hierarchic shape functions based on Legendre polynomials. This thesis applies the topology based hierarchic shape functions [115] that can be organized into vertex, edge, face and region shape functions uniquely determined by the mesh topology and the specified polynomial order on those mesh entities in Eq. 2.15.

The basic idea of the topology based hierarchic shape functions [115] is to decompose the shape function N associated with mesh entity $M_j^{d_j} \in \overline{\Omega}_e$ in the form,

$$N = \psi(M_j^{d_j}, M_e^{d_e})\phi(M_j^{d_j}) \quad d_j \leq d_e, \quad d_j,\ d_e = 0, 1, 2, 3, \tag{2.16}$$

where,

- $\psi(M_j^{d_j}, M_e^{d_e})$ is the blending function defined over $M_e^{d_e}$ specific to $M_j^{d_j}$. The blending function is written in the parametric coordinate system $\boldsymbol{\xi}$ of the element $M_e^{d_e}$ and independent of the polynomial order of the shape function N.
- $\phi(M_j^{d_j})$ is the entity function written in the local parametric coordinate system $\hat{\boldsymbol{\xi}}$ of the mesh entity $M_j^{d_j}$ that depends on the mesh entity $M_j^{d_j}$ and the polynomial order of the shape function N, and is independent of the element $M_e^{d_e}$ [115]. Thus, the form of the mesh entity function for $M_j^{d_j}$ is the same for all elements (including those of different topologies and/or dimension) connected to $M_j^{d_j}$.

Previous efforts used a uniform polynomial order for the mesh faces and regions in [115]. In Part III of this this thesis this is extended to include independent directional polynomial for the use of adaptive analysis.

2.2.3 Mesh Geometry Mapping Functions

In the p-version finite element method, each element Ω^e is mapped into a standard element Ω_{st} by a mapping function $\boldsymbol{x} = Q^e_{\boldsymbol{x}}(\boldsymbol{\xi})$. Since the boundary of the general 3D curved domains generally consists of piecewise smooth curves and surfaces and most finite elements remains large, the mesh geometry mapping function $Q^e_{\boldsymbol{x}}(\boldsymbol{\xi})$ must consider using accurate mapping techniques to approximate the curved boundaries.

In case of blended mapping [46, 124], $\boldsymbol{x}(\boldsymbol{\xi})$ is exact with respect to the boundaries Γ_G of the geometric model when $\boldsymbol{\xi} \in \Omega^e \sqsubset \Gamma_G$. This approach is accomplished by assigning the mesh entities classified on the curved boundaries of the same geometry as that portion of the model boundary entity [46]. Center idea is to construct the desired mapping for a mesh entity, M_i^d, classified on a model entity, G_j^d as,

$$\boldsymbol{x}(\boldsymbol{\xi}) = Q_{\boldsymbol{x}}^e(\boldsymbol{\zeta}(\boldsymbol{\xi})), \tag{2.17}$$

where $\boldsymbol{\zeta}$ represents the local parametric coordinates associated with the model entity G_j^d. The construction of $\boldsymbol{\zeta}(\boldsymbol{\xi})$ depends on the classification of the mesh entity with respect to the geometric model [46]. However, this is a computationally expensive process.

An alternative is to construct an approximate geometry representation that uses interpolation or geometry approximation techniques to determine appropriate shapes for the mesh entities that are determined and stored together with the mesh entities. In this method, the approximated geometry $\boldsymbol{x}(\boldsymbol{\xi})$ is written as a linear combination of basis functions and control points defined over $\overline{\Omega}^e$ as follows,

$$\boldsymbol{x}(\boldsymbol{\xi}) = Q_{\boldsymbol{x}}^e(\boldsymbol{\xi}) = \sum_{i=1}^{n_q} B_i(\boldsymbol{\xi})\boldsymbol{b}_i^e, \tag{2.18}$$

where B_i is the chosen basis functions and $\boldsymbol{b}_i^e$ are the control points. n_q is the number of basis functions to complete a geometric mapping up to order q. In the isoparametric mapping used for lower order h-version finite element method, $B_i(\boldsymbol{\xi})$ is generally chosen the same as the shape function $N_a(\boldsymbol{\xi})$. For the p-version method developed in this thesis, $B_i(\boldsymbol{\xi})$ is chosen as the Bernstein polynomials on which the Bezier curves, surfaces and regions are built.

A nth order Bernstein polynomial [58] is defined explicitly by,

$$B_i^n(t) = \begin{pmatrix} n \\ i \end{pmatrix} t^i(1-t)^{(n-i)}, \qquad 0 \le i \le n, \tag{2.19}$$

where $\begin{pmatrix} n \\ i \end{pmatrix}$ is a binomial coefficient as,

$$\begin{pmatrix} n \\ i \end{pmatrix} = \frac{n!}{(n-i)!i!}. \tag{2.20}$$

The Bernstein have a number of useful properties [58].

- Bernstein polynomials form a partition of unity,

$$\sum_{i=0}^{n} B_i^n(t) = 1, \quad 0 \leq t \leq 1. \tag{2.21}$$

- Bernstein polynomials are symmetry,

$$B_i^n(t) = B_{(n-i)}^n(1-t). \tag{2.22}$$

- Bernstein polynomials are positive,

$$B_i^n(t) \geq 0. \tag{2.23}$$

- Having special values,

$$B_i^n(0) = B_i^n(1) = \begin{cases} 0 & for\ i = 1, 2, ..., n-1 \\ 1 & for\ i = 0\ or\ n \end{cases} \tag{2.24}$$

With the Bernstein polynomial defined in Eq. 2.19, a qth order Bezier mesh edge can be constructed as [58],

$$\boldsymbol{x}^q(\xi) = \sum_{i=0}^{q} B_i^q(\xi) b_i^q, \quad 0 \leq \xi \leq 1, \tag{2.25}$$

where b_i^q are the control points used to define the shape of the Bezier curve.

With Eq. 2.25, a quadrilateral Bezier mesh face with polynomial order (q_1, q_2) for its coordinates $(\xi_1,\ \xi_2)$ can be defined as the tensor product of a q_1th order Bzier curve in ξ_1 direction and a q_2th order Bezier curve in ξ_2 directions as [58],

$$\boldsymbol{x}^{(q_1,q_2)}(\xi_1,\xi_2) = \sum_{i=0}^{q_1}\sum_{j=0}^{q_2} B_i^{q_1}(\xi_1)B_j^{q_2}(\xi_2)b_{ij}^{(q_1,q_2)}, \quad 0 \le \xi_1,\xi_2 \le 1. \tag{2.26}$$

$b_{ij}^{(q_1,q_2)}$ are the control points.

For a qth order complete Bezier triangle face defined in the barycentric coordinates $\xi_1+\xi_2+\xi_3=1$, the formulation is [58],

$$\boldsymbol{x}^{q}(\xi_1,\xi_2,\xi_3) = \sum_{|\boldsymbol{i}|=q} B_{|\boldsymbol{i}|}^{q}(\xi_1,\xi_2,\xi_3)b_{|\boldsymbol{i}|}^{q}. \tag{2.27}$$

where $b_{|\boldsymbol{i}|}^{q}$ are the control points net. $|\boldsymbol{i}| = i+j+k$ and $B_{|\boldsymbol{i}|}^{q}$ are defined by,

$$B_{|\boldsymbol{i}|}^{q}(\xi_1,\xi_2,\xi_3) = \begin{pmatrix} q \\ |\boldsymbol{i}| \end{pmatrix} \xi_1^i\xi_2^j\xi_3^k = \frac{q!}{i!j!k!}\xi_1^i\xi_2^j\xi_3^k, \quad |\boldsymbol{i}| = q. \tag{2.28}$$

Follow the same algorithm, a hexahedral Bezier region with polynomial order (q_1,q_2,q_3) is a tensor product of three Bezier curves at the independent coordinates $\xi_1,\ \xi_2,\ \xi_3$ directions and is represented as,

$$\boldsymbol{x}^{(q_1,q_2,q_3)}(\xi_1,\xi_2,\xi_3) = \sum_{i=0}^{q_1}\sum_{j=0}^{q_2}\sum_{k=0}^{q_3} B_i^{q_1}(\xi_1)B_j^{q_2}(\xi_2)B_k^{q_3}(\xi_3)b_{ijk}^{(q_1,q_2,q_3)}, \quad 0 \le \xi_1,\xi_2,\xi_3 \le 1. \tag{2.29}$$

A Bezier prismatic region is the tensor product of a q_1th order triangle face along barycentric coordinates $(\xi_1,\ \xi_2,\ \xi_3)$ and a q_2th order Bezier curve along coordinate ξ_4 direction and is of the form,

$$\boldsymbol{x}^{(q_1,q_2)}(\xi_1,\xi_2,\xi_3,\xi_4) = \sum_{|\boldsymbol{i}|=q_1}\sum_{l=0}^{q_2} B_{|\boldsymbol{i}|}^{q_1}(\xi_1,\xi_2,\xi_3)B_l^{q_2}(\xi_4)b_{|\boldsymbol{i}|l}^{(q_1,q_2)}. \tag{2.30}$$

As for a qth order Bezier tetrahedral region defined in the volume coordinate ($\xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$), it can be represented as,

$$\boldsymbol{x}^q(\xi_1, \xi_2, \xi_3, \xi_4) = \sum_{|\boldsymbol{i}|=q} B^q_{|\boldsymbol{i}|}(\xi_1, \xi_2, \xi_3, \xi_4) b^q_{|\boldsymbol{i}|}, \tag{2.31}$$

where $|\boldsymbol{i}| = i + j + k + l$ and $B^q_{|\boldsymbol{i}|}$ are defined as,

$$B^q_{|\boldsymbol{i}|}(\xi_1, \xi_2, \xi_3, \xi_4) = \left(\begin{array}{c} q \\ |\boldsymbol{i}| \end{array} \right) \xi_1^i \xi_2^j \xi_3^k \xi_4^l = \frac{q!}{i!j!k!l!} \xi_1^i \xi_2^j \xi_3^k \xi_4^l, \quad |\boldsymbol{i}| = q. \tag{2.32}$$

The Bezier mesh geometry shape posses a number of advantageous properties include:

- The Convex Hull Property - A Bezier curve, surface, or volume is contained in the convex hull formed by its control points.
- The Variation Diminishing Property - An infinite plane can not intersect a Bezier curve more times than it intersects control polygon which allows more efficient intersection calculations.
- All derivatives and products of Bezier functions are easily computed Bezier functions.
- Computationally efficient algorithms for degree elevation and subdivision are available which can be used to refine the shape's convex hull as well as adaptively refine the mesh's shape.

The determination of the control points $\boldsymbol{b}^e_i$ for a Bezier mesh entity shape requires the solution of n_q dimension linear system of equations [45] as follows,

$$\begin{bmatrix} m_{11} & m_{12} & ... & m_{1n_q} \\ m_{21} & m_{22} & ... & m_{2n_q} \\ ... & ... & ... & ... \\ m_{n_q1} & m_{n_q2} & ... & m_{n_qn_q} \end{bmatrix} \begin{Bmatrix} \boldsymbol{b}^e_1 \\ \boldsymbol{b}^e_2 \\ ... \\ \boldsymbol{b}^e_{n_q} \end{Bmatrix} = \begin{Bmatrix} c_1 \\ c_2 \\ ... \\ c_{n_q} \end{Bmatrix}, \tag{2.33}$$

where

$$m_{ij} = B_j(\xi^{(i)}), \tag{2.34}$$

$$c_i = \boldsymbol{x}(\xi^{(i)}), \tag{2.35}$$

$\{\xi^{(i)}\},\ i = 1...n_q$ defines the set of parametric coordinates at which the approximate geometry shape is evaluated. In the case of mesh edges classified on the curved model boundaries, their control points are determined based on Bezier curve and surface interpolation methods resulting from evaluating the model geometry at a set of discrete locations. Common methods usually assume the interpolation points are uniformly distributed in the parametric space of the mesh entity. However, this approach can lead to poor geometric approximations. An alternative method that improves the geometric approximation for a given order uses a chord length method [59]. In cases where there are large changes in the curvature in the portion model entity being approximated by the mesh entity the use of a curvature-based procedure for selecting the interpolating points is appropriate [85]. Specific attention needs to be paid to model entities that contain either parametric degeneracies and/or periodicity.

2.2.4 Computation of Element-Level Stiffness Matrix and Load Vector

With the finite element mesh discretization and high order shape functions the Garlerkin form in Eq. 2.12 is computed at the element level and subsequently assembled to produce the equivalent global matrix forms [71] as,

$$\boldsymbol{K}\boldsymbol{d} = \boldsymbol{F}, \tag{2.36}$$

where $\boldsymbol{K}$ is the global stiffness matrix and $\boldsymbol{F}$ is the global load vector. $\boldsymbol{d}$ is the unknown coefficients vector. $\boldsymbol{K}$ and $\boldsymbol{F}$ are assembled from the local stiffness matrix $\boldsymbol{k}^e$ and load vector $\boldsymbol{f}^e$ of each element [71].

In case of the Poisson equation, the local stiffness matrix $\boldsymbol{k}^e$ and load vector $\boldsymbol{f}^e$ of each element are computed as [71],

$$\boldsymbol{k}^e_{ab} = \int_{\Omega^e} B_a^T B_b \; d\Omega^e, \tag{2.37}$$

$$\boldsymbol{f}^e_a = \int_{\Omega_e} N_a f \; d\Omega_e + \int_{\Gamma_h^e} N_a h \; d\mathit{\Gamma} - \sum_{b=\mathit{1}}^{n_{el}} \boldsymbol{k}^e_{ab} g^e_b, \tag{2.38}$$

with $B_a = \nabla N_a$.

For the linear elasticity problem [71],

$$\boldsymbol{k}^e_{pq} = e_i^T \boldsymbol{k}^e_{ab} e_j, \quad \boldsymbol{k}^e_{ab} = \int_{\Omega^e} B_a^T D B_b \; d\Omega^e, \tag{2.39}$$

$$\boldsymbol{f}^e_p = \int_{\Omega^e} N_a f_i \; d\Omega^e + \int_{\Gamma^e_h} N_a h_i \; d\mathit{\Gamma} - \sum_{q=1}^{n_{el}} \boldsymbol{k}^e_{pq} g^e_q, \tag{2.40}$$

the indices $p,\ q$ are defined as follows [71],

$$p = n_{ed}(a-1) + i, \qquad q = n_{ed}(b-1) + j, \tag{2.41}$$

n_{ed} is the degrees of freedom number. e_i denotes the ith Euclidean basis vector for the real space $\mathbb{R}^{n_{sd}}$. D is a symmetric, positive definite matrix of material properties and is defined in terms of c_{ijkl}. B_a is of the form as [71],

$$B_a = \begin{bmatrix} N_{a,x} & 0 \\ 0 & N_{a,y} \\ N_{a,y} & N_{a,x} \end{bmatrix}, \tag{2.42}$$

for $n_{sd} = 2$ and of the form as,

$$B_a = \begin{bmatrix} N_{a,x} & 0 & 0 \\ 0 & N_{a,y} & 0 \\ 0 & 0 & N_{a,z} \\ 0 & N_{a,z} & N_{a,x} \\ N_{a,z} & 0 & N_{a,y} \\ N_{a,y} & N_{a,x} & 0 \end{bmatrix}, \tag{2.43}$$

for $n_{sd} = 3$ and $1 \leq a, b \leq n_{shp}$.

The local stiffness matrix $\boldsymbol{k}^e$ and load vector $\boldsymbol{f}^e$ are computed at the standard element Ω_{st} and $d\Omega^e$ can be computed as,

$$d\Omega^e = \; d\boldsymbol{x} = \; J \; d\boldsymbol{\xi}, \tag{2.44}$$

where J is the determinant of Jacobian of the mapping matrix $\left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right]$. In case $n_{sd} = 3$, $\boldsymbol{x}$ and $\boldsymbol{\xi}$ are defined as $\boldsymbol{x}(x, y, z)$ and $\boldsymbol{\xi}(\xi_1, \xi_2, \xi_3)$, matrix $\left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right]$ is computed as,

$$\left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right] = \begin{bmatrix} x_{,\xi_1} & x_{,\xi_2} & x_{,\xi_3} \\ y_{,\xi_1} & y_{,\xi_2} & y_{,\xi_3} \\ z_{,\xi_1} & z_{,\xi_2} & z_{,\xi_3} \end{bmatrix}. \tag{2.45}$$

Based on the Eq. 2.18, the components are,

$$x_{,\xi_1} = \sum_{i=1}^{n_q} b_i^x \frac{\partial B_i}{\partial \xi_1},\ x_{,\xi_2} = \sum_{i=1}^{n_q} b_i^x \frac{\partial B_i}{\partial \xi_2},\ x_{,\xi_3} = \sum_{i=1}^{n_q} b_i^x \frac{\partial B_i}{\partial \xi_3}, \tag{2.46}$$

$$y_{,\xi_1} = \sum_{i=1}^{n_q} b_i^y \frac{\partial B_i}{\partial \xi_1},\ y_{,\xi_2} = \sum_{i=1}^{n_q} b_i^y \frac{\partial B_i}{\partial \xi_2},\ y_{,\xi_3} = \sum_{i=1}^{n_q} b_i^y \frac{\partial B_i}{\partial \xi_3}, \tag{2.47}$$

$$z_{,\xi_1} = \sum_{i=1}^{n_q} b_i^z \frac{\partial B_i}{\partial \xi_1},\ z_{,\xi_2} = \sum_{i=1}^{n_q} b_i^z \frac{\partial B_i}{\partial \xi_2},\ z_{,\xi_3} = \sum_{i=1}^{n_q} b_i^z \frac{\partial B_i}{\partial \xi_3}. \tag{2.48}$$

J is computed as,

$$J = x_{,\xi_1}(y_{,\xi_2} z_{,\xi_3} - y_{,\xi_3} z_{,\xi_2}) + x_{,\xi_2}(y_{,\xi_3} z_{,\xi_1} - y_{,\xi_1} z_{,\xi_3}) + x_{,\xi_2}(y_{,\xi_1} z_{,\xi_2} - y_{,\xi_2} z_{,\xi_1}). \tag{2.49}$$

The derivative of N_a with respect to $\boldsymbol{x}$ can be computed as follows,

$$N_{a,\boldsymbol{x}} = \frac{\partial N_a}{\partial \boldsymbol{x}} = \frac{\partial N_a}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}, \tag{2.50}$$

where $\left[\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right] = \left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right]^{-1}$.

2.3 Effective Numerical Integration of p-Version Finite Element Method

The stiffness matrix and load vector in Eq.2.37 - 2.40 can not be integrated exactly and are generally computed using numerical integration. Let $\{R_p\}$ be the selected numerical integration scheme defined by a set of integration points $\{\boldsymbol{\xi}^l\}$ and weights $\{w^l\}$. In p-version method, the precision of the $\{R_p\}$ must also increase as the polynomial order p increases. The influence of the numerical quadrature error on the rate of convergence of p-version method has been investigated in detail in [27, 76]. It was stated that the integration scheme $\{R_p\}$ on the standard reference element Ω_{st} must satisfy two assumptions [27, 76]:

1. The weights are positive and the integration points lie within Ω_{st}.
2. $\{R_p\}$ is exact for all of the functions defined in Ω_{st} with polynomial order $m \geq 2p$.

to ensure that the numerical quadrature error will decay as fast as the discretization error.

In case of tensor product type elements such as quadrilateral and hexahedral element, $p_i + 1$ points Gauss-Legendre and p_i points Gauss-Lobatto in each direction are minimal required when polynomial order of p_i is used along the direction [27]. Therefore the total number of integration points n_g are

$$n_g = \underbrace{(p_i + 1) \times \ \ldots \ \times (p_j + 1)}_{n_{sd} \ \text{times}} \tag{2.51}$$

for higher dimension elements in case of Gauss-Legendre scheme. As for other topological elements such as triangle, tetrahedral, prism and pyramid that do not have orthogonal local coordinates, the integration points can be derived by applying the coordinates mapping strategy presented in [46] and will be described in Part III of this thesis.

The matrix equation in Eq. 2.36 is solved to obtain the unknown coefficients vector $\boldsymbol{d}$ to construct the finite element solution u^h.

2.4 The Convergence Rate of the p-Version Method

The accuracy of u^h to u is measured with the finite element approximation error $e = u^h - u$ in norms of interest. Let $||\ .\ ||$ be the energy norm defined as,

$$||u|| = a(u,\ u)^{\frac{1}{2}}. \tag{2.52}$$

The error measured in energy norm $||e||$ is the overall quality measure of the finite element solution and is of the most interest in this thesis [71, 124]. A priori estimates of the convergence rate of p-version method in energy norm based on the character of the exact solution and discretized finite element meshes has been investigated in [12, 14, 17, 26, 48, 49, 66, 67, 124].

The exact solution u can be grouped into three categories over the closure of the solution domain $\overline{\Omega}$ ($\overline{\Omega} = \Omega \cup \partial\Omega$) [124]:

(*a*) u is analytic over $\overline{\Omega}$. The function u is analytic in a point if it can be expanded into a Taylor series about that point on the entire solution domain $\overline{\Omega}$.

(*b*) u is analytic over $\overline{\Omega}$ except a finite number of singular points or lines. An example of problems in this category is the crack tip (2D) or edge (3D) of an elastic domain.

(*c*) u is neither in category (***a***) nor (***b***).

The linear elasticity and Poisson equation problems considered in this thesis belong in either category (***a***) or (***b***).

Knowing the classification of a problem, the rate of convergence measured in energy norm can be compared for different finite element methods. It has been shown that in the case of h-version method with quasiuniform meshes, the error measured in energy norm is [17, 124],

$$||e|| \leq CN^{-\beta}, \tag{2.53}$$

where $\beta = \frac{1}{2}min(\lambda,\ p)$ and λ is smoothness parameter of the exact solution. C is positive constant and N is the number of degree of freedom. Eq. 2.53 shows the rate of convergence using h-version method is $\frac{1}{2}p$ when the solution u is smooth (category ***a***) and is $\frac{1}{2}\lambda$ when the solution has singularities over the domain (category ***b***).

For the p-version method, in the case the exact solution is smooth (category ***a***) without any singularity inside the model domain or on the boundary of the model domain, the rate of convergence is exponential [12, 124],

$$||e|| \leq Ce^{-\gamma N^{\theta}}, \tag{2.54}$$

where C, γ, and θ are positive constants, $\theta \geq 1/2$ [124].

When there are singularities inside the domain or on the boundary of the domain (u belongs to category ***b***) and quasiuniform meshes are applied, the rate of convergence for the p-version method is of the same form in Eq. 2.53. The constant β is determined as follows [12, 124]:

- When the singularities are inside the domain but not on the boundary points or edges of the domain, $\beta = \frac{1}{2}\lambda$. The rate of convergence of p-version method is exactly the same as that of h-version method.
- When the singularities are on the boundary points or edges of the element boundaries, $\beta = \lambda$. It shows that the p-version method can achieve twice the rate of convergence as the h-version method. However, the convergence rate for both of the methods is algebraic.

In case of nonquasiuniform meshes with geometric gradation q towards the solution singularity (so called hp-version method), the rate of convergence for p-version method is exponential and can be written as [17],

$$||e|| \leq Ce^{-\sqrt{(\lambda-1/2)N}\sqrt{2\log q \log r}}, \tag{2.55}$$

with

$$r = \frac{1-\sqrt{q}}{1+\sqrt{q}}. \tag{2.56}$$

The optimal rate of convergence in Eq. 2.56 is achieved if the quantity,

$$\log q \log \frac{1-\sqrt{q}}{1+\sqrt{q}}, \tag{2.57}$$

reaches its maximum.

The optimal q exists at [66],

$$q_0 = (\sqrt{2}-1)^2 \approx 0.1715. \tag{2.58}$$

In practical implementation of p-version method, the geometric gradation is usually chosen as 0.15 to overrefine the mesh [17, 122, 124].

Although the above conclusions are drawn based on the studies of one or two dimensional linear elliptical model problem, the analogous results hold in three dimension [17, 124].

CHAPTER 3

The Implementation Requirements of the p-Version Method on General 3D Curved Domains

This chapter discusses the influence of geometry on the character of the exact solution for second order linear partial differential equations on general 3D curved domains. The mesh requirements for p-version method to achieve its high rate of convergence are then given. An overall algorithm to generate near optimal p-version meshes is presented.

3.1 Geometric Mesh Gradation for Solution Singularities

Eq. 2.55 shows that the mesh must be graded towards the solution singularities with an optimal gradation 0.15. The solution of second order linear partial differential equations in three dimensions in the vicinity of any singular point can be decomposed into three different forms, depending whether the singular point is in the neighborhood of an edge, a vertex or an intersection of the edge and the vertex. Let $\{G^{d_j}\}$, $d_j = 0, 1$ denote the model vertices and edges of a general 3D curved model. In the vicinity of a model edge G^1_j with local cylindrical coordinates (r, x_3, ϕ) along the model edge, the exact solution u can be written as [17],

$$u = \sum_{j=1}^{J} \sum_{s=0}^{S} \sum_{t=0}^{T} c_{jst}(x_3)\psi_{jst}(\phi) r^{(\alpha_j + t)} ln^s(r) + u_0, \tag{3.1}$$

where $j,\ s,\ t$ are integers. $c_{jst} \in C^{\infty}(G^1_j)$ and ψ_{jst} are analytic in ϕ. u_0 is smooth function. J, S, T are chosen such that u_0 is smoother than the smooth terms c_{jst} and ψ_{jst} in the sum. $\alpha_j = j\pi/\omega$ and ω is the interior dihedral angle of the edge G^1_j.

Eq. 3.1 shows that the solution u behaves as r^{α_j} type functions and the smoothness is characterized by the value α_j. In case the interior dihedral angle $\omega \geq \pi$, Eq. 3.1 indicates that the solution is smooth along the direction of the model edge (x_3 direction) and singular in the perpendicular direction (r direction). Thus, the mesh must be controlled with cylindrically geometric gradation layout in the perpendicular direction of a singular model edge and coarse mesh along the edge direction shown in Figure 3.1.

For a model vertex G_j^0 with local spherical coordinates (ρ, θ, ϕ), the solution u can be written as [17],

$$u = \sum_{j=1}^{J}\sum_{s=0}^{S}\sum_{t=0}^{T}(\sum_{i=0}^{I}\tilde{c}_{jsti}\rho^{-1/2+\beta_i}+f_{jst}(\rho))\tilde{\psi}_{jst}(\phi)\theta^{\tilde{\alpha}_j+t}ln^s(\theta)+\sum_{i=0}^{I}d_i\rho^{-1/2+\beta_i}\tilde{g}_i(\theta,\phi)+\tilde{u}_0, \tag{3.2}$$

where $\tilde{c}_{jsti}$, f_{jst}, $\tilde{\psi}_{jst}$, $\tilde{g}_i$ and $\tilde{u}_0$ are smooth functions. J, S, T, I are chosen following the same pattern in Eq. 3.1. β_i is determined by the Laplace-Bertrami eigenvalues λ_i [17, 49] as,

$$\beta_i = \sqrt{1/4 + \lambda_i}. \tag{3.3}$$

It was seen from Eq. 3.2 that the exact solution u behaves as $\rho^{-1/2+\beta_i}$ [17, 49] that requires the mesh spherically graded towards to the model vertex. The determination of λ_i usually requires numerical approximations that makes the prediction of the vertices singularities apriori more difficult. This thesis focus on edge singularities for 3D curved domains.

3.2 Mesh Requirements for Thin Sections of 3D Curved Models

In addition to the singular model edges and vertices that affect the character of the exact solution, a complicated 3D curved model may have some additional structures such as thin sections, stiffeners, fillets, cutouts, bosses etc. that affect the character of the solution u and requires specific mesh construction in the p-version method [79, 100]. This thesis considers the influence of thin section structures in 3D linear elasticity problem. For the portions of domain where the thickness dimension is no greater than one-tenth of the smallest in-plane dimension and is subjected to loads that cause bending deformation in addition to stretching, these regions can be regarded as thin section structures. Such sections are often modeled by applying dimensional reduction to use 2D shell theories. The widely used conventional plate

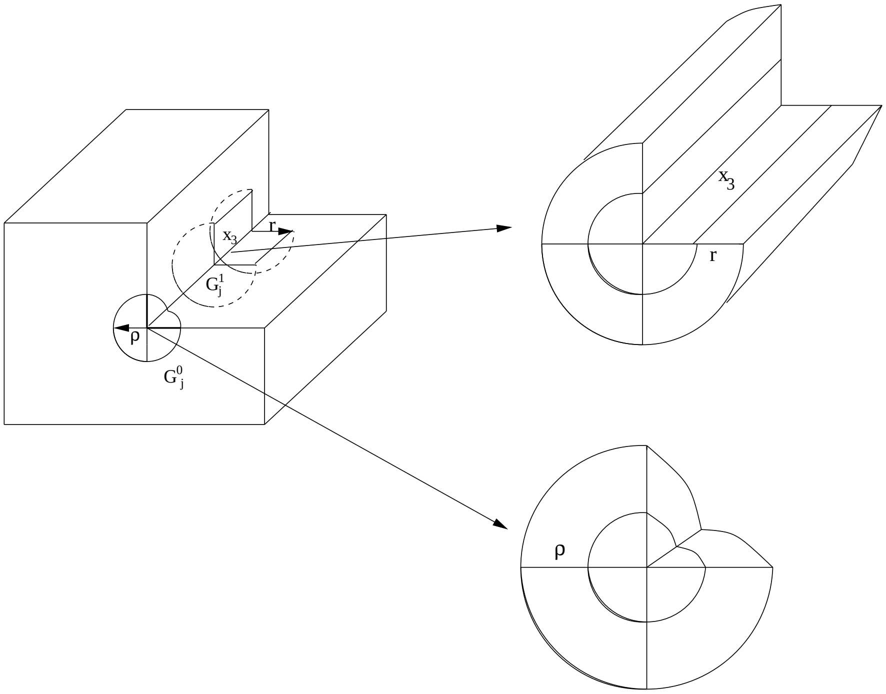


Figure 3.1: The Required Meshes for Singular Model Edges and Vertices

and shell theories are Kirchhoff and Reissner theories. In the Kirchhoff's theory, the displacement components u_x, u_y, u_z are represented as [105, 124],

$$\begin{aligned} u_x &= u_{x0}(x,y) - z\frac{\partial u_{z0}}{\partial x}, \\ u_y &= u_{y0}(x,y) - z\frac{\partial u_{z0}}{\partial y}, \\ u_z &= u_{z0}(x,y), \end{aligned} \tag{3.5}$$

where functions u_{x0}, u_{y0} represent the components of the in-plane displacement components in the x and y directions respectively. u_{z0} represents the traverse displacement component. It is assumed that $\varepsilon_z = \frac{\partial u_z}{\partial z} = 0$ such that the shear strains γ_{xz} and γ_{yz} are zero.

$$\gamma_{xz} = \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} = -\frac{\partial u_{z0}}{\partial x} + \frac{\partial u_{z0}}{\partial x} = 0. \tag{3.7}$$

For the Reissner-Mindline plate model, the displacement components can be written as [103, 105, 124],

$$\begin{aligned} u_x &= \phi_x z, \\ u_y &= \phi_y z, \\ u_z &= \beta. \end{aligned} \tag{3.8}$$

The physical interpretation of ϕ_x and ϕ_y are the rotation about x- and y- axis respectively and β is the deflection in z- direction. ϕ_x, ϕ_y and β are the primary variables in the plate problem $u = [\phi_x,\ \phi_y,\ \beta]^T$. With the above assumption, the bilinear and linear operations of weak form for the three dimensional model problem are modified as follows [103, 124],

$$a(w,\ u) = \frac{d^3}{12} \int_\Omega \varepsilon_R^T D_R \varepsilon_R dxdy + d \int_\Omega \varepsilon_D^T D_D \varepsilon_D dxdy, \tag{3.9}$$

$$(w,\ q) = \int_\Omega \beta q dxdy, \tag{3.10}$$

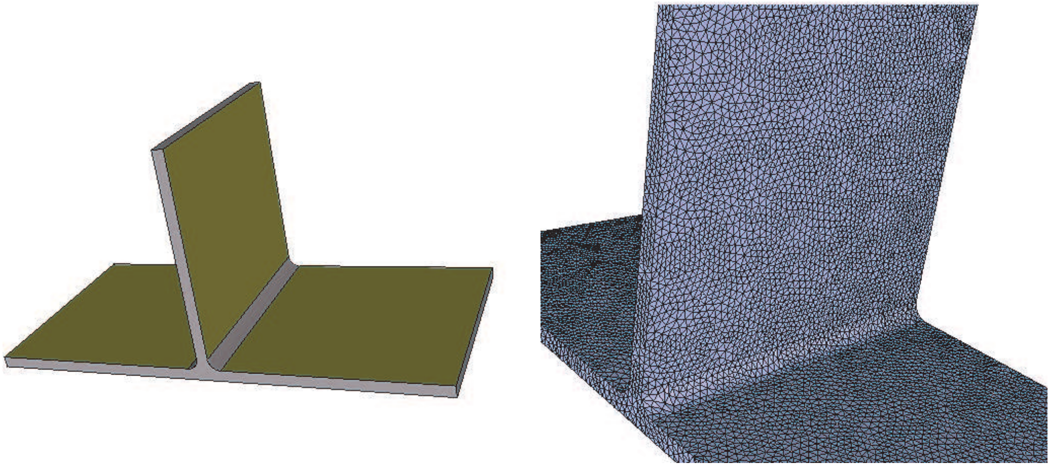
$$(w,\ h) = \oint_\Gamma (M_n \phi_n + M_{nt} \phi_t - Q_n \beta) ds, \tag{3.11}$$

where q is the imposed load in z-direction. $\varepsilon_R = \{\partial\phi_x/\partial x,\ \partial\phi_y/\partial y,\ (\partial\phi_x/\partial y + \partial\phi_y/\partial x)\}$ and $\varepsilon_D = \{\phi_x + \partial\beta/\partial x,\ \phi_y + \partial\beta/\partial y\}$. $M_n, M_{nt}, Q_n, \phi_n, \phi_t$ are the moments, shear forces deflection and rotation in the normal and tangential directions [103, 105, 124]. In case that isotropic material is used with Young's module E and Possion ratio v, the material property matrix D_D and D_R are of the form,

$$D_R = \frac{E}{(1 - v^3)} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & (1-v)/2 \end{bmatrix}, \quad D_D = \frac{E}{2(1+v)} \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}, \tag{3.12}$$

k is the shear correction factor used in the Reissner-Mindline plate model [103, 105, 124].

In many practical modeling problems the solution in the interior regions can be approximated well with the 2D shell models. However, if the local quantities such as stresses at the intersection regions are of interest, the 2D shell models can not yield reliable results [1, 124]. Since the stresses are truly three dimensional for these intersection regions and the assumptions in Eq. 3.5 and 3.8 for the 2D shell models do not hold. As an example, Figure 3.2(a) shows a model that three thin sections intersect with each other and the joint regions must be modeled with fully three-dimensional models. Special finite elements are needed to deal with the transition between 2D and 3D elements that introduce additional computational difficulties [1, 124].



(a) Model with Three Thin Sections Intersected

(b) Tetrahedral Mesh for h-version Method with 225440 Regions

Figure 3.2: Model and Tetrahedral Meshes for h-version Method

An alternative is to model the whole domain in three dimension. In case the h-version method is applied to solve such the problem, an excessive number of elements is needed (shown in Figure 3.2(b)) to mesh the domain because accurate solution can only be obtained if the ratio aspect of an element is close to one for the low order elements that apply isoparametric mapping. This makes fully three dimensional h-version approximation inefficient.

In the case of the p-version method that uses reasonably coarse meshes to discretize the domain, it is important to treat the in-plane directions and thickness direction differently. The basic theory goes back to the p-version hierarchic plate and shell models developed by Szabo etc. [123] where the displacement fields are typically approximated by expressions of the form,

$$\begin{aligned} u_x &= \sum_{i=0}^{n} f_i(z) u_{xi}(x, y), \\ u_y &= \sum_{i=0}^{n} f_i(z) u_{yi}(x, y), \\ u_z &= \sum_{i=0}^{m} f_i(z) u_{zi}(x, y), \end{aligned} \tag{3.14}$$

where $f_i(z)$ represents some approximations to the traverse variation of the displacement components. Typically, $f_i(z) = z^i$ [104, 123]. Each hierarchic lower order model is embedded in all higher order models that covergence to the exact three dimensional problems with $n \to \infty$ and $m \to \infty$ [123].

Eq. 3.14 indicates that the displacement components can be decompsed as the product of function $f_i(z)$ that only depends on the direction z and functions $u_{xi}(x, y)$, $u_{yi}(x, y)$ and $u_{zi}(x, y)$ that depends on the x direction and y direction and independent of z. If prismatic elements are used to discretize the thin section structures shown in Figure 3.3, the triangle faces of the prismatic elements are aligned along the in-plane directions such that the area coordinates ξ_1, ξ_2, and ξ_3 ($\xi_1 + \xi_2 + \xi_3 = 1$) corresponds to the in-plane directions (x, y) and the independent coordinate ξ_4 corresponds to the thickness direction z. Therefore it is possible to choose the polynomial order in the thickness ξ_4 direction different from the in-plane directions [1, 57, 104, 123].

Considering a polynomial order vector $\boldsymbol{p} = \{p_1, p_2, p_3\}$ for the three independent coordinates (ξ_1, ξ_2, ξ_4) of a prismatic element, it is critical to have $p_1 > p_3$, $p_2 > p_3$ for efficient discretization of the thin sections [1, 57]. In addition, in the structural problem, the three primary variables (u_x, u_y, u_z) can choose different polynomial order vectors as $\boldsymbol{p}_x$, $\boldsymbol{p}_y$, and $\boldsymbol{p}_z$.

Since the polynomial order vector $\boldsymbol{p}$ defines the desired polynomial order for the local independent coordinates on the prismatic region level, the polynomial order of the closure topological mesh entities of the prismatic region $\overline{\Omega}^e$ must be defined based on $\boldsymbol{p}$ such that correct number of shape functions as in Eq. 2.16 can

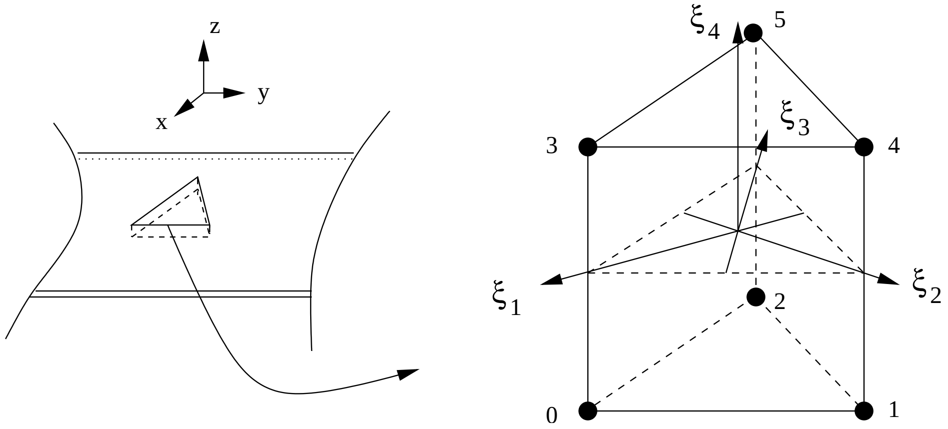


Figure 3.3: Prism Element for Thin Section Structure

be constructed.

Given a polynomial order $\boldsymbol{p} = \{p_1, p_1, p_3\}$ of a prismatic region M_e^3 and the local vertex indices as shown in Figure 3.3, the polynomial orders for its closure topological mesh entities are:

- Mesh edge
    1. The in-plane mesh edges M_{01}^1, M_{12}^1, M_{02}^1, M_{34}^1, M_{45}^1, M_{35}^1 have polynomial order p_1.
    2. The thickness direction mesh edges M_{03}^1, M_{14}^1, M_{25}^1 have polynomial order p_3.
- Mesh face
    1. The in-plane triangle mesh faces M_{012}^2 and M_{345}^2 have polynomial order (p_1, p_1) for the two face independent coordinates.
    2. Assuming the quadrilateral mesh faces M_{0143}^2, M_{1254}^2 M_{0253}^2 have the same orientation such that the second face independent coordinate along the thickness direction, they have polynomial (p_1, p_3).

Based on the above analysis, the desired mesh for the thin section structures would be one layer prismatic elements without long diagonal through the thickness direction.

3.3 Geometry Approximation Representation of the Model Boundary

For the second order linear partial differential equations, the functions related to the geometry appearing in the element stiffness and load vector computation can be represented by the following two terms:

- $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$ in the derivative computation of the high order shape function N_a with respect to $\boldsymbol{x}$ shown in Eq. 2.50.
- J - the determinant of the Jacobian mapping matrix shown in Eq. 2.49.

Since the computation of $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$ requires the inverse mapping $\left[\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right]^{-1}$ exists, the mapping function $Q^e_{\boldsymbol{x}}$ should satisfies the following conditions [71]:

1. One to one which indicates two different points in $\boldsymbol{\xi}$-space of $\overline{\Omega}_{st}$ do not get mapped into the same point in $\overline{\Omega}^e$.
2. Onto which indicates each point in $\overline{\Omega}^e$ is the image of a point in $\overline{\Omega}_{st}$ under the mapping $Q^e_{\boldsymbol{x}}$.
3. $J(\boldsymbol{\xi}) > 0$ for all $\boldsymbol{\xi} \in \overline{\Omega}_{st}(\boldsymbol{\xi})$.

Considering that the stiffness matrix and load vector are computed at a set of integration points $\{\boldsymbol{\xi}^l\}$ determined by the selected numerical integration scheme R_p for a specified polynomial order p in the p-version method (See Section 2.3), it is essential for every mesh entity satisfies the condition 3 as $J(\boldsymbol{\xi}^l) > 0$.

One important application of the Bezier geometric shape (See Section 2.2.3) in the p-version mesh generation is the curved element shape validity test. In the case of a Bezier tetrahedron volume, the determinant of Jacobian J is written as,

$$J = (\frac{\partial \boldsymbol{x}}{\partial \xi_1} \times \frac{\partial \boldsymbol{x}}{\partial \xi_2}) \bullet \frac{\partial \boldsymbol{x}}{\partial \xi_3}, \tag{3.16}$$

where $\frac{\partial \boldsymbol{x}}{\partial \xi_1}$, $\frac{\partial \boldsymbol{x}}{\partial \xi_2}$ and $\frac{\partial \boldsymbol{x}}{\partial \xi_3}$ are the three partial derivatives of $\boldsymbol{x}(\boldsymbol{\xi})$ which are also Bezier functions.

Because the product of Bezier functions are also Bezier functions, the J can be represented as a polynomial in Beizer form which is bounded by its convex hull of control points. If all of the control points of the J are greater than zero then the region's J must be greater than zero everywhere. Comparing to validity checks that test the value of the J at the integration points used in performing the analysis, the approach has the following advantages for simplex mesh entities [85]:

- The element geometric shape is determined only by its own control points. Once the element is valid, the determinant of Jacobian J is greater than zero inside the element independent of the basis functions, polynomial order, or integration rules of the finite element approximation.
- The relation of the J to the control polygon provides insight on how a region found to be invalid due to some portion having negative J can be made valid most effectively.

Error is introduced in the stiffness matrix and load vector when the mesh geometry is an approximation to the actual geometric domain. Although the geometric approximation error can be improved by raising the geometric approximation order q, it increases the computational cost. The correct geometric approximation order q, in term of the polynomial approximation order p, has been investigated in [45] through the basic approximation theory. It has been shown that the geometric approximation error measure in energy norm $||e^{geo}||$ is of the form,

$$||e^{geo}|| \leq Ch^{q+2-m}||u||_{H^r(\Omega)}, \quad r \geq q+1. \tag{3.17}$$

In order to maintain the rate of convergence of p-version method, the geometric approximation error must be as fast as that of the discretization error and is met by [45],

$$q \geq p-1. \tag{3.18}$$

Eq. 3.18 demonstrates that the geometric approximation of the curved geometric model must be represented with correct order to ensure the rate of convergence of the p-version method. This requirement is met by increasing the control points n_q in Eq. 2.18 to complete higher geometric approximation order q.

3.4 Overall Algorithm for Automatic p-Version Mesh Generation

Near optimal meshes for the use of p-version method have been characterized in the above three sections that includes the following aspects:

- Geometric mesh gradation towards solution singularities,
- Controlled geometric approximation error for curved model boundaries,
- Structured elements without long diagonals through for thin sections.

There exist a number of developed approaches that addressed gradation layout in the mesh generation for the use of p-version method. For example, given a starting coarse hexahedral and prism mesh, the edge and vertex refinement strategy automatically replace finite elements connected to the singular model edges and vertices with sets of strongly graded cylinder and spherical meshes [7] to produce the required mesh gradation layout. This procedure is much simpler to implement but has strict requirements on the initial given structured mesh and loses its generalization. Another type of mesh generator that considers mesh layout is the advancing layers method [39, 60, 73, 75, 84, 97, 113] that is primarily developed for viscous flow problems exhibiting strong gradients in certain local directions compared to the other directions. Central to these approaches is the generation of a graded mesh next to surfaces where a boundary layer is needed and filling the rest of the domain by an isotropic mesh generator. However, these procedures focused on dealing with straight-sided linear mesh and must be extended to deal with curved elements.

The satisfaction of geometric approximation needs for the p-version meshes is met through the assignment of appropriate geometric shapes to mesh edges and faces classified on curved model boundaries. In case that the geometric approximation shape of the elements is defined through the Lagrange interpolation polynomials, the mesh vertices and the edge and faces nodes must be placed on the model boundaries. When geometric approximation methods are used, the control points selected to define the geometric shapes of the elements must be properly placed.

One approach to construct p-version meshes starts from linear straight-sided mesh generation developed for the h-version method, followed by assigning high-order geometric approximation shapes to the mesh entities classified on curved model boundaries and has been described in detail in [36, 47, 117]. However, this approach does not consider the mesh gradation layout for the p-version method at singularities.

On the assumption that the entire model is thin, a number of developed procedures [79, 100] that aimed at meshing the thin domains with all hexahedral or prism elements have been developed. Central to these algorithms is the generation of a surface mesh for the medial surface of the curved models that is converted to 3D solid mesh by extrusion in the normal direction of the surface. However, these approaches do not consider singularities and curved mesh entities.

In summary, the existing mesh generation procedures related to the p-version method focus on only one of the three basic mesh requirements. This thesis provides an algorithm that considers all three requirements in an integrated manner. Starting from a general curved solid model, the steps in the automatic p-version mesh generation procedure are:

Step1: Generate a coarse surface mesh.

Step2: Automatically isolate the singular edges and vertices of the model.

Step3: Automatically identify the thin section structures.

Step4: Apply boundary layer mesh generation procedure to account for the isolated singular features.

Step5: Mesh the thin sections with structured prismatic elements.

Step6: Apply general volume mesh generator to fill the remaining domains.

Step7: Curve the singular feature isolation mesh to maintain proper mesh gradation layout

Step8: Curve the remaining mesh entities classified on the curved boundaries.

Step9: Apply local mesh modifications as needed to ensure validity of the mesh.

In this work, the singular model edges are isolated through the checking of the interior dihedral angles of the model edges. The identification of thin sections is achieved based on a set of medial surface points obtained by an octant tracing algorithm over a coarse surface mesh classified on the model. These isolated singular features and thin sections require the mesh to posses a graded layout and without long diagonal through the thickness direction for p-version method. This requirement is fulfilled in step 4 by applying the a boundary layer mesh generation procedure [60] and step 5 by a developed process to match the surface configuration between the two model faces that define a thin section structure.

The volume mesh generated at step 6 is straight-sided and needs to be curved. Bezier mesh geometric shapes discussed in Section 2.2.3 are used as the geometric approximation shape representation for the mesh entities that need higher order shape. The advantageous properties of Bezier polynomial [58] enable the extension of the curving procedure to any higher order. Considering the character of the structured meshes in critical regions such that mesh gradation is maintained is central to the curving procedure. For example, Figure 3.4 shows the difference in meshes without (left image) and with (right image) consideration of maintaining structure in the geometrically graded portion of the mesh.

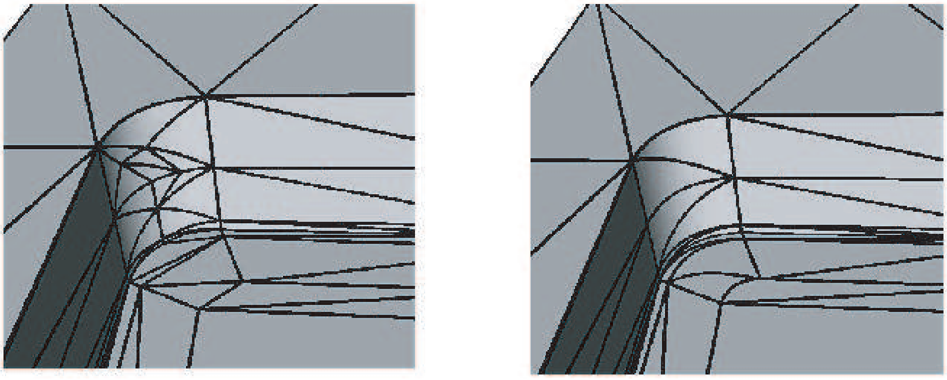


Figure 3.4: Meshes without (left) and with (right) Considering the Ordering of Curving

The local mesh modification based procedure to curve a straight-sided mesh for quadratic mesh geometries given in reference [47] provides key tools for the procedures presented here. The current procedure has been extended in this thesis to:

- Support higher order mesh entity geometry.
- Provide generalized local mesh modification accounting for curved mesh entities.
- Maintain the structure of the mesh configuration around isolated singular edges and vertices by carefully ordering and controlling the process of curving mesh entities to ensure that the appropriate mesh gradations are maintained.

All three of these improvements are important to the generation of curved meshes appropriate for optimal p-version analysis. The proper order and control of the mesh modification operations accounting for the mesh gradations is a most critical aspect of this process. The mesh curving procedure orders the mesh entity curving into the following two steps:

- Curve the singular feature isolation mesh to ensure that a proper structure is maintained in the curved mesh.
- Curve the remaining mesh entities classified on curved boundaries to the required order of approximation. Since curving entities can lead to invalid mesh, this step includes the application of curved mesh modification operations including entity shape change, splits, collapses and swaps as needed to ensure that a valid mesh with acceptable curved elements is created.

The implementation of the above algorithm will be presented in part II of this thesis in detail.

Part II

Construction of Near Optimal p-Version Meshes

CHAPTER 4

Identifying and Meshing Singular Features and Thin Section Structures

This chapter discuss the procedure to automatically identify and mesh the singular features and thin sections of the 3D curved domains for the use of p-version method.

4.1 Identifying Singular Features

Let G^{d_j}, $d_j = 0, 1$ donate the model vertices and edges of the problem domain Ω. The singular features are isolated as the set of model edges whose interior dihedral angles are larger than π [17, 49],

$$\hat{G}_i^1 = \{G_i^1 \mid \omega_i \geq \pi\}. \tag{4.1}$$

Since the interior dihedral angle ω_i for a model edge G_i^1 in a curved domain may not be a constant, a variation ω along the model edge needs to be examined that requires:

- The computation of the interior dihedral angle ω at a specified location of the model edge.
- An algorithm to determine whether the dihedral angle of the model edge is larger than π, and in the case of variable dihedral angles, determining where the angle becomes 180^0.

The interior dihedral angle ω at a specified location on the model edge is defined as the angle between the two tangent planes of the two connected model faces at this point. Let $E_0(x_0, y_0, z_0)$ be the specified point at the model edge G_0^1. $\boldsymbol{A}(a_1, a_2, a_3)$ and $\boldsymbol{B}(b_1, b_2, b_3)$ are the two face normal vectors at point E_0 as shown in Figure 4.1(a). Vector $\boldsymbol{A}$ and $\boldsymbol{B}$ can be easily obtained through the queries of the solid modeler. Therefore, the two tangent plane at the point E_0 can be determined as,

$$a_1(x - x_0) + a_2(y - y_0) + a_3(z - z_0) = 0, \tag{4.2}$$

$$b_1(x - x_0) + b_2(y - y_0) + b_3(z - z_0) = 0. \tag{4.3}$$

The angle θ_0 between these two tangent planes can be computed as,

$$\cos\theta_0 = \frac{a_1b_1 + a_2b_2 + a_3b_3}{\sqrt{a_1^2 + a_2^2 + a_3^2}\sqrt{b_1^2 + b_2^2 + b_3^2}} = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\sqrt{||\boldsymbol{A}|| \cdot ||\boldsymbol{B}||}} \tag{4.4}$$

Since θ_0 is computed as the angles between two planes and it always lies between $[0, \pi]$, additional information is needed to determine whether the interior dihedral angle $\omega = 2\pi - \theta_0 > \pi$ or $\omega = \theta_0 < \pi$.

The approach used here starts from computing the intersection edge $\boldsymbol{L}$ between the two tangent planes. Since $\boldsymbol{L}$ must go through the point E_0, the edge can be easily determined as,

$$\frac{x - x_0}{p} = \frac{x - y_0}{q} = \frac{x - z_0}{r}, \tag{4.5}$$

where,

$$p = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix}, q = \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix}, r = \begin{vmatrix} a_2 & a_1 \\ b_2 & b_1 \end{vmatrix}. \tag{4.6}$$

Therefore, two tangent vectors $\boldsymbol{C}$ and $\boldsymbol{D}$ perpendicular to the intersection edge L at the tangent planes can be easily determined.

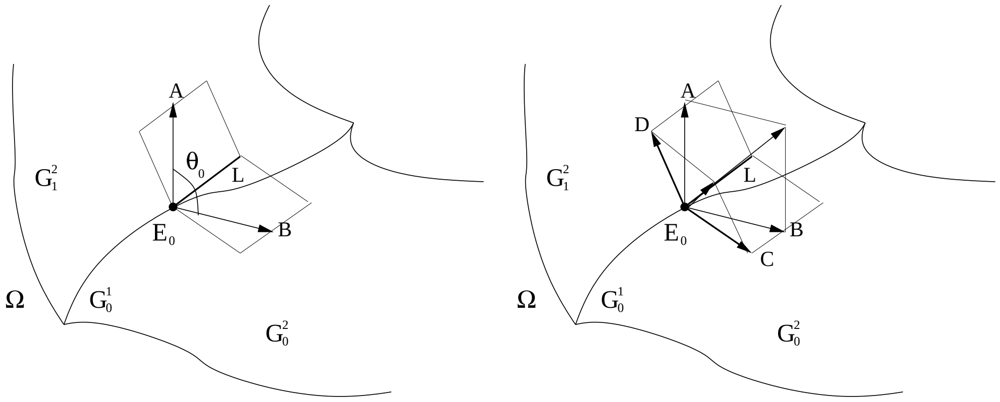
From this, compute

$$\cos\beta_0 = \frac{(\boldsymbol{A} + \boldsymbol{B}) \cdot (\boldsymbol{C} + \boldsymbol{D})}{\sqrt{||\boldsymbol{A} + \boldsymbol{B}||||\boldsymbol{C} + \boldsymbol{D}||}} \tag{4.7}$$

Since the vector $\boldsymbol{A} + \boldsymbol{B}$ always points outside of the model region, the value of $\cos\beta_0$ can serve as the indicator to determine how the interior dihedral angle ω is related to the angle θ_0. Considering that $\boldsymbol{A, B, C, D}, \perp \boldsymbol{L}$, $\boldsymbol{A} \perp \boldsymbol{C}$ and $\boldsymbol{B} \perp \boldsymbol{D}$, $\cos\beta_0 = \pm 1$. Thus,

- $\cos\beta_0 = 1$, the vector $\boldsymbol{C} + \boldsymbol{D}$ points outside of the model region, the interior dihedral angle $\omega = 2\pi - \theta_0$ as shown in Figure 4.1(b).

- $\cos\beta_0 = -1$, the vector $\boldsymbol{C}+\boldsymbol{D}$ points inside of the model region, the interior dihedral angle $\omega = \theta_0$.



(a) Angle θ_0 Between the Two Tangent Planes

(b) Computation of the Angle between Vectors $\boldsymbol{A}+\boldsymbol{B}$ and $\boldsymbol{C}+\boldsymbol{D}$

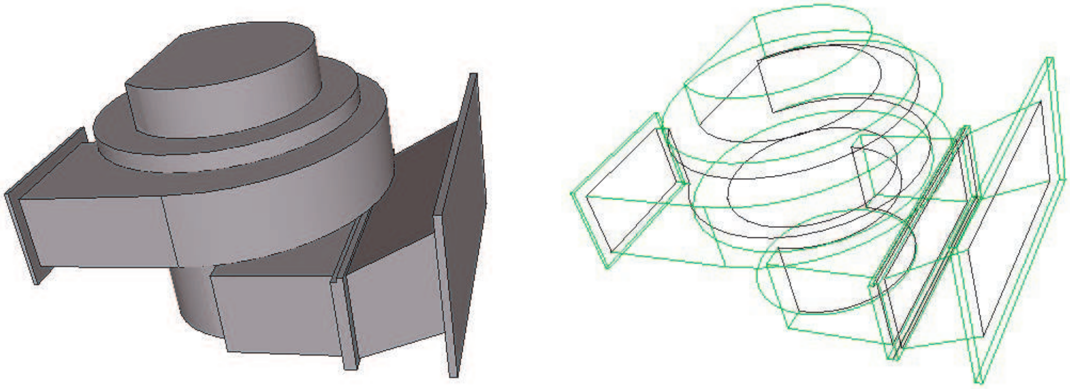
Figure 4.1: Computation of the Interior Dihedral Angle at Points E_0 for Model Edges G_0^1

The algorithm to determine whether interior dihedral angle ω of a model edge is larger than π depends on the model faces coming to the model edge. If both of the model faces are planar the interior dihedral angle is constant along the model edge and can be calculated at any location along the edge. If either one of the model faces is non-planar, the dihedral angle may no longer be constant. Therefore, the maximum dihedral angle ω along the model edge is searched through Golden Section method [29]. The process terminates either the maximum dihedral angle if found or the computed dihedral angle in the middle searching procedure is larger than π.

Figure 4.2(a) shows a real 3D geometric model that the isolated singular model edges are marked as dark in shown Figure 4.2(b).

4.2 Generation of Geometrically Graded Meshes Around Singular Features

Starting from a linear surface mesh, the structured cylindrical layer meshes around the singular edges are generated by growing the boundary layer mesh on it.



(a) Geometric Model (b) Singular Model Edges Marked as Dark

Figure 4.2: Automatically Isolate Singular Model Edges

The boundary layer mesh is constructed using the growth curves originated from the surface nodes [60]. These boundary nodes are connected to form the cylindrical elements of the layer mesh. The intersection between the exposed boundary layer mesh faces and surface mesh faces is check. If there are intersections, it is necessary to reduce the height of a layer, or trim the number layers through the thickness in order to resolve the intersections [60]. Each step in the processes performs checks to ensure the validity of the mesh and to optimize the quality of the created mesh entities.

The structured geometric graded mesh generation is controlled through a set of attributes applied to the model entities being isolated. In this thesis, the attributes are defined in a general form and then automatically associated to the singular model edges once they are identified. The attributes include the parameters of:

- Total height,
- Gradation factor,
- Number of layers.

The geometric gradation factor q that accounts for the strength of solution singularity is set to value 0.15 with respect to the total layer height and two initial layers of elements are generated that will be reset in the adaptive analysis [61].

The total height T of the graded layer mesh defines how far the singular behavior of the solution is expected to exist into the domain. For a general curved three dimensional model, the range and strength of the solution singularities caused by the geometric model shape is difficult to predict a-priori. On the assumption that the adaptive procedure will adjust the layer height and reset layer numbers based on correction indicator in each step, the automatic p-version mesh generator initially defines the height of the graded layer mesh T for an isolated singular edge as one half of the distance to its closest non-connected model boundary entity that considers the dominate singular behavior is reasonably local to the isolated singular edges. A non-connected entity is one which does not share any common lower order model entities. A simple algorithm based on the shortest distance between a set of equally parameterized distributed sampling locations to their closest points on non-connected model boundaries is used to get a useful estimation of T. Refinement is applied for the initial equal sub-parametric domains if the variation of the computed shortest distances is larger comparing to the local geometry. Therefore, the thickness for each layer t_i is calculated as,

$$t_i = q^i T,\ i = 0, 1. \tag{4.8}$$

Figure 4.3 shows an example of singular edge isolation for a "penny shaped crack" on the interior of a domain. In this example cylindrical hexahedral and prismatic elements are generated around the isolation edge.

4.3 Identify Thin Section Structures

The definition of a thin section is closely related to size of and order of the elements in the mesh. The geometric characteristic for a thin section is the dimension through the thickness is far less than the "in-plane" dimensions. A key challenge to properly mesh the thin sections is to automatically identify and isolate thin sections from the rest of domain. A plausible approach appropriate for identifying thin

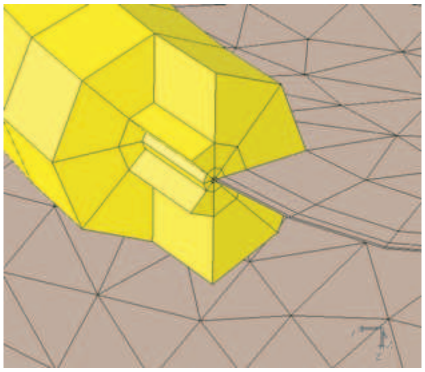


Figure 4.3: Example of Graded Mesh for an Isolated Singular Model Edge

section is to use medial surface information [50, 96, 129].

The medial surface is the locus of the center of interior points that is equidistant to more than one points on the model boundary. The medial surface can provide the information about the region's geometry and topology [98, 99] as follows:

- Indication of local feature size (or local thickness) by the distance from a medial surface point to its closest boundary points.
- Information on opposite boundary points by relating the closest boundary points to a medial surface point.

Given a general curved solid model with a surface triangulation, a set of points on the medial surface is computed using an Octree-based tracing algorithm [129]. Such points along with the topology classification and geometry similarities information are used to automatically identify the thin sections. Central to the algorithm is to find pairs of triangles on "opposite model faces" that are close to each other relative to their size, thus indicating they are within a thin section.

Given a set of intersecting points between the medial surface and Octants computed from the tracing algorithm [129], a pair of triangles M_i^2 and $M_{i'}^2$ is candidate *thin section triangle pair* if there exist a pair of closest boundary points P_1 and P_2 from a medial surface point E_i^0, such that $P_1 \subset \overline{M}_i^2$ and $P_2 \subset \overline{M}_{i'}^2$ (over bar means closure of a triangle), where the P_1 and P_2 have the following properties:

1. The ratio of thickness (defined as the diameter of the maximum inscribed

sphere associated with E_i^0) to the average size of M_i^2 and $M_{i'}^2$ is smaller than a default value, for example 1/2 of the average edge length of the element.

2. The angle formed by the outward normal to M_i^2 and $M_{i'}^2$ is close to π.

Figure 4.4 shows the situation of two medial surface points E_i^0, E_j^0 and the candidate thin section triangles defined by conditions of (1) and (2).

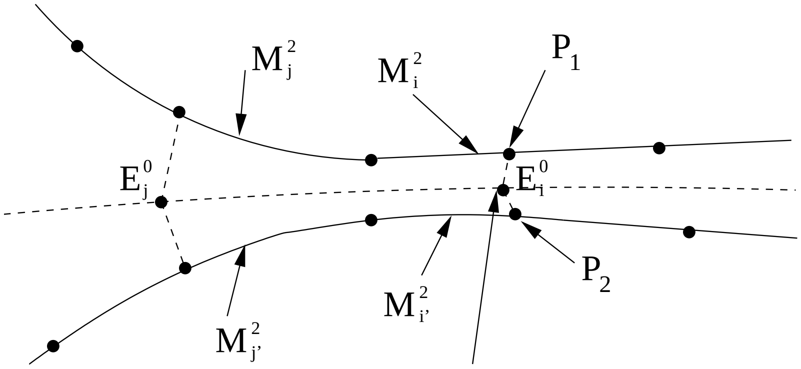


Figure 4.4: Candidate Thin Section Triangle Pairs Identified by Medial Surface Points E_i^0 and E_j^0

A candidate thin section triangle pair is further processed to ensure that all points on their closures meet those two conditions. For example, the triangle pair $(M_j^2, M_{j'}^2)$ in Figure 4.4 will determined as non-thin section pair.

4.3.1 Collect Starting Thin Section Triangle Sets

Given a medial surface point E_i^0, introduce,

$$|E_i^0|_* = \begin{cases} 1 & \text{thin} \\ 0 & \text{not thin} \end{cases} \tag{4.9}$$

to indicate whether E_i^0 define a thin section triangle pair, where 1 indicates the medial surface point is associated with thin triangle pair and 0 means they are not part of a thin section. Denote the triangle pair as,

$$[E_i^0]_* = \{M_k^2, M_{k'}^2\}. \tag{4.10}$$

With the above notations, a starting thin section triangle set is defined as,

$$\hat{G}_j^2 = \{\ M_i^2 \mid M_i^2 \sqsubset G_j^2 \ \mathit{and}\ M_i^2 \in [E_i^0]_* \ \mathit{and}\ |E_i^0|_* = 1\ \}. \tag{4.11}$$

At this point, each $\hat{G}_j^2$ is uniquely associated with a G_j^2 that may be subdivided later as discussed in Section 4.4.1 to define paired thin section surface patches. For this unique association, the identity tags of "opposite" sets for $\hat{G}_j^2$ can be recorded during the construction of $\hat{G}_j^2$. Generally, $\hat{G}_j^2$ may have one or more opposite sets denoted as $opp(\hat{G}_j^2)$. A simple 2D example in Figure 4.5 shows that,

$$\hat{G}_1^1 = \{M_a^1,\ M_b^1,\ M_c^1\}\ \mathit{and}\ opp(\hat{G}_1^1) = \{\hat{G}_3^1\}, \tag{4.12}$$

$$\hat{G}_2^1 = \{M_d^1,\ M_e^1,\ M_f^1\}\ \mathit{and}\ opp(\hat{G}_2^1) = \{\hat{G}_3^1\}, \tag{4.13}$$

$$\hat{G}_3^1 = \{M_{a'}^1,\ M_{b'}^1,\ M_{c'}^1,\ M_{d'}^1,\ M_{e'}^1\}\ \mathit{and}\ opp(\hat{G}_3^1) = \{\hat{G}_1^1,\ \hat{G}_2^1\}. \tag{4.14}$$

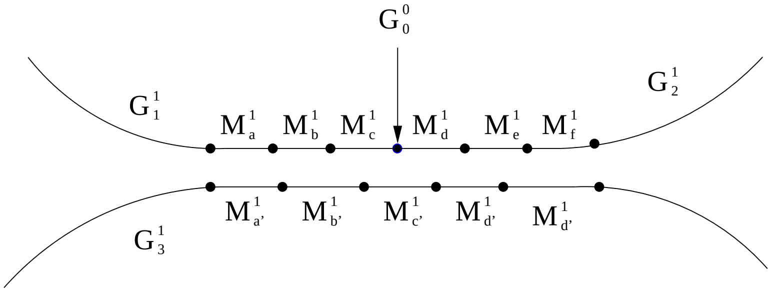
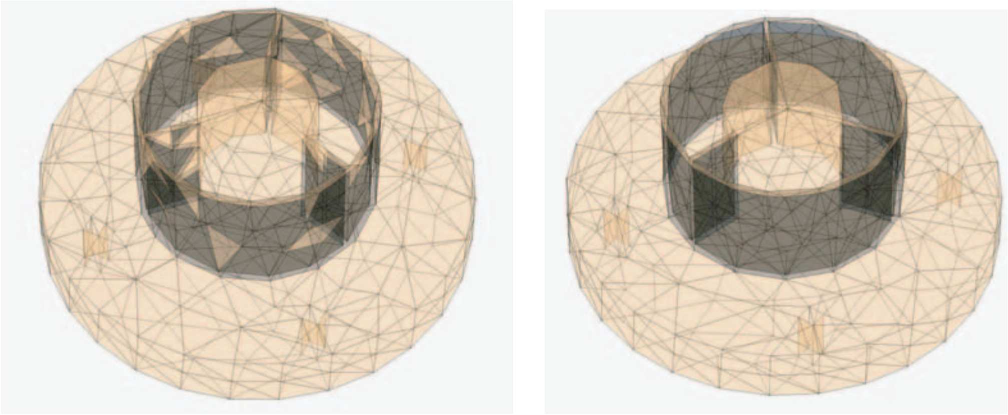


Figure 4.5: An Example to Demonstrate the Starting Triangle Sets

The sets as defined to this point may have to be later split or merged to represent one thin section surface patch.

4.3.2 Determining the Missing Thin Section Triangles

The majority of thin section triangles are identified by the medial surface points in the tracing algorithm. However, some are missed, see Figure 4.6(a), where the dark shaded triangle faces are the identified thin section triangles by using the medial surface points and the light grey shaded triangle faces are the missing triangles. To determine whether a missing triangle M_e^2 on G_j^2 belongs to $\hat{G}_j^2$, the comparison between the local thickness h_d and the local mesh size h_e at M_e^2 is needed. h_e is computed as the longest edge length connected to the vertices of the triangle face M_e^2. h_d is obtained by searching for the closest points of the vertices of the triangle face M_e^2 on the model faces that are known to be opposite to G_j^2. If h_d/h_e is smaller than a default value, place M_e^2 in $\hat{G}_j^2$. The triangle $M_{e'}^2$ that the closest point is on is defined as the opposite triangle to M_e^2 and put $M_{e'}^2$ in the set $\hat{G}_{j'}^2$ that is opposite to $\hat{G}_j^2$. Based on such criterion, Figure 4.6(b) show the complete thin section triangles to define the thin section surface patches.



(a) [The Thin Section Triangles Obtained by Medial Surface Point

(b) The Complete Thin Section Triangle Surface Patches

Figure 4.6: Examples for Determine Complete Thin Section Triangles

4.4 Meshing Thin Sections

The primary goal to mesh the thin section structures is to generate one layer of prismatic elements through the thickness direction. Because the initial surface triangulations of the two thin section surface patches do not match each other, the direct connection between the thin section triangle pairs will often not produce the desired meshes. Therefore surface mesh modification is used to ensure the topological and geometrical matching between the paired thin section surface patches. The algorithm consists of the following steps:

- Construct boundaries for thin section surface patches.
- Topologically match the boundaries of the paired thin section surface patches.

- Delete the surface triangulation of one thin section surface patches.
- Copy the remaining thin section surface patches to the opposite model face.
- Directly connect the two surface triangulation of the paired thin section surface patches to construct one layer structured prismatic elements.
- Split the prismatic elements to expose triangle faces for the isotropic volume mesh generator.
- Apply isotropic volume mesh to fill the remaining domain.

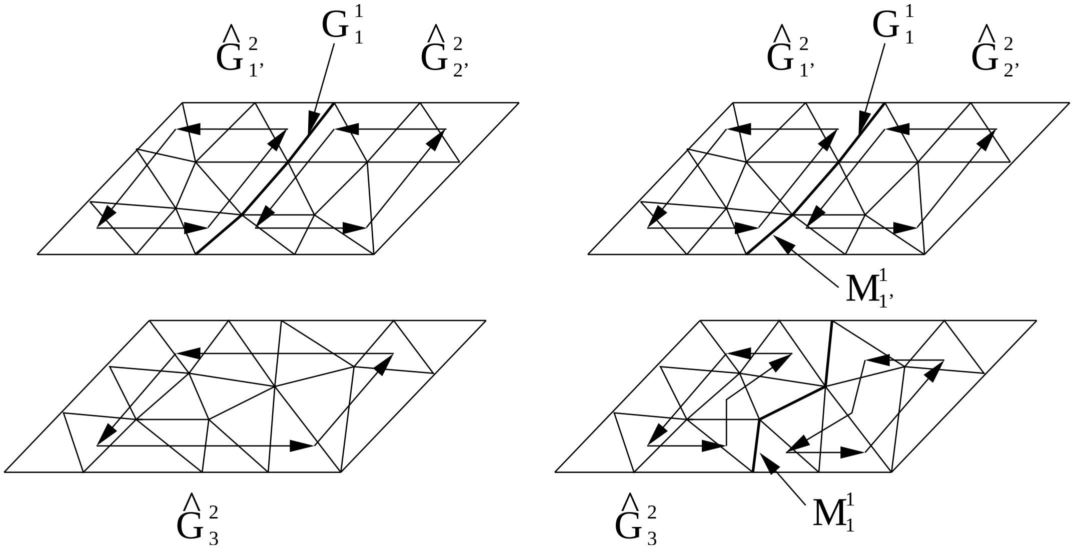
Each step in the algorithm is described in detail in the following subsections.

4.4.1 Construct the Boundary Loops on Thin Section Surface Patches

The boundary loop of a thin section triangle set $\hat{G}_j^2$ is constructed by a set of edges used only by one triangle in the set. To complete the definition of the thin section surface patches opposite each other, the boundary loops of the paired surface patches have to be matched. The process can lead to the need to split one thin section triangle loop into multiple thin section loops. As an example, Figure 4.7(a) shows three thin section triangle sets $\hat{G}_1^2$, $\hat{G}_2^2$, and $\hat{G}_3^2$, where $\hat{G}_3^2$ is opposite to both $\hat{G}_1^2$ and $\hat{G}_2^2$. In this case, splitting $\hat{G}_3^2$ to match the surface patches of $\hat{G}_1^2$ and $\hat{G}_2^2$ is needed as shown in Figure 4.7(b). Note the loops in $\hat{G}_1^2$ and $\hat{G}_2^2$ can not be merged to form one loop since the triangulation on the model edge G_1^1 must be maintained. For a thin section triangle set $\hat{G}_j^2$ that has multiple opposite surface patches $\{\hat{G}_{k'}^2\}$, the procedure to split the set by using the opposite triangle pairs and connectivity information as follows:

- Construct the boundary loop of the surface patches $\hat{G}_j^2$ and $\{\hat{G}_{k'}^2\}$ shown in Figure 4.7(a).
- For each of the opposite surface patch $\hat{G}_{k'}^2$, start from any mesh edge $M_{i'}^1$ of the boundary loop and looks for its opposite mesh edge M_i^1. The opposite mesh edge M_i^1 is determined as the boundary edge of surface patch $\hat{G}_j^2$ whose connected thin section triangle face is opposite to the thin section triangle face connected to $M_{i'}^1$.

- If M_i^1 is not in the boundary loop, add M_i^1 to the boundary loop opposite to $\hat{G}_{k'}^2$. Else go to next mesh edge $M_{i'+1}^1$. For example, M_1^1 shown in Figure 4.7(b) is added to the boundary loop because its opposite edge $M_{1'}^1$ is on the loop of $\hat{G}_{1'}^2$.



(a) Loops on Thin Section Surface Patches (b) Opposite Loops

Figure 4.7: Construction of Boundary Loops for Each Thin Section

4.4.2 Boundary Matching for the Thin Section Surface Patches

Since the boundary loops determined by the initial surface mesh triangulation of the two opposite thin section surface patches often do not match each other in terms of the number of edges, a local mesh modification based procedure is applied to ensure topologically matching between the each paired boundary loops. Given a paired of opposite loops (L, L') with different number of edges, the process begins to divide each loop into a set of segments (S_i, S_i') as,

$$L = \cup S_i, \qquad L' = \cup S_i'. \tag{4.15}$$

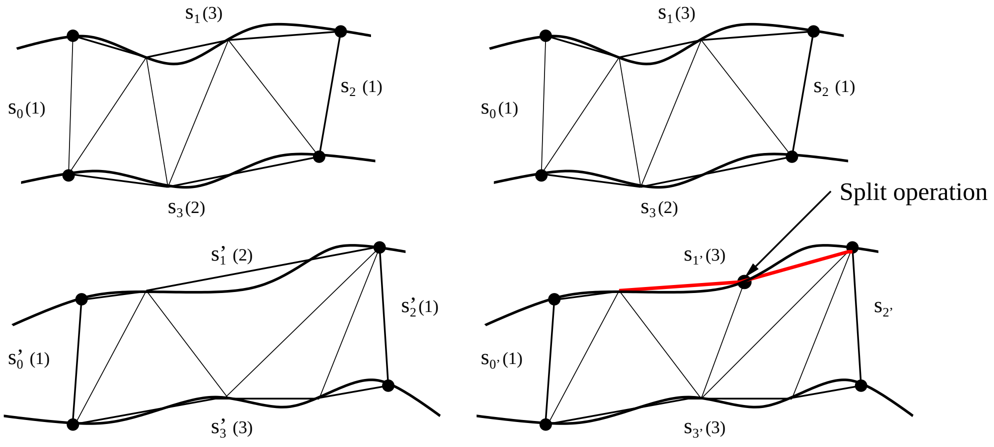
Each segment is determined by two vertices whose two connected mesh edges on the boundary loop classify on different model entities. As an example, Figure 4.8(a) shows that a pair of opposite loops have been decomposed into four sets of segments. Since the segments S_0, S'_0, S_2 and S'_2 only have one mesh edge, S_0 automatically matches S'_0 and S_2 automatically matches S'_2. As for the segments S_1 and S_3 with 2 and 3 mesh edges, their opposite segments S'_1 and S'_3 have 3 and 2 mesh edges respectively. Split and/or collapse operations are applied on each set of segments to create and/or delete mesh edges to ensure the resulting number of the opposite segments is equal. For example, Figure 4.8(b) shows a split operation is applied on the segment S'_1 to produce one more edge to match the segment S_1 and a collapse operation is applied on the segment S'_3 to match segment S_3 shown in Figure 4.8(c).

For a set of opposite segments (S_i, S'_i) that have different number of edges, the selection of the segment and the type of operations to be applied is determined based on the shape quality measured with aspect ration [46] of the resulting mesh. The collapse operation is applied on the shortest edge possible of the segment with more mesh edges and the split operation is always applied on the longest edge possible of the segment with fewer mesh edges. The split operation can always be applied that is essential for the set of opposite segments to have the same number of mesh edges in the case that the collapse operations fails to be applied. The process terminates till the number of the mesh edges of the set of opposite segments is equal.

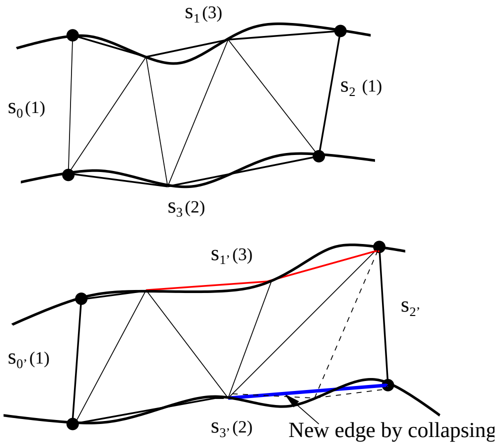
4.5 Match Thin Section Triangle Surface Patches

The surface triangulation between the thin section surface patches after matching the boundary loops often do not match each other as shown in Figure 4.8(c). The procedure used here starts from deleting the triangle surface patch on one patch (see Figure 4.9(a)) and then copying the surface triangulation of the remaining patch to the opposite model face (see Figure 4.9(b)). The process consists of the following steps:

1. Delete one thin section triangle surface patch that produces an empty polygon on the opposite model face. The deleted thin section triangle surface patch is selected as the one with worse mesh shape quality measured with aspect ratio metrics [46].



(a) Paired Opposite Segments (b) Split Operation for Segment $S_{1'}$



(c) Collapse Operation on Segment $S_{3'}$

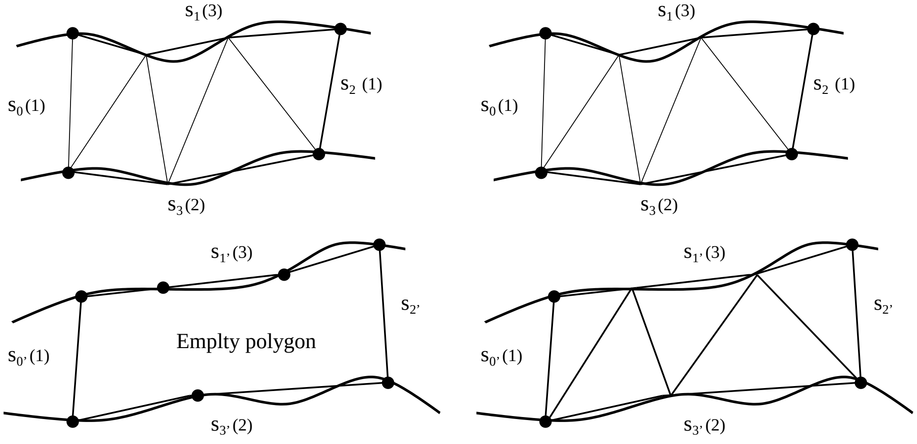
Figure 4.8: Split and Collapse Operations to Match Paired Opposite Segments for the Boundary Loops

2. For the vertices M_i^0 on the boundary loop of the remaining thin section triangle surface patch, find the opposite vertices $M_{i'}^0$ on the opposite boundary loop through edge adjacency. Compute the target locations for the opposite vertices $M_{i'}^0$ by querying the closest points from solid modeler for vertices M_i^0 on the model face that $M_{i'}^0$ classified on. Incrementally move the vertices $M_{i'}^0$ to the target locations.

3. Copy the rest of the vertices of the remaining triangle surface patch to the

opposite model face. The locations for the copied vertices on the opposite model face are determined by finding the closest points on the opposite model face.

4. Topological connect the vertices on the opposite model face to create the matched surface triangulation.
5. Connect the two matched thin section triangle surface patches to create one layer prismatic elements.

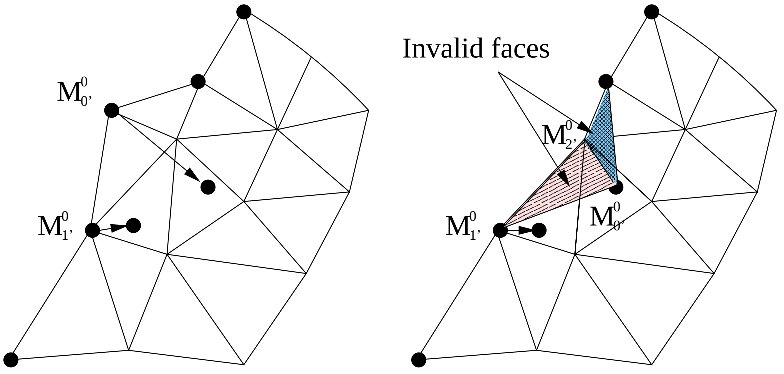


(a) Delete One Triangle Surface Patches

(b) Match Triangle Surface Patches

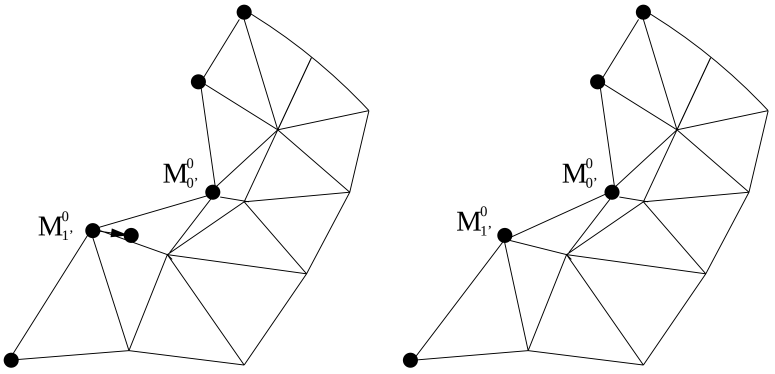
Figure 4.9: Surface Triangulation Match for Thin Sections

Each vertex M_i^0 on the remaining surface patch must have an opposite vertex $M_{i'}^0$ on the opposite model face to produce one layer prismatic elements. In the case of the vertices M_i^0 on the boundary loop, $M_{i'}^0$ can easily be found on the opposite boundary loop through edge adjacency. The desired target locations for $M_{i'}^0$ can be obtained by directly querying the closest points of M_i^0 on the opposite model face. These vertices are incrementally moved to the computed target locations. The movement of the mesh vertices $M_{i'}^0$ on the boundary loop can cause the surface triangle faces to become unacceptable. Figure 4.10(b) shows an example where two shaded triangle faces marked are unacceptable because of moving vertices $M_{0'}^0$ to its target location moves too far over existing elements. This problem is solved by applying the local mesh modification procedure [82] and has to come before the matching of surface triangulation.



(a) Initial Mesh

(b) Moving $M_{0'}^0$ Causing Invalid Faces



(c) Collapse $M_{2'}^0$ to $M_{0'}^0$

(d) Moving $M_{1'}^0$

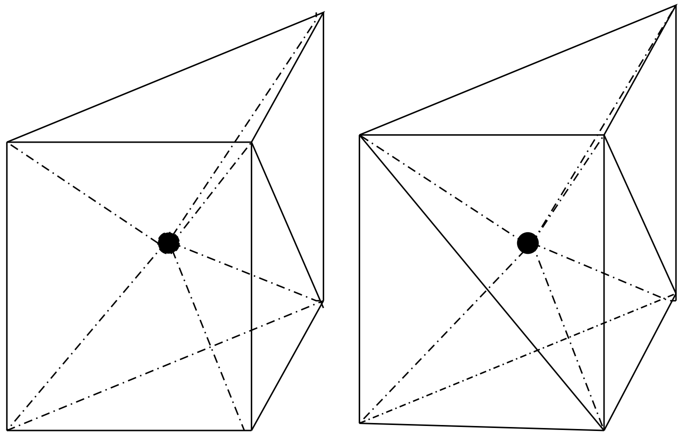
Figure 4.10: Incrementally Move Vertices on the Boundary Loop to the Target Locations

For the rest of the vertices on the remaining surface patch, new vertices are created on the opposite model face. The edges are created by the connection of vertices and faces by edges.

4.6 Volume Mesh Generation

When the thin section triangle surface patches are topologically and geometrically matched, prismatic elements are constructed by directly connecting each paired triangles classified on the two opposite model face. The remaining domains are filled by applying the generalized isotropic volume mesh generator developed for the h-version method that can handle very complex 3D geometries [114]. Since the volume mesh generator requires the exposure mesh faces to be triangles, splitting one layer of prismatic elements into a combination of tetrahedra and pyramids is needed.

The process of splitting of a prismatic element with one or more faces exposed to the tetrahedral volume mesher is a function of how many quadrilateral faces are exposed. Figure 4.11(a) shows the template for splitting one quadrilateral face that the original prismatic element is decomposed with 4 tetrahedra and 2 pyramid elements. Figure 4.11(b) shows the template for splitting two quadrilateral faces that the original prismatic element is decomposed with 6 tetrahedra and 1 pyramid elements. Both of the templates introduce one interior vertex inside the original prismatic region and the location for the interior vertex is computed with centroid parametric coordinate $(1/3, 1/3, 0)$ on the original prismatic element. The case of three exposed faces identifies a surrounded prismatic element which can not be generated by the procedures used to isolated thin sections.



(a) Split One Quadrilateral Face

(b) Split Two Quadrilateral Faces

Figure 4.11: Split One Prismatic Elements to Create Exposure Triangles for Volume Mesh Generation

Complete mesh results are shown in Section 5.4.

CHAPTER 5
p-Version Mesh Curving

This chapter describes a local mesh modification based Bezier curving procedure needed by the automatic p-version mesh generation algorithm. The objective of the procedure is to create a valid curved mesh up to any required order while maintaining the geometric gradation for the singular model edges and prismatic thin section meshes.

5.1 p-Version Mesh Curving with Gradation and Thin Section Mesh Control

The straight-sided p-version meshes generated by the procedure presented in Chapter 4 are mixed topological meshes that contain tetrahedral, prism, hexahedral and pyramid elements. Comparing to the conventional isotropic straight edges volume meshes, the straight-sided p-version meshes are characterized by:

- Graded layer mesh for the isolated singularities.
- One layer structured prismatic elements for the identified thin sections.

The conventional quadratic curving procedure developed for the straight edged isotropic mesh [47] does not consider the characters of the straight-sided p-version meshes. In the process of applying local mesh modifications to correct the invalid elements caused by curving the mesh entities, the characters of the p-version meshes, particular the graded layer meshes for the singularities, are likely disrupted (See the right image of Figure 3.4).

The p-version mesh curving procedure developed in this work curves the mesh entities classified on the curved boundaries in an appropriate order with gradation and thin section meshes control as follows:

- Curve the graded layer meshes to ensure that the proper structure of the singular mesh regions is maintained. This step is achieved by simultaneously curving the mesh entities on the boundary and those within the structured layer above it as discussed in Section 5.2.

- Curve the prismatic thin section meshes. This step is accomplished by curving prismatic element one at a time as discussed in Section 5.3.
- Curve the remaining mesh entities on the curved boundaries.
- In all the three steps, curving mesh entities may lead to an invalid mesh. A set of curved local mesh modification operations build on the straight edge mesh modification operations are applied in an appropriate order to incrementally correct the invalid elements as discussed in Section 5.5.

Curving the graded layer and thin section mesh before curving the remaining mesh entities on the boundaries is central to this procedure to maintain the characters of the p-version meshes. The application of the curved local mesh modification operations is the key technique to ensure that a valid curved mesh is created.

5.2 Curve the Graded Meshes for Singular Features

To ensure a properly curved graded mesh around the singular features, the process simultaneously curves the mesh entities on the boundary and those within the structured layer above it. Figure 5.1 demonstrates this process with Figure 5.1(a) showing the linear structured layer mesh associated with a portion of a curved model edge. Figure 5.1(b) shows how the local gradation would be disrupted if only the mesh edge, M_0^1, classified on the model edge G_0^1 were curved. To avoid this problem the "parallel" and "diagonal" mesh edges (see Figure 5.1(a)) are also curved such that the entire layer maintains its gradation and element shapes are smooth. This process is accomplished by moving the control points of the parallel and diagonal edges based on how far the control points of the edge on the model boundary are moved with consideration given to the change in length of the edges being curved. The need to account for the length of the edges being curved is demonstrated in Figure 5.1(c) where all control points were moved the same distance as the associated control points on the mesh edge M_0^1 classified on the curved boundary. It is clear that the short edges are curved more than desired. The method used scales the shape change of the base mesh edge accounting for edge lengths as follows. Let L_{base} , L_{P_i} and L_{D_i} be the straight-sided edge length of the mesh edge on the curved boundary (the base edge), *ith* parallel and *ith* diagonal edge. The scale factors S_{P_i} and S_{D_i} for *ith* parallel and diagonal edges applied to their movements with respect to that of the base edge are,

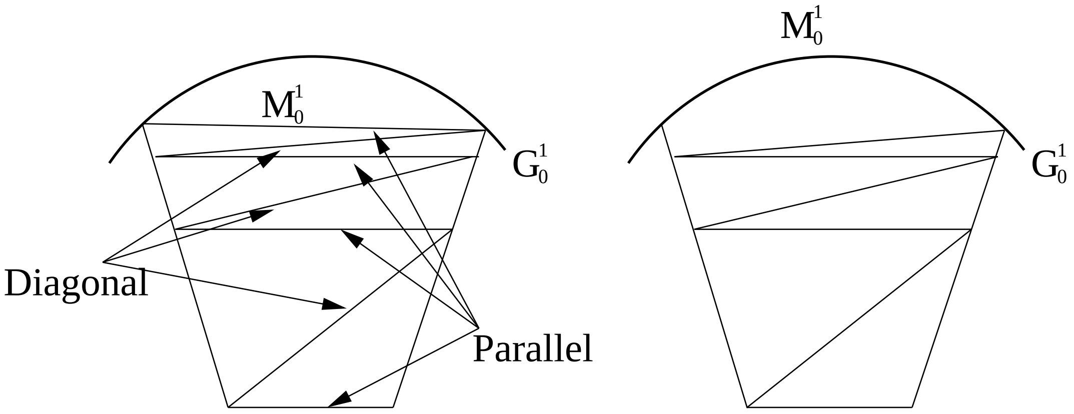
$$S_{P_i} = \frac{L_{P_i}}{L_{base}}, \tag{5.1}$$

$$S_{D_i} = 0.5(S_{P_{i-1}} + S_{P_i}). \tag{5.2}$$

Figure 5.1(d) shows a smooth curved isolation mesh by using this method. The curving procedure at this point only curves the graded layer meshes as needed.

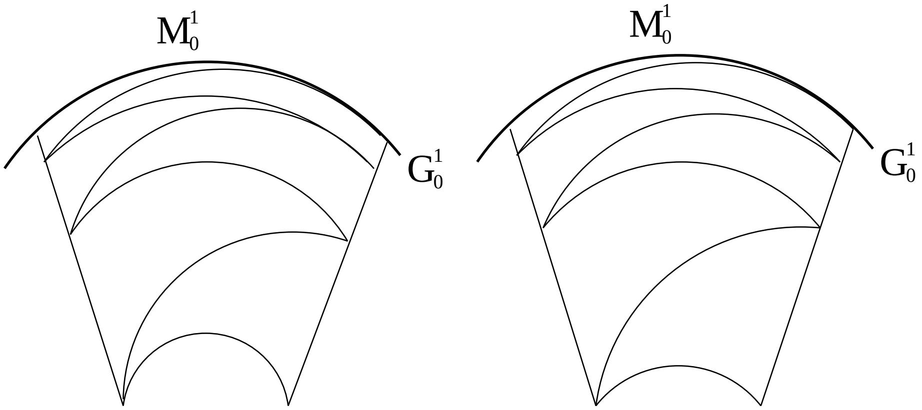
5.3 Curve the Prismatic Thin Section Meshes

For the one layer prismatic meshes classified on the curved thin sections, the process curves the prismatic elements one at a time to maintain the structure as demonstrated with an example in Figure 5.2. Figure 5.2(a) shows one of the quadrilateral faces of a prismatic region. Figure 5.2(b) shows that the quadrilateral face becomes invalid after the mesh edge M_0^1 is curved to the boundary. The invalidity is caused by the interference between the curved mesh edge M_0^1 and the uncurved mesh edge M_1^1. The conventional curving procedure [47] will apply local mesh modification operations to eliminate the invalidity and the resulting mesh is shown in Figure 5.2(c) that disrupts the structures for the thin sections. However, this invalidity is avoided if the mesh edges M_0^1, M_1^1 are curved to the boundary at the same time shown in Figure 5.2(d). Therefore, the process picks up one prismatic element from the thin section meshes and curves all the lower bounded mesh entities of the prismatic element before applying the validity check. In case that the curved prismatic element remains invalid, curved local mesh modification operations as discussed in Section 5.5 will be applied to correct the invalidity.



(a) Linear Graded Mesh

(b) Curve M_0^1 to G_0^1



(c) Curve the Graded Mesh the Same as M_0^1

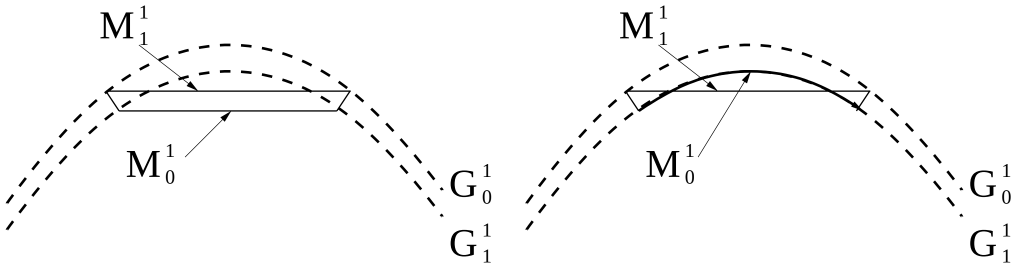
(d) Scaled Curve the Graded Mesh

Figure 5.1: Scaled Curve the Graded Mesh

5.4 Curve Mesh Entities on the Curved Boundaries

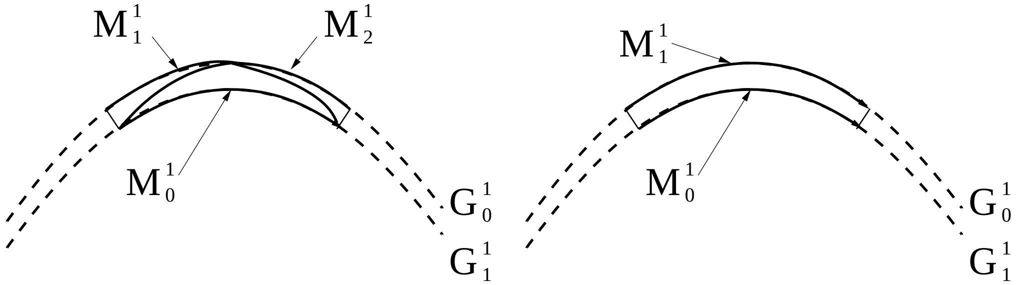
After curving the graded mesh around the isolated mesh edges and the prismatic meshes for the thin sections, the curving procedure curves the remaining mesh edges/faces classified on curved model boundaries. The entities are put into a list with the attachment of a proposed Bezier geometric shape to curve them to the boundary.

The process of curving the mesh entities traverses the list. If the movement associated with this curving maintains validity of the connected mesh entities, the entity is curved and removed from the list. If any invalidates arise, additional



(a) Linear Prismatic Mesh

(b) Curve M_0^1 to G_1^1 and Cause Invalidity



(c) Local Mesh Modification to Correct the Invalidity

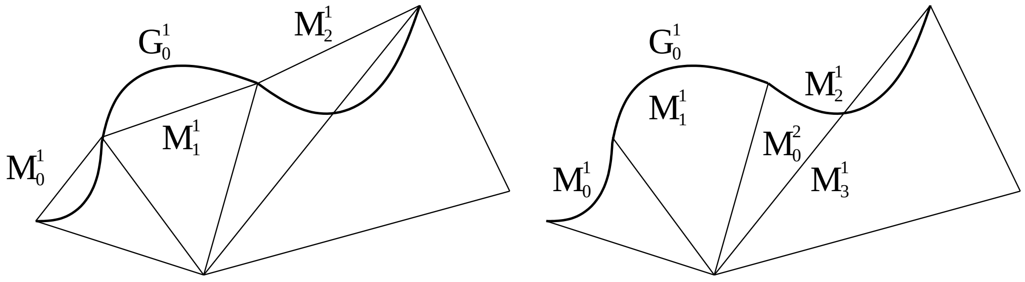
(d) Simultaneously Curve the Edges M_0^1, M_1^1

Figure 5.2: Curved the Prismatic Thin Section Meshes

processing is required. Since changing the shape of a mesh entity changes the shape of the mesh entities it bounds, the process of checking the validity of the mesh must check the shape validity of the connected mesh entities as well. This means that in addition to verifying no self-intersection of the curved mesh face or edge, the shape validity of the connected high dimensional mesh entities are checked using the shape check algorithm outlined in Section 3.3. The verification of self-intersection of any curved Bezier mesh edge or faces uses the variation diminishing property that no no straight line intersects a curve more times than it intersects the curve's control polygon [58].

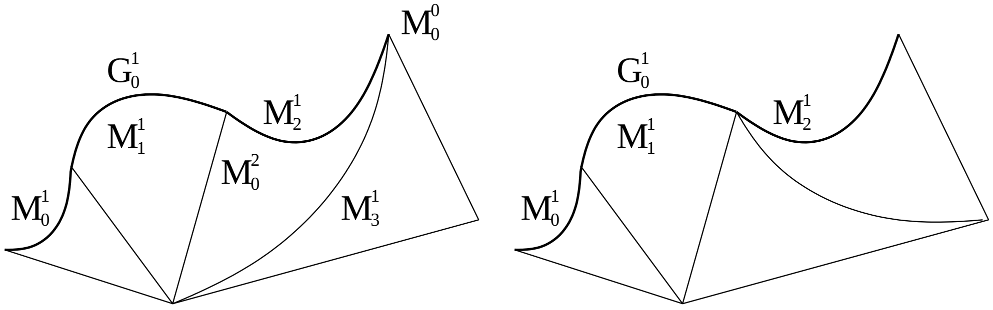
In those cases when a mesh entity does cause an invalidity, local mesh modification operations are applied as indicated in Section 5.3. After the needed mesh modifications are performed the current mesh entity is removed from the list and any new mesh entities created that are classified on curved boundaries are added to the list.

Figure 5.3(a)-5.3(d) shows a simple example of curving the mesh edges on model edge G_0^1. Mesh edges M_0^1 and M_1^1 are curved to the boundary without causing any mesh invalidates. However, face M_0^2 becomes invalid when curving M_2^1. Considering that the bounded angle is already small at vertex M_0^0, although possible shown in 5.3(c), just curving M_3^1 would produce two elements with small angles at vertex M_0^0. Therefore, swapping M_3^1 provides a better opportunity for maintaining larger angles. However, the swapped edge must still be curved as shown in Figure 5.3(d) to avoid mesh invalidities.



(a) Initial Straight-Sided Mesh

(b) Curve M_0^1, M_1^1 and M_2^1



(c) Curve M_3^1 to Correct Invalid Element M_0^2

(d) Swap and Curve M_3^1 to Correct Invalid Element M_0^2

Figure 5.3: Curve the Mesh Entities on Curved Boundary

5.5 Curved Mesh Modification Operations to Correct Invalid Elements

The curved local mesh modifications processes build upon a set of operations that include mesh entity shape modification, split, collapse and swap. These operations are essential to ensure the reliability of the procedure to create valid curved element meshes.

5.5.1 Mesh Entity Shape Modification

Mesh entity shape modification is often successful when curving a mesh entity on the boundary has caused an intersection with another mesh entity and there is adequate room on the "other side" of the intersected mesh entity to re-shape it such that the intersection is eliminated. The properties of the Bezier control polygon are used in this process to determine when and how to perform re-shaping to eliminate the intersection and make the element valid again. For example, in the case of a q^{th} order Bezier tetrahedral region, the geometric approximation shape is represented as in Eq. 2.31,

$$\boldsymbol{x} = \sum_{|\boldsymbol{i}|=q} B_{|\boldsymbol{i}|}(\xi_1, \xi_2, \xi_3, \xi_4) b_{|\boldsymbol{i}|}, \qquad |\boldsymbol{i}| = i + j + k + l = q, \tag{5.3}$$

where $B_{|\boldsymbol{i}|}$ are the Bernstein polynomial defined in Eq, 2.32 and $b_{|\boldsymbol{i}|}$ are the control points. The determinant of the Jacobian J is computed as discussed in Eq. 3.16,

$$J = (\frac{\partial \boldsymbol{x}}{\partial \xi_1} \times \frac{\partial \boldsymbol{x}}{\partial \xi_2}) \bullet \frac{\partial \boldsymbol{x}}{\partial \xi_3}. \tag{5.4}$$

Based on the property of the derivative of a Bezier curve, surface and region are a one order lower Bezier curve, surface and region by differentiating the original control polygon [58], $\frac{\partial \boldsymbol{x}}{\partial \xi_1}$, $\frac{\partial \boldsymbol{x}}{\partial \xi_2}$ and $\frac{\partial \boldsymbol{x}}{\partial \xi_3}$ are another three Bezier tetrahedral regions with polynomial $q-1$. For example, $\frac{\partial \boldsymbol{x}}{\partial \xi_1}$ can be written in the form of,

$$\frac{\partial \boldsymbol{x}}{\partial \xi_1} = \sum_{|\boldsymbol{i}|=q-1} B_{|\boldsymbol{i}|}(\xi_1, \xi_2, \xi_3, \xi_4) b'_{|\boldsymbol{i}|}, \tag{5.5}$$

where $b'_{|\boldsymbol{i}|}$ are vectors obtained by differentiating the original control points $b_{|\boldsymbol{i}|}$ as follows,

$$b'_{ijkl} = q(b_{(i+1)jkl} - b_{ijkl}). \tag{5.6}$$

As an example, Figure 5.4(a) shows a linear quadratic Bezier tetrahedral with the corresponding vectors determined by Eq. 5.6.

Therefore, the J is a Bezier tetrahedral region of order $3(q-1)$ that can be written as,

$$J = \sum_{|\boldsymbol{i}|=3(q-1)} B_{|\boldsymbol{i}|}(\xi_1, \xi_2, \xi_3, \xi_4) P_{|\boldsymbol{i}|}, \tag{5.7}$$

where $P_{|\boldsymbol{i}|}$ are constants. Considering that a Bezier polynomial is bounded by its maximum and minimum control points [58], Eq. 5.7 indicates that $min(P_{|\boldsymbol{i}|}) \leq J \leq max(P_{|\boldsymbol{i}|})$. If $min(P_{|\boldsymbol{i}|}) \geq 0$ that means the determinant of the Jacobian of the region is positive over the region domain. However, if $min(P_{|\boldsymbol{i}|}) \leq 0$, the mesh shape may be invalid. The portion of the element domain that is invalid depends on where the $P_{|\boldsymbol{i}|} \leq 0$ appears. The mesh shape modification procedure checks how the $P|\boldsymbol{i}|$ become negative to determine the candidate control points to correct the invalidity.

If $P_{|\boldsymbol{i}|} < 0$ appears at the vertices, one of the three partial derivative vectors $\boldsymbol{b}$ lies on the "opposite" side of the plane defined by the other two derivative vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ as shown in Figure 5.4(b) that,

$$\boldsymbol{b} \cdot (\boldsymbol{a} \times \boldsymbol{c}) < 0. \tag{5.8}$$

By constraining two of the derivative vectors, for example the vectors $\boldsymbol{a, c}$, the control point defined for the third vector $\boldsymbol{b}$ becomes the candidate point and need to be moved to the other side of plane defined by vectors $\boldsymbol{a, c}$ to make $\boldsymbol{b} \cdot (\boldsymbol{a} \times \boldsymbol{c}) > 0$. The new vector $\boldsymbol{b'}$ after moving the control point can be determined as,

$$\boldsymbol{b'} = \frac{\boldsymbol{b} \cdot \sin\theta}{\cos t}, \tag{5.9}$$

where $\theta = \arccos(\boldsymbol{b} \cdot (\boldsymbol{a} \times \boldsymbol{c}))$ and t can be the default minimal angle for $\boldsymbol{b'}$ pass through the plane defined by $\boldsymbol{a} \times \boldsymbol{c}$ shown in Figure 5.4(c). The resulting valid curved tetrahedral is shown in Figure 5.4(d).

In the case that $P_{|\boldsymbol{i}|} < 0$ appears as the value of the edge, face or region control points, the Bezier shape for the determinant of Jacobian determined in Eq. 5.7 is subdivided at this specified location and check the determinant of Jacobian J for the new created tetrahedral regions. If the negative value at the new vertex still remains negative, apply the above procedure to correct the invalidity. This process is continued until all the negative control points values have been corrected. In the case that the movement of the candidate control points fail to move, other local mesh modifications are applied to correct the invalidity. An example where the valid curved mesh is regained by only applying mesh entity shape manipulation is shown in Figure 5.5(b) where some mesh edges on the model faces have been reshaped.

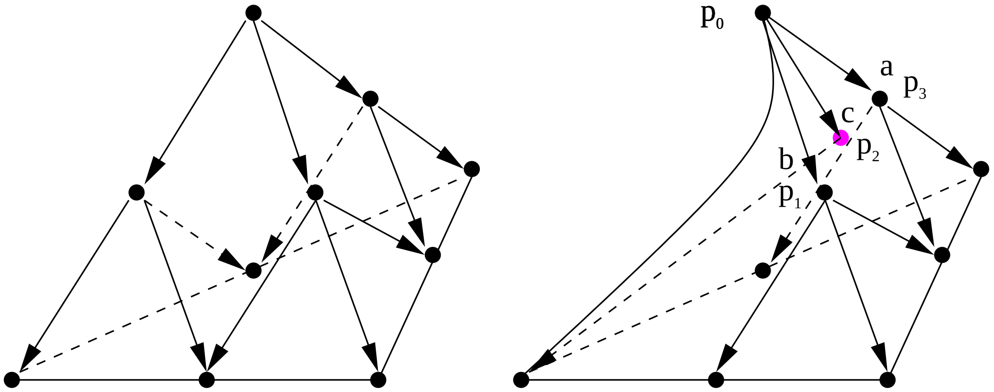
5.5.2 Curved Split Operations

In those cases where two neighboring mesh entities of a single element are on a curved boundary, the curving process will create in the limit angles of 180^0. The only option in this case is to execute a split operation so that a new mesh entity not classified on that same boundary entity is introduced between those two entities. Figure 5.5.2 gives an example in which when the initial linear mesh is curved there are 180^0 angles at the four mesh vertices $M_i^0,\ i = 0...3$ with circles around them in Figure 5.6(b) as well as along the two edges connecting the two pairs of them. In this case introducing edge splits of the edges M_0^1, M_1^1 on the top and bottom of the cylinder eliminate the problem as shown in Figure 5.6(c) where the new created mesh edges are marked as dark. The process of splitting takes advantage of the property that the Bezier geometries maintain their shapes under a subdivision process.

The Bezier split operation is implemented based on the assumption that the original curved entity shape remains unchanged up to any order in parametric space [58]. The benefits for applying split operation are:

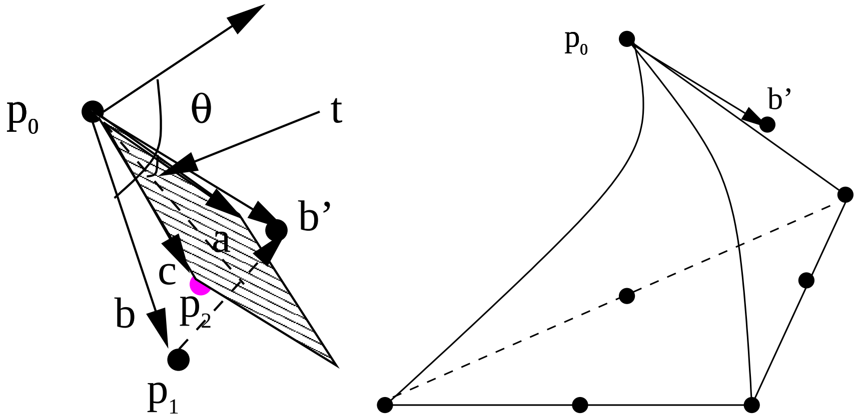
- No topological validity determination required.
- The geometric shapes of the new created mesh entities are well formulated by the subdivision algorithm of Bezier curves, surfaces and regions [58].

For example, when splitting a q order Bezier mesh tetrahedron region at para-



(a) Linear Quadratic Tetrahedral Region

(b) Moving P_1 Causes the Region Invalid because $b \cdot (a \times c) < 0$



(c) Move P_1 to opposite side plane defined by $(a \times c)$

(d) The Valid Tetrahedral Region

Figure 5.4: Correct Invalid Region by Shape Modification

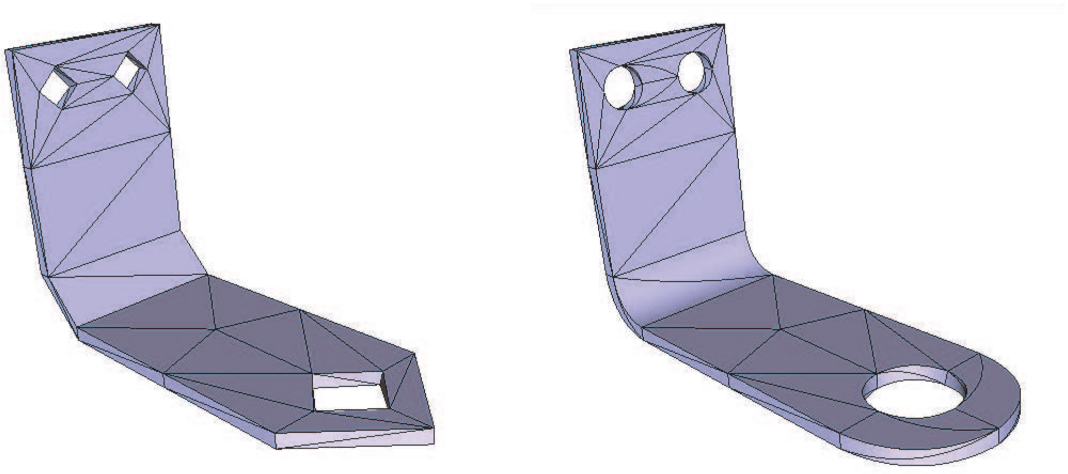
metric location $\xi'(\xi'_1,\ \xi'_2,\ \xi'_3,\ \xi'_4)$, the control points net of the new four regions can be computed as,

$$\hat{b}^0_{(r,j,k,l)} = b_{i_0}(\xi'), \tag{5.10}$$

$$\hat{b}^1_{(j,r,k,l)} = b_{i_1}(\xi'), \tag{5.11}$$

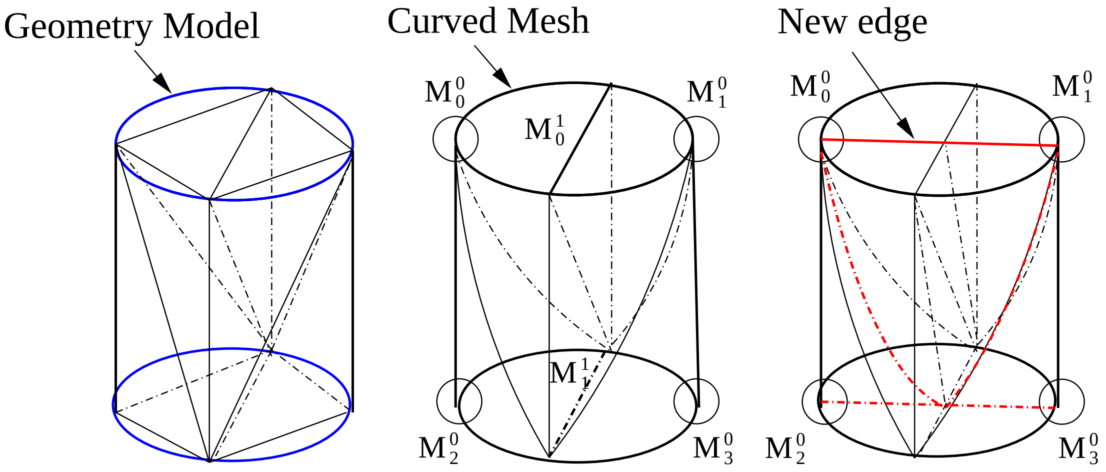
$$\hat{b}^2_{(j,k,r,l)} = b_{i_2}(\xi'), \tag{5.12}$$

$$\hat{b}^3_{(j,k,l,r)} = b_{i_3}(\xi'), \tag{5.13}$$



(a) Straight-side Mesh (b) Curved Mesh by Shape Modification

Figure 5.5: Curved Mesh Generation only by Shape Modification



(a) Straight-side Mesh (b) Invalid Curved Mesh After Curving Because Face Interior Angles are Equal to 180^0 (c) Valid Curved Mesh by Edge Split Operation

Figure 5.6: Split Operation to Correct Model Tangency Problem

where ξ' is volume coordinates and $\xi'_1 + \xi'_2 + \xi'_3 + \xi'_4 = 1$. $i_0 = (0, j, k, l)$, $i_1 = (j, 0, k, l)$, $i_2 = (j, k, 0, l)$ and $i_3 = (j, k, l, 0)$. b_{i_0}, b_{i_1}, b_{i_2} and b_{i_3} are corresponding control points obtained by evaluate the original curved tetrahedron region at ξ' coordinate with respect to i_0, i_1, i_2, i_3.

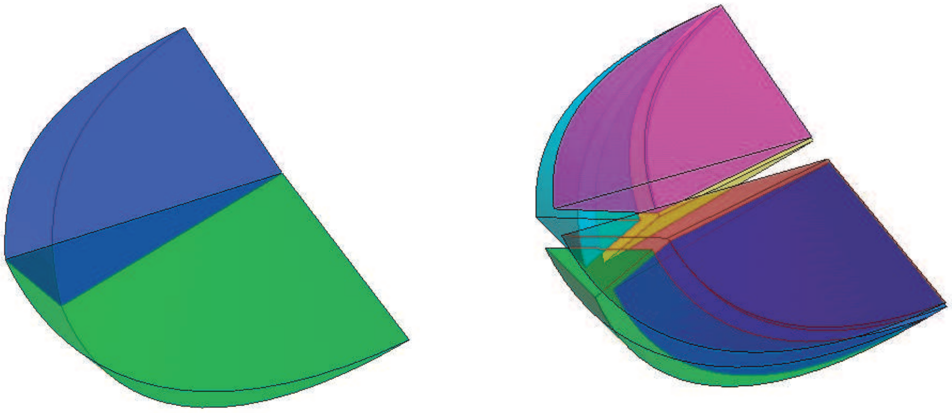
Bezier edge and face split operations are the special cases of region split where one or two parameters of are equal to zero.

For example, a curved face and region split are shown in Figure 5.7(a) - 5.7(d) respectively where the newly created curved regions exactly decompose the shape of the parent regions.

5.5.3 Curved Collapse and Swap Operations

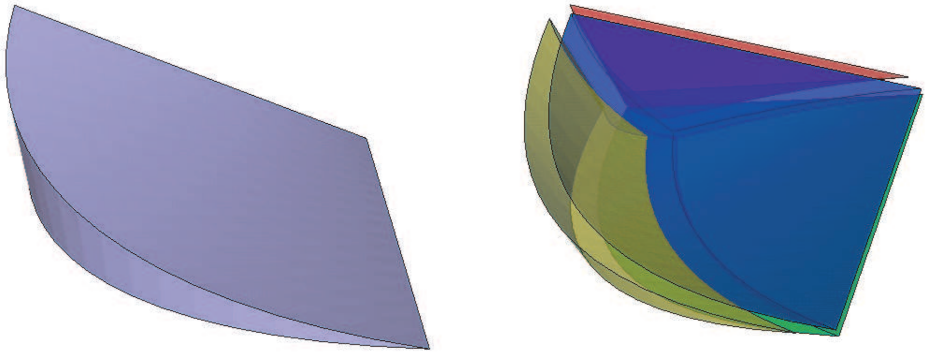
There are cases where the applying only curving and splitting 180^0 angles will not produce a valid or satisfactory mesh result in terms of element shape. This typically occurs where the current mesh configuration is such that there is insufficient room in the desired direction of motion to re-shape mesh entities. The goal of the curved mesh entity collapse and swap operators is to change the mesh topology and geometry to produce sufficient space. Such curved collapse and swap operations are build on straight-sided collapse and swap operations [82] and this thesis only focuses on the curved geometry aspects. Since these operators are computationally more demanding than the others, they are only considered when needed in case that the re-shape operation fails to produce valid or acceptable result mesh. Figure 5.8(a) shows a case where the invalidity caused by curving M_0^1 is effectively corrected by collapsing edge M_1^1 from vertex M_1^0 to M_2^0 (see Figure 5.8(b)). The collapse produces a better mesh configuration than just curving edge M_2^1 shown in 5.8(c) in terms of maintaining better face angles at the vertices.

The primary new complexity in collapse and swap operations introduced by curved mesh geometry is determining appropriate geometric shapes for the new mesh entities created during those operations. Figure 5.9(a) shows the curved domain defined by the five regions connected to vertex M_7^0 that will be deleted when collapsing edge M_0^1 by moving M_7^0 to M_6^0. New edges M_1^1 and M_2^1 shown in Figure 5.9(b) as well as their higher order connected entities need to have proper curved



(a) Original Face Shared by Two Regions

(b) Curved Face Split

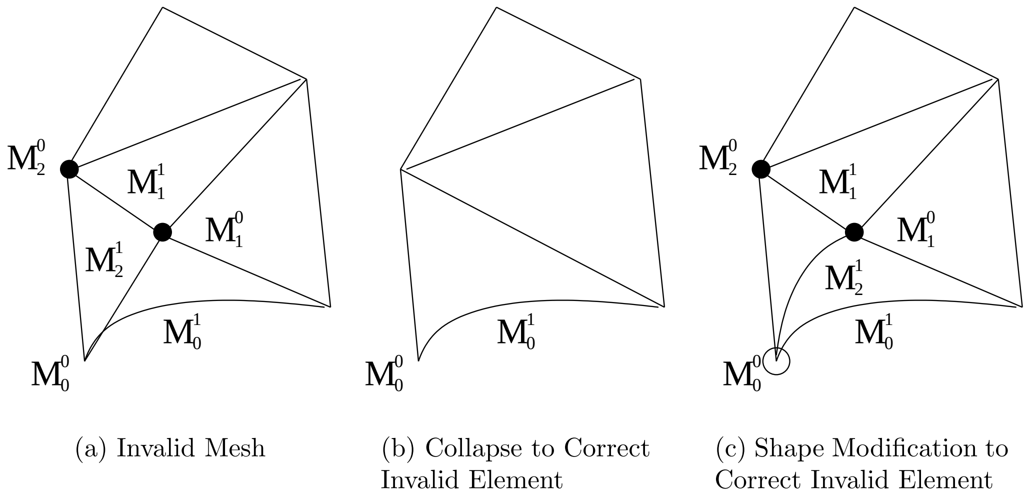


(c) Original Region

(d) Curved Region Split

Figure 5.7: Curved Face and Region Split Operations

shapes assigned to them to make the operation valid. Compatibility between geometric shape and finite element basis indicates the geometric order of these mesh entities should be equal to or less than that used for finite element to satisfy the standard finite element completeness requirement. Thus, the geometric order of a new edge or face can only elevate up to the allowable order. Within the limits of the order of geometry allowed, it is important to determine an appropriate shape for the new mesh entities. For example, if the edge shape of M_2^1 was constructed



(a) Invalid Mesh (b) Collapse to Correct Invalid Element (c) Shape Modification to Correct Invalid Element

Figure 5.8: Edge Collapse to Correct Invalid Element

as straight-sided instead of curved it would intersect the boundary of the affected region which indicates a mesh invalidity. To avoid this problem blending operators [65, 69] are applied to compute the shape for new edges that need high order shapes inside of curved quadrilateral polygons without interfering with others.

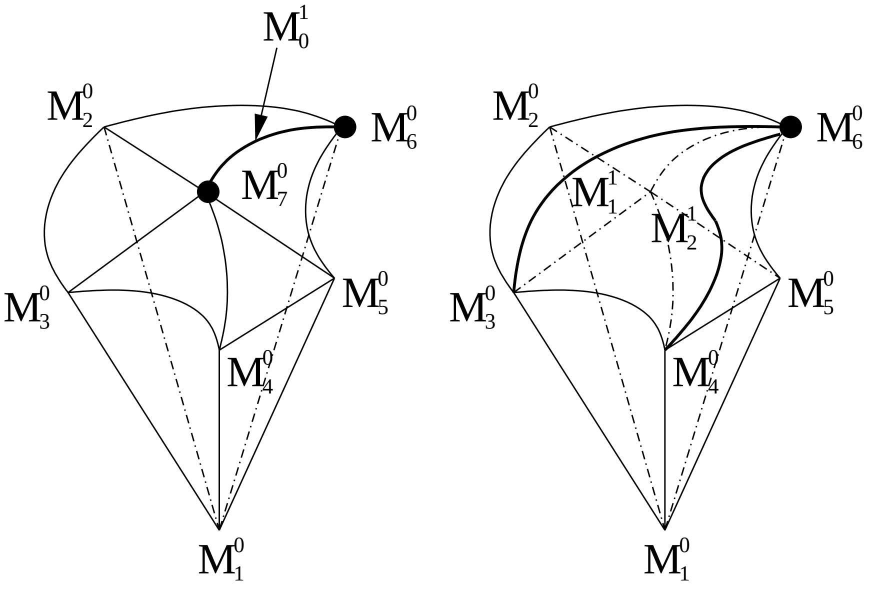
The general blending procedure to compute a curved edge shape inside a quadrilateral polygon is as follows. Let $(S_0,\ S_1,\ S_2,\ S_3)$ be the shapes of the curved quadrilateral polygon and $(P_0,\ P_1,\ P_2,\ P_3)$ are the four vertices. The blending mapping inside the polygon is,

$$x=(1-\nu)S_0+\mu S_1+\nu S_2+(1-\mu)S_3-(1-\mu)(1-\nu)P_0-(1-\nu)\mu P_1-\mu\nu P_2-(1-\mu)\nu P_3. \tag{5.14}$$

where $\mu,\ \nu$ are the corresponding two independent coordinates with $0 \leq \mu \leq 1$ and $0 \leq \nu \leq 1$. If the new diagonal edge is quadratic, substitute $\mu = \nu = 0.5$ into Eq. 5.14 to compute the control point location,

$$x = \frac{1}{2}(S_0 + S_1 + S_2 + S_3) - \frac{1}{4}(P_0 + P_1 + P_2 + P_3). \tag{5.15}$$

If the new edge is cubic, $\mu = \nu = \frac{1}{3}$ and $\mu = \nu = \frac{2}{3}$ are used for the two control points respective shown in Figure 5.10(b). Since simple blend operations do not



(a) Curved Polyhedral for Collapsing M_0^1

(b) Mesh After Collapsing

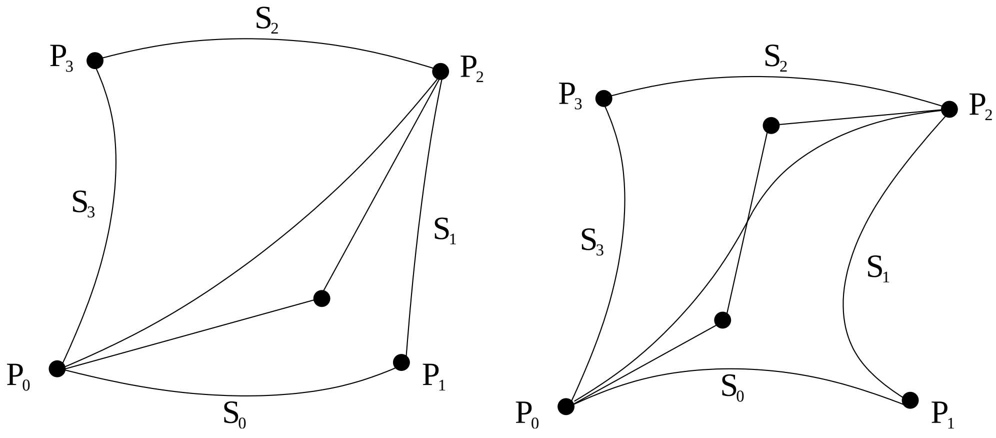
Figure 5.9: Curved Edge Collapse

ensure the generated entities will be contained in the bounding entities, the validity checks for mesh regions as discussed in Section 3.3 must still be performed and the control point locations adjusted as in Section 5.3.1 if needed.

5.5.4 The Incremental Curving Procedure

The incremental procedure developed builds on the linear geometry algorithm of reference [47] with appropriate extensions to account for the added complexity of curved mesh entities. Based on the computational cost and complexity of each operation, the process works in the following five steps:

1. Determine if the invalidity is caused by pairs of neighboring mesh faces or edges classified on the boundary such that angles of 180^0 or greater are created. In these cases apply a split operation that will introduce additional mesh entities to subdivide those angles see Section 5.5.2.
2. Check whether there is adequate room on the "other side" of the intersected mesh entity, its geometric order is sufficient to re-shape and the resulting



(a) Quadratic Diagonal Edge (b) Cubic Diagonal Edge

Figure 5.10: Blending Method to Compute the New Edge Shape

curved mesh shape is acceptable. If these conditions are met, apply mesh entity re-shape to fix the problem see Section 5.5.1.

3. Examine the possibility of applying a collapse operation to eliminate the invalidity. Apply the collapse if possible see Section 5.5.3.
4. Examine the application of a swap operation to create the space required to have appropriately shaped mesh entities in the swapped configuration to eliminate the invalidity see Section 5.5.3.
5. If none of the above operations is successful, the mesh entity can not be curved to the model boundary. The process keeps the original shape of the mesh entity and applies local refinement on the mesh entity [82]. All new mesh entities classified on the boundary into the list of entities to be processed. Subdivision creates more options for applying other operations.

One important issue in the above process is the shape quality of the curved mesh entity. The quality metrics are important when choosing which mesh modification or configuration to apply to correct the invalid elements. For example, the choice of the edge collapse operation in Figure 5.8(b) over the re-shape operation in Figure 5.8(c). Some of the currently existing quality measures, such as interior determinant of Jacobian variation [45], are proposed only based on element geometric shape without any direct relation to the solution accuracy. Under the assumption that a future adaptive procedure will consider altering element interior shape to optimize accuracy, the current procedure presented is focused only on ensuring a positive Jacobian throughout the element domain and acceptable user specified face angles.

Although the last operation does typically allow completion of the process, it does introduce undesired mesh refinement (see Figure 5.11(c) in the examples section). In such cases an alternative is to inflate the order of the geometry of the mesh entity to provide additional shape freedom without the need to add additional elements. As shown in Figure 5.11(d) this approach can be successful in obtaining a valid mesh without the need of refining the mesh. Of course, inflating the geometry order can force the need to inflate the order of finite element used at that location past what may have been required for analysis accuracy. However, this is likely to be more cost effective than the additional cost introduced by the refinement process.

5.6 Meshing Results

Curvilinear mesh examples are presented in this section to demonstrate the capability of the procedure to automatically generate the p-version meshes considering geometric approximation, mesh gradation and thin section structure together from the beginning.

5.6.1 Example 1

The first example is a biomechanical model used to simulate blood flow. Linear, quadratic, cubic and quartic meshes are shown in Figure 5.11(b)- 5.11(e). The quality of curved mesh with respect to geometry model is measured by Δd_{min} which are the normalized maximum distance deviation Δd to the shortest linear mesh edge length. Δd is computed as the maximum distance between sample points of each mesh entity classified on curved model boundary and their corresponding closest points on the boundary. Statistics for curving the blood vessel meshes are presented in Table 5.1. The results clearly demonstrate that geometric approximation error in terms of normalized maximum distance deviation has been improved by increasing the geometric approximation order especially in going from linear to cubic geometry.

Table 5.1: Statistics for Curving the Blood Vessel Model

	Linear	Quadratic	Cubic	Quartic
Δd_{min}	1.7028	0.9897	0.5006	0.4648
Collapse		2	1	1
Collapse		3	1	1
Split		17	5	5
Recurving		29	15	14
Refinement		10	3	2

Table 5.1 also includes the statistics on which local mesh modifications were used to correct the invalid elements. The shape manipulation operations are the most frequently applied in this example. The number of refinements required has been reduced from 10 for quadratic shaped elements to 3 for cubic shaped elements and 2 for quartic shaped elements, thus demonstrating the benefit of using high geometric order to alleviate the need for local refinement to fix invalidity.

5.6.2 Examples 2, 3

The next two examples are models with singular model edges as shown in Figure 5.12 that the linear graded meshes are generated by the procedure discussed in Chapter 4. The straight-sided meshes are curved with quadratic geometry shape without considering gradation (top) and with considering gradation (bottom) by using the curving procedures discussed above. In the curving procedure without considering graded mesh, the gradation of the meshes around the singularities are destroyed when local mesh modifications are applied to correct the invalid elements. The number of elements for the meshes generated by the two different approaches is compared in Table 5.2. For the left model, the number of mesh entities does not change significantly because there are only two curved singular edges and each of them with only one mesh edge classified on. For the right model, the number of regions has been reduced 15% by applying the curving procedure considering gradation.

Table 5.2: **Statistics for Curving Models with Singular Edges without/with Considering Gradation**

	Left Model		Right Model	
	Mesh a	Mesh c	Mesh b	Mesh d
Regions	881	863	1923	1643
Faces	1911	1881	3456	3356
Edges	1265	1254	3356	2980
Vertices	236	237	467	406

5.6.3 Examples 4, 5

These two examples are models with thin sections given in Figure 5.13. The first row shows the input surface triangulation for the two models. The second row shows just the isolated opposite thin section surface patches after loop construction. As can be seen from the figures the two surface meshes are not yet matched. The bottom row shows the final meshes where the thin section meshes have been matched, the prismatic thin section elements created and volume mesh completed and curved.

The statistics for these two meshes are compared with the all tetrahedral meshes generated by the isotropic volume mesh in Table 5.3. The number of elements has been reduced by almost 20% for the left model and by more than 40% for the right model.

Table 5.3: **Statistics for Curved Models with Thin Sections**

	Left Model		Right Model	
	All Tet Mesh	Mixed Mesh	All Tet Mesh	Mixed Mesh
Regions	492	413	787	463
Faces	1342	975	2345	1745
Edges	1045	732	2643	2034
Vertices	184	170	467	436

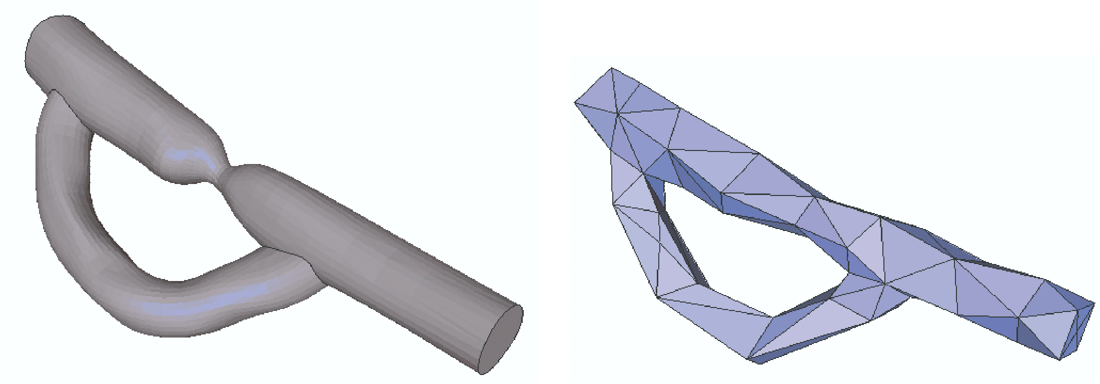
5.6.4 Examples 6, 7

The next two examples are models with singular edges and thin sections together shown in Figure 5.14 and Figure 5.16 respectively. Three different types of meshes, curved isotropic volume mesh (type a), curved tetrahedral mesh without considering thin sections and gradation control (type b), and curved mixed mesh with considering gradation control and thin sections (type c), are compared in Table 5.4. The statistic shows that the curved mixed mesh generated by applying the automatic p-version mesh procedure for the examples in Figure 5.14 produces 1/2 the elements of the type b mesh which does not account for the thin section or gradation control. Figure 5.15 shows the close-up of mesh around the singular edge for type b and type c meshes. The gradation for the type c mesh is maintained through the curving procedure with gradation control. Even in the example given in Figure 5.16 where the thin section represents a small portion of the domain, the type c mesh generated a mesh with 240 fewer elements, which is a 24% reduction in the number of elements than the type b mesh.

Table 5.4: Statistics for Curved Models with Singular Edges and Thin Sections

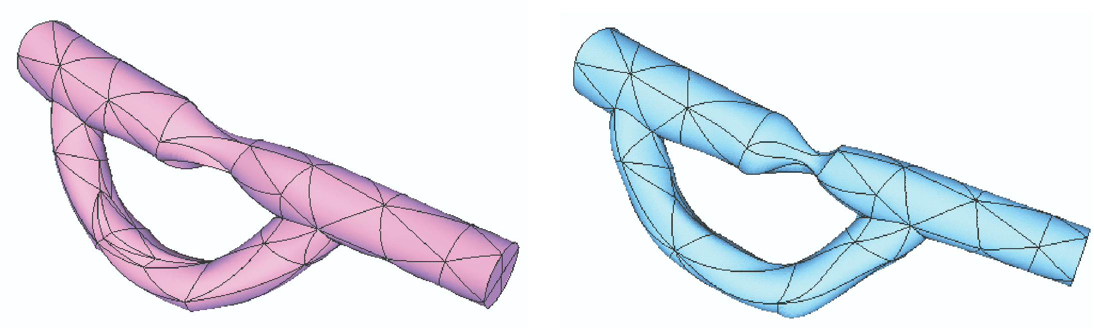
	Model in Figure 5.14		Model in Figure 5.16	
Meshes	Regions	Gradation	Regions	Gradation
Type a	306	0.0	260	0.0
Type b	705	0.15	996	0.15
Type c	350	0.15	756	0.15

More p-version meshes can be found in Chapter 8 of the p-version analysis results.



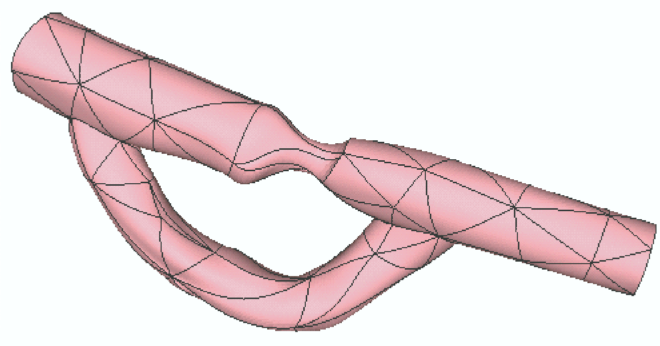
(a) Geometry Model

(b) Linear Mesh



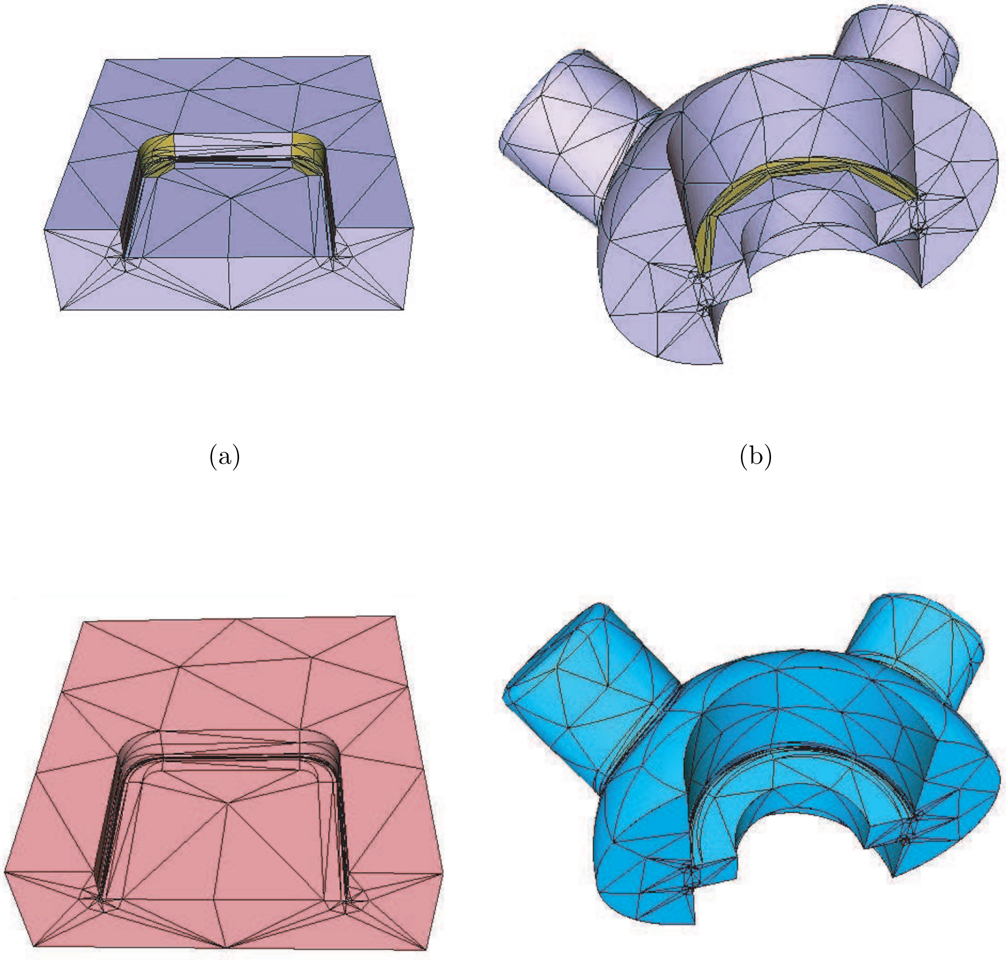
(c) Curved Quadratic Mesh

(d) Curved Cubic Mesh



(e) Curved Quartic Mesh

Figure 5.11: Blood Vessel Meshes



(a) (b)

(c) (d)

Figure 5.12: Curved Meshes for Models with Singularities: Curving without Considering Gradation (top). Curving with Considering Gradation (bottom).

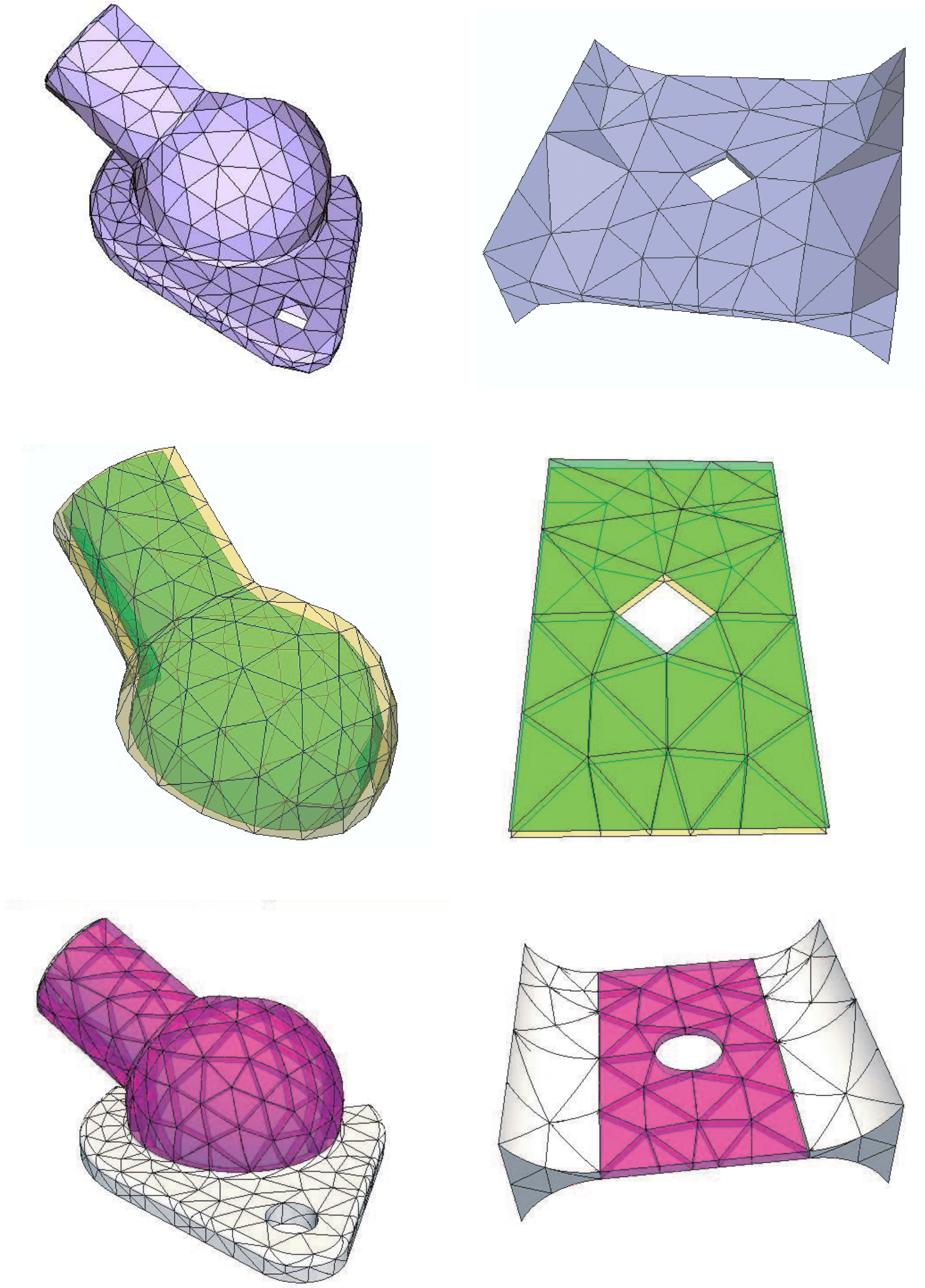


Figure 5.13: Examples Meshes for Models with Thin Sections: Surface Triangulations(top). Thin Sections(middle). Complete Volume Mesh with Prismatic Elements(bottom).

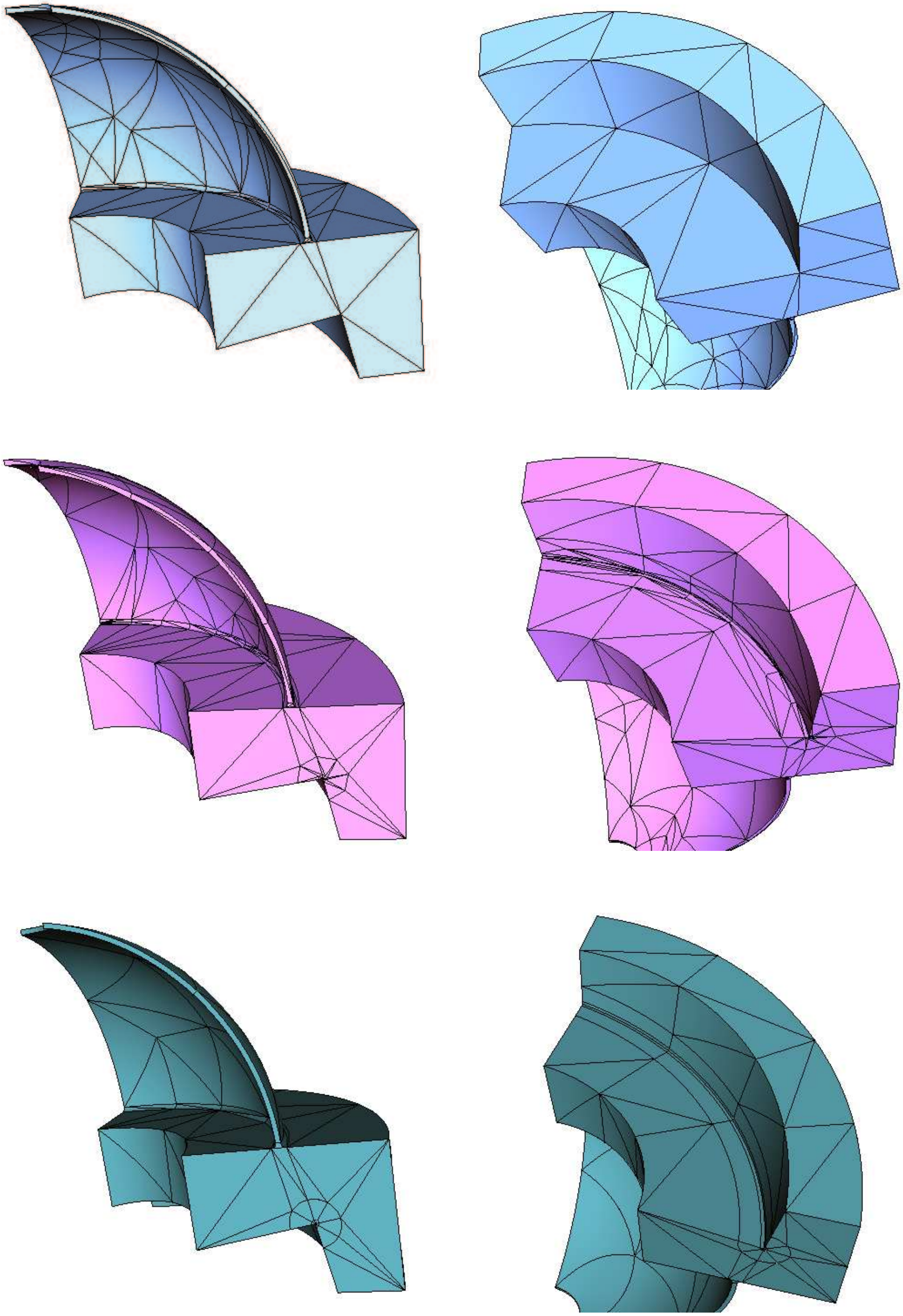


Figure 5.14: Examples Meshes for Models with Singular Edges and Thin Sections: Isotropic Volume Mesh (top). Curved Mesh without Considering Gradation and Thin Sections (middle). Curved Mixed Mesh with Considering Gradation and Thin Sections (bottom).

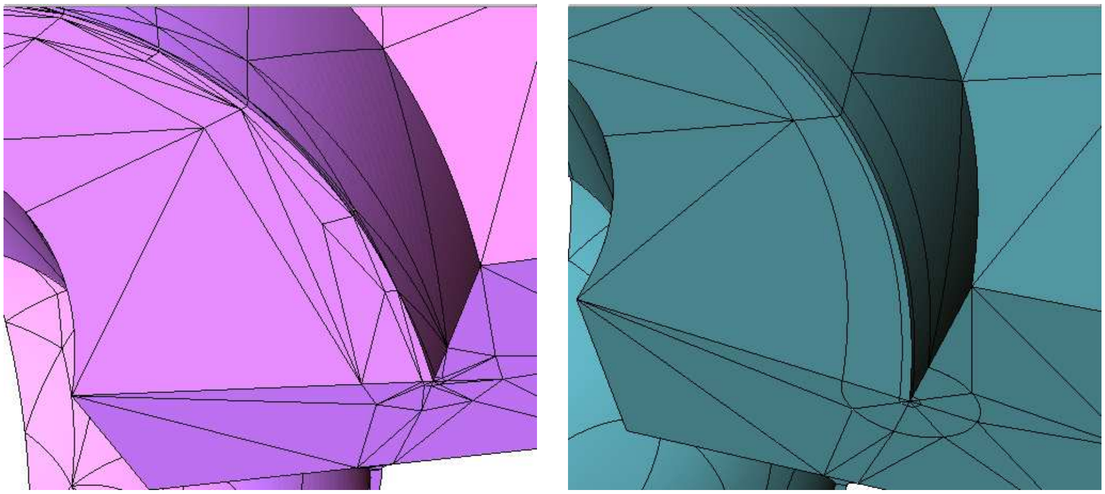


Figure 5.15: Close-up of the Meshes Around Singular Edge for Curved Mesh without (Left) and with (Right) Considering Gradation.

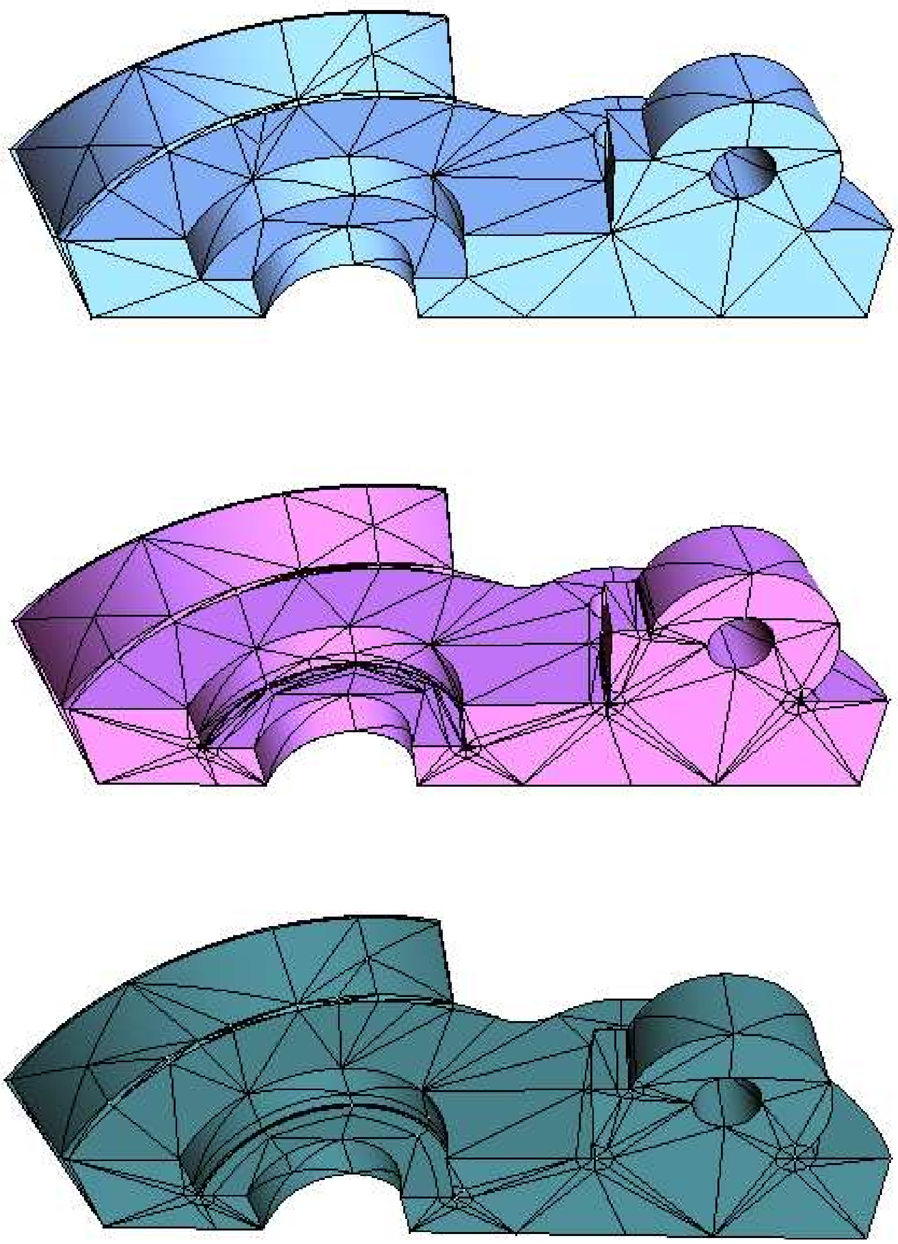


Figure 5.16: Examples Meshes for Models with Singular Edges and Thin Sections: Isotropic Volume Mesh (top). Curved Mesh without Considering Gradation and Thin Sections (middle). Curved Mixed Mesh with Considering Gradation and Thin Sections (bottom).

Part III

An Automated Adaptive Directional Variable p-Version Analysis

CHAPTER 6
Flexible Topology Based Directional Variable p-Version Approximation

The p-version method requires a flexible data structure to deal with the situation when p-orders varies from element to element. This chapter first reviews the data structures for variable p-order approximation. Efforts are focused on the necessary modifications that enable the mesh faces and regions to have different directional polynomial orders. The implementation of the directional variable p-version approximation in Trellis is then discussed.

6.1 Overview of the Variable p-Order Approximation Data Structure

Most of the variable p-version approximation data structures specify the polynomial order p based on element level Ω^e then construct the high order shape functions for the closure topological mesh entities of $\bar{\Omega}^e$ [44, 124]. The common method [44] used for such type of structures to deal with neighboring elements of different order is to add and/or delete hierarchic shape functions on the mesh entity shared by the elements such that C^0 continuity is maintained. A typical example is shown in Figure 6.1 that two elements M_0^2 and M_1^2 are of the linear and cubic order respectively. The solid dark circle indicates the shape functions for linear element M_0^2 and the non-filled circle indicates the shape functions for the cubic element M_1^2. Either deleting the high order shape functions along mesh edge M_0^1 for M_1^2 (left) or adding shape functions along mesh edge M_0^1 for M_0^2 (right) to maintain the approximation continuity along mesh edge M_0^1 is required.

Another method to deal with the p-orthotropy is the generalized finite element method (GFEM) (so called "hp-clouds" method) [19, 21, 51-55, 95]. In the GFEM method, the domain is partitioned with a collection of open sets Ω_e , $\bar{\Omega} \subset \cup_{e=1}^{n_{el}} \Omega_e$. The sets Ω_e are so called clouds and a set of basis functions φ_e which form a partition

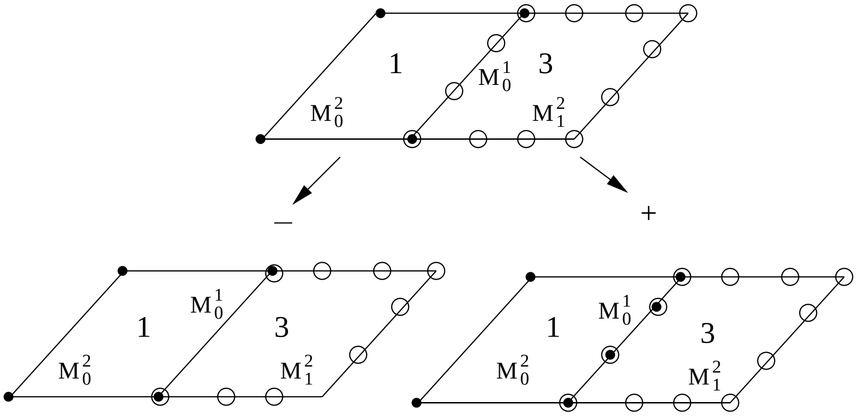


Figure 6.1: Continuity by Hierarchical Shape Functions

of unity on Ω are constructed on the clouds [95],

$$\sum_{e=1}^{n_{el}} \varphi_e(\boldsymbol{x}) = 1, \qquad \boldsymbol{x} \in \bar{\Omega}. \tag{6.1}$$

The typical hp-version general finite element shape functions are defined as [95],

$$\varphi_{ai} = N_a L_i, \tag{6.2}$$

where N_a is the conventional standard finite element shape function and L_i is a polynomial of degree p. Since the shape functions form a partition unity (PU) [95],

$$\sum_a \varphi_{ai} = \sum_a N_a L_i = L_i \sum_a N_a = L_i. \tag{6.3}$$

Therefore, the polynomials L_i can be recovered through linear combinations of the cloud basis functions φ_{ai} that allow the assignment of different polynomial order of L_i. This new hierarchic basis shape functions concentrate all the unknown of degree of freedoms at the vertices of the elements and independent of the element topology. However, the integration of the stiffness matrix can be considerable more expensive than traditional finite element method, depending on the choice of the partition of unity [54].

This thesis applies the topology based variable p-version approximation data structure [115] that easily supports the general specification of shape functions over the mesh entities with the full set of shape functions determined by the order assigned to the mesh entities of the closure of the element. The hierarchic shape function for each topological entity is uniquely constructed in the form of Eq. 2.16 for the specified polynomial order. The data structure can easily satisfy the C^0 continuity requirement [115]. As an example, Figure 6.2 (left) shows a p-order variation for the edges in a two dimension mesh. Variation on the p-order of the independent coordinate of the mesh faces and regions is needed in this thesis (see Section 3.1.3) and has thoes been added. For example, the p-order is 6 for ξ_1 and 4 for ξ_2 as in Figure 6.2 (right) for the mesh face M_0^2.

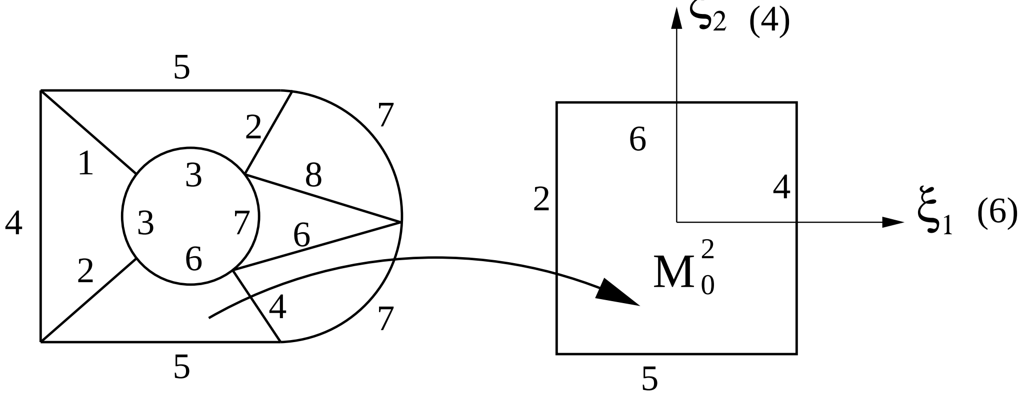


Figure 6.2: Variable p-Order Mesh for Mesh Edge and Faces in Two Dimension

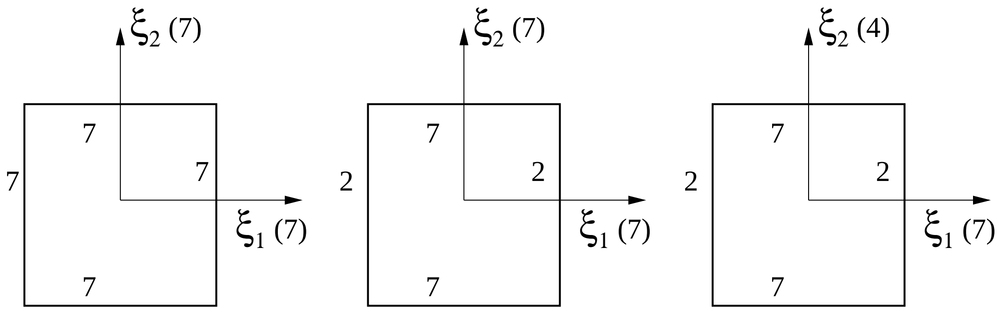
6.2 Directional p-Order Approximation

Although the basic convergence rate in Eq. 2.55 would indicate no obvious advantage of different p-order in each direction for the mesh faces and regions, it is clear that the computational efficiency in terms of the number of degree of freedom and computational cost can be improved [1, 57]. The example in Figure 6.3 demonstrates the influence of the directional variable p-order on the number of shape functions for a quadrilateral face, where the integers indicate the assigned polynomial order for each mesh edge and the independent coordinates of the quadrilateral face:

Case a Uniform p-order for the mesh edges and face.

Case b Variable p-order for the mesh edges and uniform p-order for the independent coordinates of the mesh face.

Case c Variable p-order for the mesh edges and the independent coordinates of the mesh face.



(a) Case a (b) Case b (c) Case c

Figure 6.3: Uniform, Variable and Directional Variable p-Order for a Quadrilateral Face

Let n denote the number of shape functions for the quadrilateral face shown in Figure 6.3 and can be computed as:

Case a: $$n = \underbrace{4}_{\text{Vertices}} + \underbrace{4 \times (7-1)}_{\text{Edges}} + \underbrace{1+2+3+4}_{\text{Face}} = 38,$$

Case b: $$n = \underbrace{4}_{\text{Vertices}} + \underbrace{1+1+6+6}_{\text{Edges}} + \underbrace{1+2+3+4}_{\text{Face}} = 29,$$

Case c: $$n = \underbrace{4}_{\text{Vertices}} + \underbrace{1+1+6+6}_{\text{Edges}} + \underbrace{1+1+1+1}_{\text{Face}} = 22.$$

The example shows that the number of shape functions can be reduced from 38 to 29 if variable p-order is applied and the number can be further reduced down to 22 if directional p-order is applied for the independent coordinates of the face. Thus the number of degrees of freedom can be reduced close to 1/2 for a case with strong gradients in one direction and weak ones in the other. The influence of this ability to reduce the number of degree of freedom, thus reducing the computational cost is demonstrated by examples in the chapter 8. This section presents the modification of the topology variable p-order data structure [115] to allow the p-order variation at the independent coordinates of mesh faces and regions as in case c.

Based on the construction of topology based hierarchic shape function presented in Eq. 2.16, the blending function for a mesh entity $M_j^{d_j}$ remains the same once the topology of $M_j^{d_j}$ and $M_e^{d_e}$ are determined. Therefore it is the entity shape function $\phi(M_j^{d_j})$ to complete the polynomial order requirement of the shape function $N(M_j^{d_j})$. Let p_i be the requested polynomial order for the independent coordinate ξ_i of a mesh entity $M_j^{d_j}$ and q_i is the polynomial order of the blending function used for $\psi(M_j^{d_j})$. Thus, the polynomial order of $\phi(M_j^{d_j})$ for an entity is given by $p_i - q_i$ at each direction. Considering there is only one parametric coordinate for a mesh edge, no modification is needed for a *pth* order mesh edge function $\phi(M_j^1)$ that remains the same as in [115],

$$\phi(M_1^1) = \frac{2}{\xi^2 - 1}\sqrt{\frac{2p-1}{2}} \int_{-1}^{\xi} P_{p-1}(t)dt, \quad p \geq 2, \quad -1 \leq \xi \leq 1, \tag{6.4}$$

where P_n are the Legendre polynomials [8].

6.2.1 Triangle Mesh Face

For a triangle mesh face with barycentric coordinate (ξ_1, ξ_2, ξ_3), $\xi_1 + \xi_2 + \xi_3 = 1$, the blending shape function $\psi(M_j^2, M_e^{d_e})$ is of linear order in each coordinate ξ_i [115]. Here, the directional p-order structure only varies the polynomial order on two of the independent coordinates. Thus, the polynomial order for the mesh entity shape function $\phi(M_j^2)$ is $(p_i - 2)$ for each coordinate ξ_i respectively and the modified set of triangle face functions based on Legendre polynomials is given as,

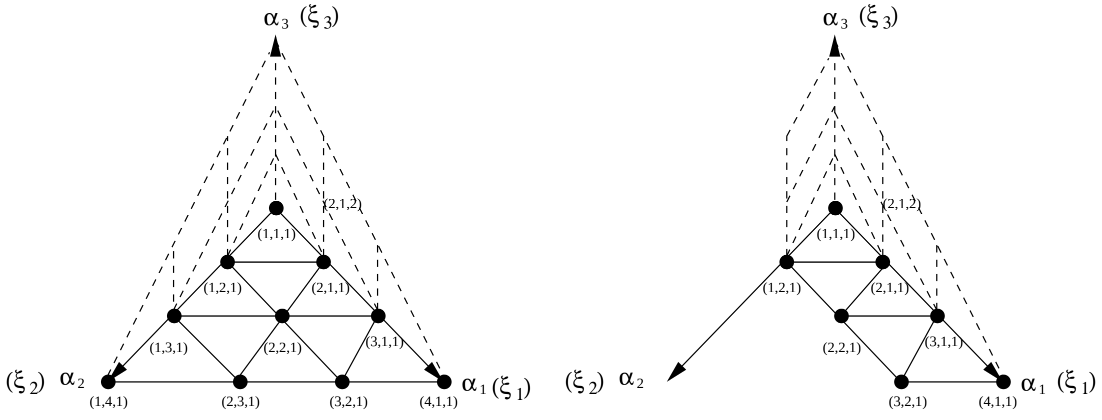
$$\phi(M_j^2) = P_{\alpha_1 - 1}(\xi_2 - \xi_1) P_{\alpha_2 - 1}(2\xi_3 - 1), \tag{6.5}$$

$$\begin{gathered} 1 \ \leq \ \alpha_i \ \leq \ p_i - 2, \\ \Sigma\alpha_i \leq max(p_i) - 1, \ i = 1, 2, \end{gathered} \tag{6.6}$$

where indices α_i indicates the directional behavior for the two independent coordinates ξ_i and defines the highest power of ξ_i in the resulting mesh entity shape functions ϕ. It is clear that the formulation of the hierarchic shape functions for the directional p-order are the same as those in [115] and are uniquely determined by Eq. 6.5. It is the modified Eq. 6.6 that controls the total number of the shape functions to achieve the variable p-orders requirements at the independent coordinates. As an example, Figure 6.4 demonstrates the difference between the number of triangle face entity shape functions $\phi(M_j^2)$ for the shape function N having uniform p-order 6 and directional p-order (6, 4) respectively for the independent coordinate pair $(\xi_1,\ \xi_2)$. Here, the indices $(\alpha_1, \alpha_2, \alpha_3)$ satisfy $\alpha_1+\alpha_2+\alpha_3 = p,\ p = 3, ..., 6$ for uniform p-order up to order 6 shown in Figure 6.4(a) and the indices in Figure 6.4(b) satisfy Eq. 6.6 as,

$$1 \leq \alpha_1 \leq 4, \quad 1 \leq \alpha_2 \leq 2, \quad \alpha_1 + \alpha_2 \leq 5, \tag{6.7}$$

that only include the terms needed up to 6th order in ξ_1 direction and 4th order in ξ_2 direction in the resulting shape function $N(M_j^2)$.



(a) Uniform p-Order

(b) Directional p-Order

Figure 6.4: Triangle Face Entity Shape Function $\phi(M_j^2)$ for Uniform and Directional p-Order

The symmetric counterpart for the independent coordinate pairs $(\xi_1,\ \xi_3)$ and $(\xi_2,\ \xi_3)$ shown as the dash lines in Figure 6.4 can be obtained as a linear combination of the existing modes [115], where the dependent coordinate picks up the higher polynomial order of the independent coordinates $(\xi_1,\ \xi_2)$. For example, $(2, 1, 2)$ for a 5th order polynomial can be represented by the combination of $(2, 2, 1)$ and $(3, 1, 1)$ as follows,

$$\begin{aligned}(2,1,2) &= \xi_1\xi_3 - \frac{1}{4}(\xi_1+\xi_3) + \frac{1}{14} = \xi_1(1-\xi_1-\xi_2) - \frac{1}{4}(\xi_1+1-\xi_1-\xi_2) + \frac{1}{14} \\ &= \xi_1 - \xi_1^2 - \xi_1\xi_2 + \frac{1}{4}\xi_2 - \frac{5}{28} = -(\xi_1^2 - \frac{3}{4}\xi_1 + \frac{3}{28} + \xi_1\xi_2 - \frac{1}{4}(\xi_1+\xi_2) + \frac{1}{14}) \\ &= -((2,2,1)+(3,1,1)). \end{aligned} \tag{6.8}$$

6.2.2 Quadrilateral Mesh Face

For a quadrilateral mesh face with independent coordinates $(\xi_1,\ \xi_2)$, the blending shape function $\psi(M_j^2, M_e^{d_e})$ is of quadratic order in each coordinate ξ_i [115]. Thus, the polynomial order for the mesh entity shape function $\phi(M_j^2)$ is $(p_i - 2)$ for each orthogonal coordinate ξ_i respectively and the modified set of quadrilateral face functions based on Legendre polynomials as,

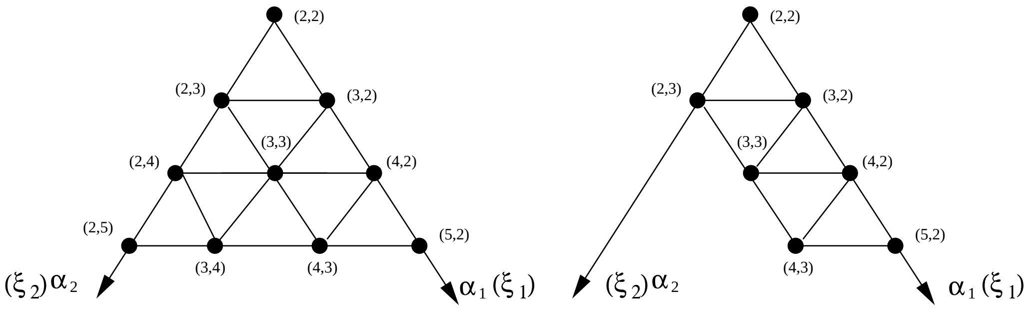
$$\frac{1}{4}(\xi_1^2-1)((\xi_2^2-1)\phi(M_j^2) = \sqrt{\frac{2\alpha_1-1}{2}}\int_{-1}^{\xi_1} P_{\alpha_1-1}(t)dt\sqrt{\frac{2\alpha_2-1}{2}}\int_{-1}^{\xi_2} P_{\alpha_2-1}(t)dt, \tag{6.9}$$

$$\begin{gathered} 2 \ \leq \ \alpha_i \ \leq \ p_i - 2, \\ \Sigma\alpha_i \leq max(p_i). \end{gathered} \tag{6.10}$$

The term $\frac{1}{4}(\xi_1^2-1)((\xi_2^2-1)$ is the blending shape function that contributes to the resulting shape function N together with the mesh entity function $\phi(M_j^2)$. Again, indices α_i indicates the directional behavior for the two independent coordinates ξ_i and defines the highest power of ξ_i in the resulting mesh entity shape function ϕ. The modified Eq. 6.10 determines the number of face modes to complete the requested polynomial order p_i for each independent coordinate. As an example, the mesh entity shape functions $\phi(M_j^2)$ for a quadrilateral mesh face with uniform p-order 7 and directional p-order (7, 5) are shown in Figure 6.5.

6.2.3 Tetrahedral Mesh Region

For a tetrahedral mesh region with volume coordinate $(\xi_1, \xi_2, \xi_3, \xi_4)$, $\xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$, the blending shape function $\psi(M_j^3, M_e^{d_e})$ is of linear order in each coordinate ξ_i [115]. Here, the directional p-order structure only varies the polynomial order on three of the independent coordinates. Thus, the polynomial order for the



(a) Uniform p-Order

(b) Directional p-Order

Figure 6.5: Quadrilateral Face Entity Shape Function $\phi(M_j^2)$ for Uniform and Directional p-Order

mesh entity shape function $\phi(M_j^3)$ is $(p_i - 3)$ for each coordinate ξ_i respectively and the modified set of tetrahedral region functions based on Legendre polynomials is as,

$$\phi(M_j^3) = P_{\alpha_1 - 1}(\xi_2 - \xi_1) P_{\alpha_2 - 1}(2\xi_3 - 1) P_{\alpha_3 - 1}(2\xi_4 - 1), \tag{6.11}$$

$$\begin{gathered} 1 \ \leq \ \alpha_i \ \leq \ p_i - 3, \\ \Sigma\alpha_i \leq max(p_i) - 1, \ i = 1, 2, 3. \end{gathered} \tag{6.12}$$

The modified Eq. 6.12 determines the number of region shape functions needed for the directional variable p-orders.

6.2.4 Hexahedral Mesh Region

The hexahedral mesh region shape functions are constructed based on the tensor product polynomial basis using the Legendre polynomials [115] as,

$$\phi(M_j^3) = \sqrt{\frac{2\alpha_1 - 1}{2}} \int_{-1}^{\xi_1} P_{\alpha_1 - 1}(t) dt \sqrt{\frac{2\alpha_2 - 1}{2}} \int_{-1}^{\xi_2} P_{\alpha_2 - 1}(t) dt \sqrt{\frac{2\alpha_3 - 1}{2}} \int_{-1}^{\xi_3} P_{\alpha_3 - 1}(t) dt, \tag{6.13}$$

where the indices α_i has been modified as follows to handle the directional variable p-orders,

$$\begin{gathered} 2 \ \leq \ \alpha_i \ \leq \ p_i - 2, \\ \Sigma\alpha_i \leq max(p_i), \ i = 1, 2, 3. \end{gathered} \tag{6.14}$$

6.2.5 Prism Mesh Region

For a prism mesh region defined in the independent coordinates $(\xi_1, \xi_2, \xi_3, \xi_4)$, $\xi_1 + \xi_2 + \xi_3 = 1$, the mesh region shape functions are products of the triangle face functions and the one dimensional edge function along ξ_4 given by [115],

$$\phi(M_j^3) = A(\xi_1,\ \xi_2,\ \xi_3) \times B(\xi_4), \tag{6.15}$$

$$A(\xi_1,\ \xi_2,\ \xi_3) = \phi(M_j^2), \tag{6.16}$$

$$B(\xi_4) = \phi(M_j^1), \tag{6.17}$$

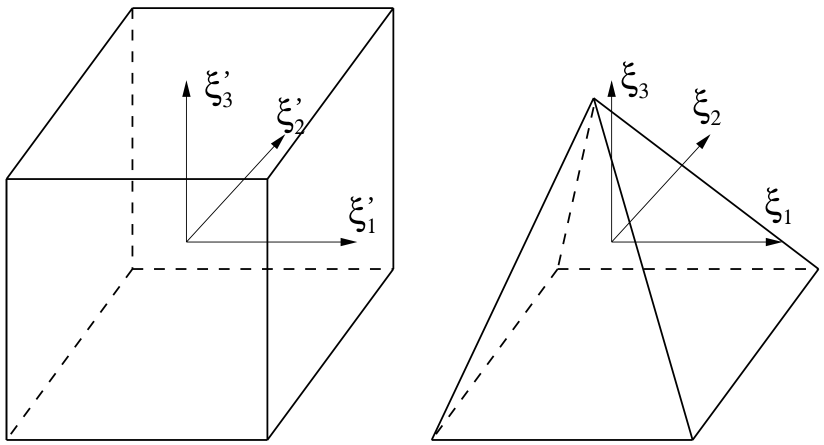
where $\phi(M_j^2)$ and $\phi(M_j^1)$ are the entity shape functions for triangle face shown in Eq. 6.5 and mesh edge shown in Eq. 6.4 [115]. The indices of $(\alpha_1, \alpha_2, \alpha_4)$ for the three independent coordinates (ξ_1, ξ_2, ξ_4) in case of directional p-order are,

$$\begin{gathered} 1 \ \leq \ \alpha_i \ \leq \ p_i - 3, \\ \Sigma\alpha_i \leq max(p_i),\ i = 1, 2, \\ 2 \ \leq \ \alpha_4 \ \leq \ p_4 - 2. \end{gathered} \tag{6.18}$$

The orthogonality between the coordinate ξ_4 and the barycentric coordinate (ξ_1, ξ_2, ξ_3) allows the assignment of variable p-orders for the prism mesh region.

6.2.6 Pyramid Mesh Region

The Pyramid mesh region acts as a buffer element to connect the four sided faces of hexahedral or prism dominated structured mesh regions to the tetrahedral element dominated unstructured mesh regions. The p-order hierarchic shape functions for pyramid is more complicated than the other topology elements because of the lack of the natural coordinates and the degeneracy of the apex [73, 134]. This thesis applies the degenerating procedure presented in [115] to construct the shape functions for a pyramid element that collapses the vertices of the top face of a hexahedral element to the center of the top face shown in Figure 6.6. The mapping functions between the coordinates (ξ_1', ξ_2', ξ_3') for hexahedral element and the coordinates (ξ_1, ξ_2, ξ_3) for pyramid element is given as [115],



(a) Coordinates for Hexahedral (b) Coordinates for Pyramid

Figure 6.6: The Mapping between Hexahedral and Pyramid

$$
\begin{aligned}
\xi_1^{'} &= \frac{2\xi_1}{1-\xi_3}, \\
\xi_2^{'} &= \frac{2\xi_2}{1-\xi_3}, \\
\xi_3^{'} &= \xi_3.
\end{aligned}
\tag{6.20}
$$

Substitute the mapping function in Eq. 6.20 into the hexahedral region shape function in Eq. 6.13 yield the region shape function for pyramid element. Thus, the indices α_i for a directional p-order pyramid is the same as that for a hexahedral element shown in Eq. 6.14.

6.3 The Implementation of Directional Variable p-Order Approximation in Trellis

Trellis is an object oriented software implemented for reliable solving problems in mathematical physics [31]. Its input consists of a geometric model with appropriate physical attributes. The geometry, physics, mathematical model, discretization and numerical methods are decomposed into interacting classes that allow new versions and forms of each of the components to be introduced [31]. In addition, Trellis is geometric-based designed that support the directly retrieving of CAD model information for the need of adaptivity.

The implementation of directional variable p-order approximation in Trellis focused on the system contributors for element stiffness matrices contributor ($\boldsymbol{k}^e$) and element force vector ($\boldsymbol{f}^e$) of an element Ω^e that require the evaluation of $N_a(\boldsymbol{\xi})$ and $N_{a,\boldsymbol{\xi}}(\boldsymbol{\xi})$ at a specified integration point $\boldsymbol{\xi}$.

Let $\boldsymbol{p}$ describes the desired p-orders for the independent coordinates of the mesh entity $M_j^{d_j}$, $M_j^{d_j} \in \bar{\Omega}^e$, $d_j = 0,\ 1,\ 2,\ 3$. The dimension of the vector $\boldsymbol{p}$ equals to the entity dimension, $dim(\boldsymbol{p}) = d_j$. The number of shape functions N_a is determined by the polynomial order vector $\boldsymbol{p}$ related to the closure mesh entities $M_j^{d_j}$ of the element Ω^e. A data structure to manage the polynomial order vector $\boldsymbol{p}$ with respect to any combination of the mesh entity, mesh entity independent coordinates and applied physical field is needed. This requirement is accomplished by attaching the vector $\boldsymbol{p}$ directly to the mesh entity $M_j^{d_j}$ that can be easily created, accessed and modified. Considering that only small portions of the mesh entities need higher p-orders in the adaptive analysis, a default value strategy is applied in managing the polynomial order field over the finite element mesh such that only the mesh entities that have different p-orders than the default value $\boldsymbol{p}_0$ need to be attached with the specified $\boldsymbol{p}$ vector. Different physical fields can be applied over the same finite element mesh such that a mesh entity may require different p-order vectors $\boldsymbol{p}$ that can be uniquely distinct with each other by the identification tags.

In Trellis, the FunctionSpace class is used to describe how a variable is interpolated over a given domain Ω^e that is able to get the degrees of freedom of the shape functions and evaluate the shape functions and the derivatives of the shape functions at the integration points [31] in the element domain. The directional variable p-order hierarchic approximation is implemented as derived shape function classes of FunctionSpace with respect to the topology of the domain Ω^e. For example, the class of hierarchic shape functions for a region domain is as follows:

```
class Hierarchic3D: public FunctionSpace {
 public:
    //Construction
```

```
        Hierarchic3D(pEntity ent, int defaultOrder, pMeshDataId attachId);
        //Evaluate the shape functions $N_a(\boldsymbol{\xi})$
        void _eval(double *pt, Mapping *m, double *result) const;
        //Evaluate the derivative of shape functions $N_{a,\boldsymbol{\xi}}(\boldsymbol{\xi})$
        void _eval1stDeriv(double *pt, Mapping *m, mVector *result) const;
        //Generate the degree of freedom of the domain $\Omega^e$
        void _KeyGenerator(const pEntity, Field *f, vecotr < DofKey > &) const;
        //Get the integration orders of the coordinates of the domain $\Omega^e$
        void getIntegrationOrder(int *intOrder) const;
        //Get the number of degree of freedom of the domain $\Omega^e$
        int getDim() const;
    private:
        int d-Order;
        int d-Dim;
        pEntity d-ent;
    }
```

The elemental hierarchic function space is initialized by the input default polynomial order (defualtOrder) and an identification tag (attachId) to retrieve the possible attached polynomial order vector. The hierarchic3D is a generic class that can compute the shape function or shape function derivatives for all of the topological region domain. The process can be described by the pseudo code in Table 6.1.

6.4 Numerical Integration Schemes for Directional Variable p-order Approximation

When Gauss-Legendre integration is applied, the number of integration points for the quadrilateral and hexahedral elements are defined as in Eq. 2.51 for directional variable p-order approximation. For other topological elements such as triangle, prism or pyramid, the definition of Gaussian integration rules at their natural coordinates for higher p-orders are not a straightforward process [37, 38, 46], especially for pyramid element. This thesis derives the integration schemes for these

Table 6.1: The Construction of Entity Hierarchic Shape Functions or Derivatives for a Region Domain

```
For each vertex of the mesh entity
    Compute the vertex shape function or derivative
For each edge of the mesh entity
    Retrieve the desired polynomial order of the edge
    For each polynomial order
        Compute the edge shape function or derivative at the specified order
For each face of the mesh entity
    Retrieve the desired polynomial order of the face
    Compute the $(\alpha_i, \alpha_j)$ set for the desired polynomial order
    For each set $(\alpha_i, \alpha_j)$
        Compute the face shape function or derivative
For the region mesh entity
    Retrieve the desired polynomial order of the region
    Compute the $(\alpha_i, \alpha_j, \alpha_k)$ set for the desired polynomial order
    For each set $(\alpha_i, \alpha_j, \alpha_k)$
        Compute the region shape function or derivative
```

elements by transforming their independent coordinates $\boldsymbol{\xi}$ into the coordinates $\boldsymbol{\xi}'$ for quadrilateral face or hexahedral region [46].

Let $\Upsilon(\boldsymbol{\xi})$ be the functional to be integrated over the element $\Omega(\boldsymbol{\xi})$ as [46],

$$\mathbf{I} = \int \boldsymbol{\Upsilon}(\boldsymbol{\xi})\mathbf{d}\boldsymbol{\xi}. \tag{6.22}$$

This function can be rewritten in the tensor product type coordinate $\boldsymbol{\xi'}$ as [46],

$$\mathbf{I} = \int \boldsymbol{\Upsilon}(\boldsymbol{\xi'})\mathbf{J}(\boldsymbol{\xi})\mathbf{d}\boldsymbol{\xi'}, \tag{6.23}$$

where $J(\boldsymbol{\xi}) = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\xi}'}\right|$. The p-orders resulting from the determinant transformation matrix $J(\boldsymbol{\xi})$ must be considered in the process to determine the Gauss-Legendre integration points for each coordinate ξ_i'. Let p_i be the resulting polynomial order at coordinate ξ_i' defined as,

$$p_i = p_i' + p_i'', \tag{6.24}$$

where p_i' is the polynomial order from $\Upsilon(\boldsymbol{\xi'})$ and p_i'' is the polynomial order from $J(\boldsymbol{\xi})$. The number of Gauss-Legendre integration points is $p_i + 1$ for coordinate ξ_i'

[27]. The influence of p_i'' from $J(\boldsymbol{\xi})$ on the number of integration points for different topological elements is investigated in detail below.

1. Triangle mesh face

   Let (p_1, p_2) be the highest power of the polynomial for the two independent coordinates (ξ_1, ξ_2) of the triangle face, $\xi_3 = 1 - \xi_1 - \xi_2$. Since the coordinate mapping between triangle face and quadrilateral face (ξ_1', ξ_2') is of the form [74],

$$\xi_1 = \frac{(1+\xi_1')(1-\xi_2')}{4},$$
$$\xi_2 = \frac{(1+\xi_2')}{2}, \tag{6.26}$$

   the corresponding mapping Jacobian is as,

$$J = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\xi}'}\right| = \begin{vmatrix} \frac{(1-\xi_2')}{4} & -\frac{(1+\xi_1')}{4} \\ 0 & \frac{1}{2} \end{vmatrix} = \frac{(1-\xi_2')}{8}. \tag{6.27}$$

   Since J is a 1st order polynomial is ξ_2' direction, the polynomial order becomes $p_2 + 1$ at ξ_2 direction and the total number of Gaussian integration points for a high order triangle face is,

$$n_g = (p_1 + 1) \times (p_2 + 2). \tag{6.28}$$

2. Tetrahedral mesh region

   Let (p_1, p_2, p_3) be the highest power of the polynomial for the three independent coordinates (ξ_1, ξ_2, ξ_3) of the tetrahedral mesh region. Since the coordinate mapping between tetrahedral region and hexahedral region (ξ_1', ξ_2', ξ_3') is of the form [46],

$$\xi_1 = \frac{(1+\xi_1')(1-\xi_2')(1-\xi_3')}{8},$$
$$\xi_2 = \frac{(1+\xi_2')(1-\xi_3')}{4}, \tag{6.30}$$
$$\xi_3 = \frac{(1+\xi_3')}{2},$$

the corresponding mapping Jacobian is as,

$$J = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\xi}'}\right| = \begin{vmatrix} \frac{(1-\xi'_2)(1-\xi'_3)}{8} & -\frac{(1+\xi'_1)(1-\xi'_3)}{8} & -\frac{(1+\xi'_1)(1-\xi'_2)}{8} \\ 0 & \frac{(1-\xi'_3)}{4} & -\frac{(1+\xi'_2)}{4} \\ 0 & 0 & \frac{1}{2} \end{vmatrix} = \frac{(1-\xi'_2)(1-\xi'_3)^2}{64}. \tag{6.32}$$

J is a polynomial with $1st$ order in ξ'_2 direction and $2nd$ order in ξ'_3 direction and the polynomial order for each coordinate become as (p_1, p_2+1, p_3+2). Thus, the total number of Gaussian integration points for a high order tetrahedral region is,

$$n_g = (p_1+1) \times (p_2+2) \times (p_3+3). \tag{6.33}$$

3. Prism mesh region

   Let (p_1, p_2, p_4) be the highest power of the polynomial for the three independent coordinates (ξ_1, ξ_2, ξ_4) of the prism mesh region. Where ξ_4 is the coordinate along the quadrilateral face of the prism region. The coordinate mapping between prism region and hexahedral region (ξ'_1, ξ'_2, ξ'_3) is of the form [74],

$$\begin{aligned} \xi_1 &= \frac{(1+\xi'_1)(1-\xi'_2)}{4}, \\ \xi_2 &= (1+\xi'_2)/2, \\ \xi_4 &= \xi'_3, \end{aligned} \tag{6.35}$$

the corresponding mapping Jacobian is as,

$$J = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\xi}'}\right| = \begin{vmatrix} \frac{(1-\xi'_2)}{4} & -\frac{(1+\xi'_1)}{4} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{vmatrix} = \frac{(1-\xi'_2)}{8}. \tag{6.37}$$

J is a $1st$ order polynomial at ξ'_2 direction and the polynomial order for each coordinate becomes as (p_1, p_2+1, p_3). Thus, the the total number of Gaussian integration points for a high order tetrahedral region is,

$$n_g = (p_1 + 1) \times (p_2 + 2) \times (p_4 + 1). \tag{6.38}$$

4. Pyramid mesh region

   Based on the coordinate mapping between prism and hexahedral mesh region shown in Eq. 6.20, the inverse mapping is [46],

$$\begin{aligned} \xi_1 &= \frac{\xi_1^{'}(1-\xi_3^{'})}{2}, \\ \xi_2 &= \frac{\xi_2^{'}(1-\xi_3^{'})}{2}, \\ \xi_3 &= \xi_3^{'}, \end{aligned} \tag{6.40}$$

   and the mapping Jacobian is,

$$J = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\xi}'}\right| = \begin{vmatrix} \frac{(1-\xi_3^{'})}{2} & 0 & \frac{\xi_1^{'}}{2} \\ 0 & \frac{(1-\xi_3^{'})}{2} & \frac{\xi_2^{'}}{2} \\ 0 & 0 & 1 \end{vmatrix} = \frac{(1-\xi_3^{'})^2}{4}. \tag{6.42}$$

   J is a polynomial with $2nd$ order at direction ξ_3' and the polynomial order for each coordinate becomes as $(p_1, p_2, p_3 + 2)$. Thus, the the total number of Gaussian integration points for a high order pyramid region is,

$$n_g = (p_1 + 1) \times (p_2 + 1) \times (p_3 + 3). \tag{6.43}$$

As an example, Table 6.2 presents the number of Gaussian-Legendre integration points for uniform and directional variable p-orders elements with linear geometry. It was shown that pyramid and tetrahedral elements demand more points than hexahedral and prismatic elements because of the polynomial order introduced by the coordinates transformation mapping. The number can be sizable reduced if directional variable p-orders are applied.

Table 6.2: **Number of Gaussian Integration Points for Uniform and Directional p-Order**

	(p_1, p_2)	n_g	(p_1, p_2)	n_g
Quadrilateral	(6, 6)	49	(3, 6)	28
Triangle	(6, 6)	56	(3, 6)	32
	(p_1, p_2, p_3)	n_g	(p_1, p_2, p_3)	n_g
Hexahedral	(6, 6, 6)	342	(2, 4, 6)	105
Prism	(6, 6, 6)	392	(2, 4, 6)	126
Pyramid	(6, 6, 6)	441	(2, 4, 6)	180
Tetrahedral	(6, 6, 6)	504	(2, 4, 6)	192

CHAPTER 7

Automated Adaptive Directional Variable p-Version Analysis in 3D Curved Domains

This chapter describes the algorithm for the adaptive directional variable p-version analysis procedure implemented in Trellis. The objective of the procedure is to achieve the prescribed tolerance with less degree of freedom and computational cost. The three key components in the procedure, error estimator, directional correction indicator, and p-order enrichment strategy, are discussed.

7.1 Automated Adaptive Directional Variable p-version Analysis in Trellis

Given the prescribed tolerance τ defined as the error measure in the energy norm, the adaptive directional variable p-version analysis procedure for 3D curved domains in Trellis consists of the following steps:

Step 1: Apply appropriate physical attributes on the geometry model. For example, the boundary conditions, the loading, the material properties etc..

Step 2: Generate a near optimal mesh configuration using the automatic p-version mesh generation procedure presented in part II of this thesis. Assign uniform initial lower polynomial order $p = 2$ over the mesh.

Step 3: Solve the Galerkin finite element formulation of the given problem using p-version method as described in Section 2.2 to obtain the finite element solution u^h. Compute the solution accuracy η discussed in section 7.2 as follows:

- Evaluate error estimator η_k on each element Ω_k.
- Compute the error estimator over the whole mesh as [6],

$$\eta^2 = \sum_k \eta_k^2. \tag{7.1}$$

Step 4: If $\eta \leq \tau$, then the procedure stops.

Step 5: Else, apply directional p-order enrichment strategy that consists of the following aspects:

- Determine all the elements with $\eta_k \geq \vartheta$ that need polynomial order enrichment. ϑ is a threshold value determined based on the principle of equilibrating the error in the elements [11] and will be detailed discussed in section 7.4.
- For each selected element Ω_k, compute the directional correction indicator $\eta_k^{\xi_i}$ as discussed in 7.3 at each independent coordinate ξ_i.
- Predict the necessary polynomial enrichment $\Delta \boldsymbol{p}_{\xi_i}$ as discussed in 7.4 at each independent coordinate ξ_i based on the directional correction indicator $\eta_k^{\xi_i}$.

Step 6: Update the finite element discretization with the new p-order distribution over the finite element mesh and go to Step 3.

In the adaptive directional variable p-version analysis procedure, the error estimator in step 3 provides an accurate measure of the error in energy norm. The directional correction indicator and enrichment strategy in step 5 determine when and how to enrich the polynomial order for the elements at each independent coordinate.

7.2 Error Estimators for the Adaptive p-Version Method

The error estimator provides an accurate measure of the finite element solution accuracy that allows the adaptive procedure to determine when the prescribed level of accuracy is achieved. The existing a posteriori error estimator can be grouped into three categories [6]: estimators based on (a) gradient recovery, (b) extrapolation, and (c) residuals.

7.2.1 Gradient Recovery Error Estimator

The recovery based method smoothes the gradients of the solutions on a given mesh and the smoothed solution is compared with the original solution to assess the error [136, 137]. This type of a posteriori error estimator is very effective for some classes of problem and h-version adaptivity [68, 70, 72]. Its use in p-version analysis in less obvious. Therefore, this approach is not considered in the current work.

7.2.2 Extrapolation Error Estimator

Extrapolation based methods have been used for uniform p-version adaptivity with the error estimator obtained using a sequences of hierarchical p-version finite element solutions [124]. The derivation of this estimator starts from the *a-priori* p-version error estimate in Eq. 2.53,

$$||e|| \leq \frac{C}{N^{\beta}}, \tag{7.2}$$

where C and β are positive constants and N is the number of degrees of freedom.

Let $\Pi(u)$ define the potential energy as [124],

$$\Pi(u) = \frac{1}{2}a(u,\ u) - (u, f) - (u, h)_{\Gamma_h}. \tag{7.3}$$

Therefore, the error measured in energy norm can be written as [124],

$$||e||^2 = \frac{1}{2}a(e,\ e) = \Pi(u^h) - \Pi(u). \tag{7.4}$$

Thus, Eq. 7.2 can be written in the following form,

$$||e||^2 = \Pi(u^h) - \Pi(u) \approx \frac{C^2}{N^{2\beta}}. \tag{7.5}$$

The rate of convergence is characterized by the constant β. Eq. 7.5 is such that for sufficiently large N the “less than or equal” sign can be replaced by “approximately equal” [124].

Let Π denote the exact potential energy. Knowing $\Pi(u^h)$ and N for three p-levels of extension as Π_p, Π_{p+1} and Π_{p+2} and their associated numbers of degrees of freedom N_p, N_{p+1} and N_{p+2}, respectively, the Eq. 7.5 for the three p-levels are,

$$\Pi_p - \Pi \approx \frac{C^2}{N_p^{2\beta}}, \tag{7.6}$$

$$\Pi_{p+1} - \Pi \approx \frac{C^2}{N_{p+1}^{2\beta}}, \tag{7.7}$$

$$\Pi_{p+2} - \Pi \approx \frac{C^2}{N_{p+2}^{2\beta}}. \tag{7.8}$$

The constant C can be eliminated from the equations by dividing Eq. 7.6 by Eq. 7.7 and Eq. 7.7 by Eq. 7.8 as,

$$\frac{\Pi_p - \Pi}{\Pi_{p+1} - \Pi} \approx \left(\frac{N_{p+1}}{N_p}\right)^{2\beta}, \tag{7.9}$$

$$\frac{\Pi_{p+1} - \Pi}{\Pi_{p+2} - \Pi} \approx \left(\frac{N_{p+2}}{N_{p+1}}\right)^{2\beta}. \tag{7.10}$$

The constant β can be eliminated from the equations by taking logarithm on both sides of the Eq. 7.9 and 7.10 then dividing the Eq. 7.9 by 7.10 to be [124],

$$\frac{\log \frac{\Pi - \Pi_p}{\Pi - \Pi_{p+1}}}{\log \frac{\Pi - \Pi_{p+1}}{\Pi - \Pi_{p+2}}} \approx \frac{\log \frac{N_{p+1}}{N_p}}{\log \frac{N_{p+2}}{N_{p+1}}}. \tag{7.11}$$

Since N_p, N_{p+1} and N_{p+2} are the known degrees of freedom for the three p-levels, the right side of the Eq. 7.11 is a constant and denote by α as,

$$\alpha = \frac{\log \frac{N_{p+1}}{N_p}}{\log \frac{N_{p+2}}{N_{p+1}}}. \tag{7.12}$$

Therefore, the Eq. 7.11 can be written as,

$$\frac{\Pi - \Pi_p}{\Pi - \Pi_{p+1}} \approx \left(\frac{\Pi - \Pi_{p+1}}{\Pi - \Pi_{p+2}}\right)^{\alpha}, \tag{7.13}$$

where the exact potential energy Π is the unknown to be solved. Since α is general not an integer, the analytic expression for Π can not be directly derived from Eq. 7.13. Numerical methods such as Newton's method [29] can be applied to solve the Eq. 7.13 to get an estimation of the exact potential strain energy Π.

Once the exact potential energy Π is known, the constants β and C can be obtained by substituting Π back to Eq. 7.6-7.10. Given Π, Π_p, Π_{p+1} and Π_{p+2} are known, the error measured in the energy norm for each p-level can be computed using Eq. 7.4.

7.2.3 Residual Error Estimator

The residual type of error estimators have been successfully applied in hp-version method [4, 5, 44, 51, 91, 92, 94] can be divided into two categories [6]: explicit residual and implicit residual. The explicit residual error estimator measures the residual of the finite element solution in the model problem to reflect how well the finite element approximation satisfies the underlying boundary value problem in the interior of the domain, the Neumann boundary condition and the regularity of the approximation on the inter-element boundaries [6]. The implicit residual error estimator solves an auxiliary boundary value problem typical on each element to attain an approximation to the actual error function. An elemental implicit residual error estimator is implemented in Trellis to serve as the *a-posterior* error estimator for the adaptive directional variable p-version analysis procedure.

For the second order partial differential equation with abstract weak form as given in Eq. 2.3, the error $e = u - u^h$ satisfies a local boundary value problem on each element Ω_k as [6],

$$a_k(w,\ e)\ =\ (w,\ f)_k - a_k(w,\ u^h) + \int_{\partial\Omega_k} \frac{\partial u}{\partial n_k} w\ ds, \quad \forall w \in W_k, \tag{7.14}$$

where $a_k(\ ,\ )$ and $(\ ,\ )_k$ are the bilinear and linear form on the element Ω_k. W_k is the variation space defined on the element Ω_k,

$$W_k = \{w \in H^1(\Omega_k), \quad w = 0 \quad on \quad \Gamma_g \cap \partial\Omega_k\}. \tag{7.15}$$

The boundary conditions imposed on the boundaries of the element Ω_k are as follows:

- The element boundaries that intersect a portion of the Γ_g such that,

$$e = 0, \qquad on \qquad \partial\Omega_k \cap \Gamma_g. \tag{7.16}$$

Eq. 7.16 indicates that the finite element solution u^h represents the Dirichlet boundary condition exactly such that the trial space for the local boundary value problem in Eq. 7.14 can be the same as W_k defined in Eq. 7.15.

- The element boundaries that intersect a portion of the Γ_h such that,

$$\alpha\frac{\partial e}{\partial n_k} = g - \alpha\frac{\partial u^h}{\partial n_k}. \tag{7.17}$$

- The element boundaries lie on the interior of the geometric domain such that

$$\frac{\partial e}{\partial n_k} = \frac{\partial u}{\partial n_k} - \frac{\partial u^h}{\partial n_k}. \tag{7.18}$$

In Eq. 7.14, $\frac{\partial u}{\partial n_k}$ is the flux on the boundary of the element Ω_k. One approach to construct the flux is to replace the true flux $\frac{\partial u}{\partial n_k}$ by an approximation obtained from the finite element approximation itself [6],

$$\frac{\partial u}{\partial n_k} \approx \langle\frac{\partial u^h}{\partial n_k}\rangle, \tag{7.19}$$

where $\langle\frac{\partial u^h}{\partial n_k}\rangle$ is the averaging of the discontinuous finite element approximation over the boundary between two neighboring elements Ω_k and Ω_j as follows [6],

$$\langle\frac{\partial u^h}{\partial n_k}\rangle = \frac{1}{2}\cdot\{(\nabla u^h)_k + (\nabla u^h)_j\}. \tag{7.20}$$

Then, the flux definition can be extended to include the exterior boundary,

$$\langle\frac{\partial u^h}{\partial n_k}\rangle = \begin{cases} \frac{1}{2}\cdot\{(\nabla u^h)_k + (\nabla u^h)_j\}, & \text{on } \partial\Omega_k \cap \partial\Omega_j \\ h, & \text{on } \partial\Omega_k \cap \Gamma_h \\ \frac{\partial u^h}{\partial n_k}, & \text{on } \partial\Omega_k \cap \Gamma_g \end{cases} \tag{7.21}$$

However, simple averaging of the boundary fluxes on neighboring elements fails to satisfy the overall equilibrium requirements for the local boundary problem in Eq. 7.14 [6]. One approach to solve this problem is to carefully construct a subspace of W_k on which the bilinear form of Eq. 7.14 will be coercive [6].

An alternative to compute the boundary flux data is the equilibrated residual method that is implemented in Trellis for the use of this work. Central to this approach is to carefully choose $\frac{\partial u}{\partial n_k}$ such that the underlying boundary value problem in Eq. 7.14 is naturally well-posed by satisfying overall equilibrium [6]. Let g_k be the approximation of the flux on the boundary of element Ω_k as,

$$g_k \approx \frac{\partial u}{\partial n_k}, \qquad \textit{on } \partial\Omega_k. \tag{7.22}$$

Because the flux is continuous on the interelement boundaries,

$$g_k + g_j = 0 \qquad \textit{on } \partial\Omega_k \cap \partial\Omega_j, \tag{7.23}$$

g_k is chosen to be the same as the flux on the exterior boundary Γ_h,

$$g_k = h \qquad \textit{on } \partial\Omega_k \cap \Gamma_h. \tag{7.24}$$

The boundary data and the interior residual must satisfy the overall equilibrium condition for the local boundary value problem in Eq. 7.14. Thus, the boundary flux g_k must be in equilibrium with the interior load as follows [6],

$$(w,\ f)_k - a_k(w,\ u^h) + \int_{\partial\Omega_k} g_k w \ ds = 0 \qquad \forall w \in W_k. \tag{7.25}$$

The solution g_k of Eq. 7.25 serves as the flux on the boundaries of the element Ω_k such that the local error boundary value problem in Eq. 7.14 can be written as,

$$a_k(w,\ e) \ = \ (w,\ f)_k - a_k(w,\ u^h) + \int_{\partial\Omega_k} g_k w \ ds, \quad \forall w \in W_k, \tag{7.26}$$

Let $W_k^h \subset W_k$ be the finite element subspace defined over the element Ω_k that only includes the subset of shape functions for higher p-order. The finite element solution for the local boundary value problem in Eq. 7.26 is:

Find $\phi_k \in W_k^h$, such that [6],

$$a_k(w^h,\ \phi_k) \ = \ (w^h,\ f)_k - a_k(w^h,\ u^h) + \int_{\partial\Omega_k} g_k w^h \ ds, \quad \forall w^h \in W_k^h. \tag{7.27}$$

Therefore, the error estimator η_k for the element Ω_k is computed as [6],

$$\eta_k = ||\phi_k||. \tag{7.28}$$

The solution accuracy over the mesh is obtained by summing the local error estimators as [6],

$$\eta = \left(\sum_k \eta_k^2 \right)^{\frac{1}{2}}. \tag{7.29}$$

The $||\phi_k||$ in Eq. 7.28 is the error functional measured in the energy norm as given in Eq. 2.52,

$$||\phi_k|| = a_k(\phi_k, \phi_k)^{\frac{1}{2}}. \tag{7.30}$$

The error estimator η_k is used in the adaptive p-version analysis procedure to determine when to perform p-order enrichment for a element Ω_k by checking whether $\eta_k \geq \vartheta$. However, additional information is needed in the adaptive directional p-version analysis to determine how to elevate the p-orders in the appropriate independent coordinates for the selected element Ω_k.

7.3 Stress Resultant Based Correction Indicator

Most of the conventional adaptive p- and hp-version analysis procedures use the error estimators η_k as defined in Eq. 7.28 as the correction indicator to elevate the p-order one or two order higher for the selected element without concerning any directional behavior [4, 5, 44, 51, 91, 92, 94].

Another type of correction indicator used for adaptive p-version analysis procedures is constructed based on the jumps in the stress resultants across shared edges (2D) or shared faces (3D) [33, 56, 120, 124]. Since this approach does not compute the local error estimator η_k for each element, it works in conjunct with the extrapolation error estimator as discussed in Section 7.2.2 to get a global estimation of the solution accuracy. This form of correction indicator is used in the StressCheck [120] based results given in chapter 8, it is briefly reviewed here.

Let Λ be the sum of the absolute values of the difference in stress resultants. Figure 7.1 shows the jumps for the shared face of the elements Ω_k and Ω_{k+1} given by [120],

$$\Lambda_j^k = \left|F_x^k - F_x^{k+1}\right| + \left|F_y^k - F_y^{k+1}\right| + \left|F_z^k - F_z^{k+1}\right|, \tag{7.31}$$

where $F_x,\ F_y,\ F_z$ are the stress resultants.

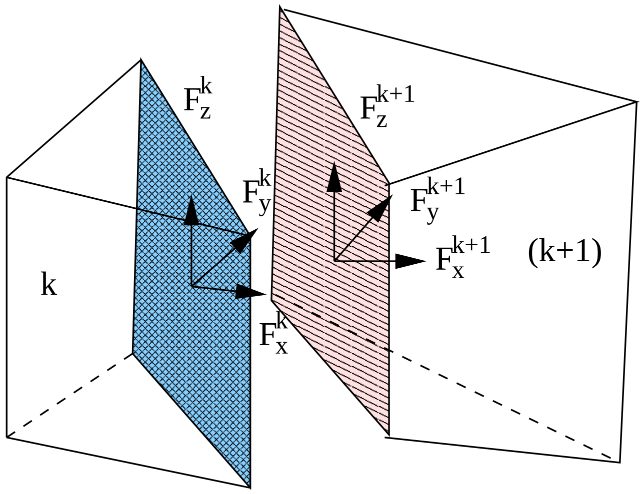


Figure 7.1: Correction Indicator Based on Stress Resultants

Let Λ_{max}^k is the largest jumps of the element and Λ^T is the sum of all the largest jumps over all the elements as 120],

$$\Lambda_{max}^k = max(\Lambda_j^k), \qquad \Lambda^T = \sum_{k=1}^{n_{el}} \Lambda_{max}^k, \tag{7.32}$$

where n_{el} is the number of element over the mesh. The relative jump Φ^k for the element is computed as,

$$\Phi^k = \frac{\Lambda_{max}^k}{\Lambda^T}. \tag{7.33}$$

A correction indicator for the element Ω_k has been defined as [120],

$$\theta_k = \frac{\Phi^k}{\Phi_{max}}, \tag{7.34}$$

where Φ_{max} is the largest Φ^k over the mesh. $0 \leq \theta_k \leq 1$. The p-order of an element Ω_k is raised to be a order higher if the correct indicator θ_k is larger than a threshold value ϑ defined between 0.75-0.85 [33, 56, 120, 124]. It is shown from the definition of the correction indicator in Eq. 7.34 that no directional behavior needed by the independent coordinates is considered.

7.4 Residual Based Directional Correction Indicator

A directional correction indicator that can be obtained by solving the local error boundary value problem in Eq. 7.27 considers the addition of terms in the specific element coordinate directions. The error e over the element Ω_k is defined as $e = u - u^h_{\boldsymbol{p}}$. Since the element space with higher polynomial order is larger than the space with lower order, an approximation to this error can be defined as [90],

$$e \approx E^h_{\boldsymbol{p}+\boldsymbol{r}} = u^h_{\boldsymbol{p}+\boldsymbol{r}} - u^h_{\boldsymbol{p}}, \qquad r \geq 1. \tag{7.35}$$

$E^h_{\boldsymbol{p}+\boldsymbol{r}}$ is obtained by solving the local boundary value problem in Eq. 7.27 with the finite element solution $\phi^{\boldsymbol{p}+\boldsymbol{r}}_k$ carefully constructed at the finite element subspace W^h_k with polynomial order $\boldsymbol{p}+\boldsymbol{r}$ [6, 90]. Let N_l be the set of shape functions in the finite element subspace W^h_k with polynomial order $\boldsymbol{p}+\boldsymbol{r}$ and d_l is the coefficients, $n_{\boldsymbol{p}+\boldsymbol{r}}$ is the number of shape functions, $\phi^{\boldsymbol{p}+\boldsymbol{r}}_k$ can be written as,

$$\phi^{\boldsymbol{p}+\boldsymbol{r}}_k = \sum_{l=1}^{n_{\boldsymbol{p}+\boldsymbol{r}}} N_l d_l. \tag{7.36}$$

Note that the summation only contains the extra set of shape functions needed by the finite element space goes from polynomial order $\boldsymbol{p}$ to $\boldsymbol{p}+\boldsymbol{r}$.

This procedure can also be applied for each coordinate ξ_i of the element Ω_k independently such that the error for each coordinate can be written as,

$$e \approx E^h_{\boldsymbol{p}+\boldsymbol{r}|_{\xi_i}} = u^h_{\boldsymbol{p}+\boldsymbol{r}|_{\xi_i}} - u^h_{\boldsymbol{p}}, \qquad r \geq 1. \tag{7.37}$$

$E^h_{\boldsymbol{p}+\boldsymbol{r}|_{\xi_i}}$ can be obtained by solving the local boundary value problem in Eq. 7.27 by carefully defining $\Phi^{\boldsymbol{p}+\boldsymbol{r}|_{\xi_i}}_k$ at the finite element subspace W^h_k with polynomial order $\boldsymbol{p}+\boldsymbol{r}\ |_{\xi_i}$.

Let N'_j be the shape functions in the finite element subspace W^h_k with polynomial order $\boldsymbol{p}+\boldsymbol{r}\ |_{\xi_i}$ and d'_j is the coefficients, $n_{\boldsymbol{p}+\boldsymbol{1}|_{\xi_i}}$ is the number of shape functions, $\Phi_k^{\boldsymbol{p+r}|\xi_i}$ can be written as,

$$\phi_k^{\boldsymbol{p+r}|\xi_i} = \sum_{j=1}^{n_{\boldsymbol{p+r}|\xi_i}} N'_j d'_j, \tag{7.38}$$

A directional correction indicator $\eta_k^{\xi_i}$ for the independent coordinate ξ_i of the element Ω_k is defined as,

$$\eta_k^{\xi_i} = ||\phi_k^{\boldsymbol{p+r}|\xi_i}||, \tag{7.39}$$

that measures difference between the finite element solution in order $\boldsymbol{p}$ and one order higher in the direction ξ_i [90].

The Eq. 7.39 indicates that the computation of the directional correction indicator $\eta_k^{\xi_i}$ requires solving the local error boundary value problem in the Eq. 7.27 for every independent coordinate ξ_i. An alternative is to solve the local error boundary value problem once in the finite element subspace that increases the polynomial order at every direction to obtain $\phi_k^{\boldsymbol{p+r}}$. Since hierarchic shape functions are applied in this thesis, the shape functions N'_j in Eq. 7.38 are a subset of the shape functions N_l in Eq. 7.36 as,

$$N'_j \subset N_l. \tag{7.40}$$

In this case, the difference between d'_j and d_l is not large [90]. Therefore the correction indicator in Eq. 7.39 can be computed as,

$$\eta_k^{\xi_i} \approx ||\phi_k^{\boldsymbol{p+r}|\xi_i}|| = ||\sum_{j=1}^{n_{\boldsymbol{p+r}|\xi_i}} N'_j d_j||. \tag{7.41}$$

for each independent coordinate ξ_i by using the same coefficients d_l.

7.5 Directional Variable p-Version Refinement Strategy

An effective refinement strategy for the directional variable p-version analysis procedure should be able to address the following two issues:

- A threshold value to determine which elements should be refined.

- A criteria to predict appropriate p-order enrichment for the element independent coordinates based on the directional correction indicator.

Most the p- and hp-version adaptive procedures determine the threshold value ϑ as a fraction of the maximum element error estimator over the mesh,

$$\vartheta = \theta\ max(\eta_k), \tag{7.42}$$

where θ is a fixed value typically between 0.75-1.0 [5, 44, 56, 94]. One or two order enrichment is performed on the element Ω_k whose error estimator is $\eta_k \geq \vartheta$. However, numerical analysis results in [5, 44, 56, 94] demonstrated that with this strategy more iteration steps are needed for the adaptive analysis procedures to achieve the optimal convergence rate, especially in the asymptotic rate, due to the following two reasons:

- The fixed constant θ may not include all the elements that need finer resolution at each step.
- One or two order enrichment many not be sufficient for the select element.

Aiming at achieving the prescribed accuracy τ with lowest degree of freedom and less computational cost, this thesis determines the threshold value ϑ based the basic theory that a finite element discretization is asymptotic optimal if the error is equidistributed over each element [11]. Thus, the prescribed accuracy τ can be translated to the threshold value ϑ as follows,

$$\tau = ||e|| = (n_{el}\vartheta^2)^{\frac{1}{2}}, \tag{7.43}$$

such that,

$$\vartheta = \frac{\tau}{\sqrt{n_{el}}}, \tag{7.44}$$

where n_{el} is the number of elements of the mesh.

Based on the definition of the directional correction indicator in Eq. 7.39, an element Ω_k needs p-order enrichment for its independent coordinates ξ_i if,

$$\eta_k^{\xi_i} \geq \vartheta, \qquad i = 1, ..., n_{sd}, \tag{7.45}$$

where n_{sd} is the problem dimension.

The p-order enrichment $\Delta\boldsymbol{p}_{\xi_i}$ for the coordinate ξ_i is assuming the error of a p-version analysis at a level $\boldsymbol{p} + \Delta\boldsymbol{p}_{\xi_i}$ can be predicted as [61],

$$||e||_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}} = (\frac{n_{\boldsymbol{p}}}{n_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}}})^{\beta}||e||_{\boldsymbol{p}}. \tag{7.46}$$

where $n_{\boldsymbol{p}}$ and $n_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}}$ are the number of degrees of freedom at polynomial order $\boldsymbol{p}$ and $\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}$ respectively. β is the estimated rate of convergence at polynomial order $\boldsymbol{p}$ that can be approximately computed as [61],

$$\beta = \frac{\nu}{2}max(\boldsymbol{p} + \boldsymbol{1}), \tag{7.47}$$

ν is a reduction factor selected as $\frac{1}{1.25} = 0.8$ to allow 25% error in the estimated value of β [61].

In this work, the p-order enrichment $\Delta\boldsymbol{p}_{\xi_i}$ for the coordinate ξ_i is determined assuming that the error over the element Ω_k can be reduced to the threshold value ϑ by increasing the p-order on coordinate ξ_i independently. Thus, Eq. 7.46 can be written as,

$$\vartheta = (\frac{n_{\boldsymbol{p}}}{n_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}}})^{\beta}\eta_k^{\xi_i}. \tag{7.48}$$

Therefore, the number of degrees of freedom needed for the element Ω_k at the polynomial order $\boldsymbol{p} + \Delta\boldsymbol{p}_{\xi_i}$ is computed as,

$$n_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}} = n_{\boldsymbol{p}} \times (\frac{\vartheta}{\eta_k^{\xi_i}})^{\frac{1}{\beta}}. \tag{7.49}$$

$n_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}}$ in Eq. 7.49 is solved using numerical methods such as Newton's method [29]. In the adaptive p-version analysis, the polynomial enrichment $\Delta\boldsymbol{p}_{\xi_i}$ at the coordinate direction ξ_i of the element Ω_k is determined as the minimal order that the degrees of the freedom of the element Ω_k is greater than $n_{\boldsymbol{p}+\Delta\boldsymbol{p}_{\xi_i}}$ in Eq. 7.49.

The directional p-order refinement strategy in Eq. 7.49 may lead to the overrefinement of the p-orders on the coordinate ξ_i because it does not consider the interference effect between each coordinate. Such overrefinement is acceptable comparing to the uniform refinement strategy at each coordinate over the element.

Since $\Delta \boldsymbol{p}_{\xi_i}$ predicts the enriching p-order at the ξ_i direction of the element Ω_k^e, the closure mesh entities of the element must also update their polynomial orders along ξ_i direction. As a simple example, Figure 7.2 shows that the p-orders of the mesh edges M_0^1, M_2^1 and ξ_1 of the element M_0^2 must be raised two order higher in case that $\Delta \boldsymbol{p}_{\xi_1} = 2$ and the p-orders of the mesh edges M_1^1, M_3^1 and ξ_1 of the element M_0^2 must be raised one order higher in case that $\Delta \boldsymbol{p}_{\xi_2} = 1$. For a mesh entity shared by two different polynomial order elements, it is always the higher order element wins. For example, the mesh edge M_1^1 in Figure 7.3 is 6th order due to the connected mesh face M_1^2 is of polynomial order 6 at this direction.

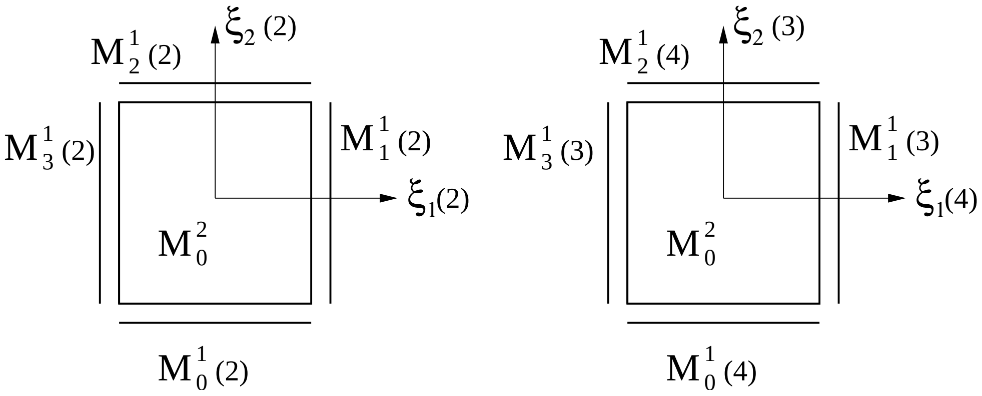


Figure 7.2: p-Order Enrichment At Direction ξ_1 and ξ_2

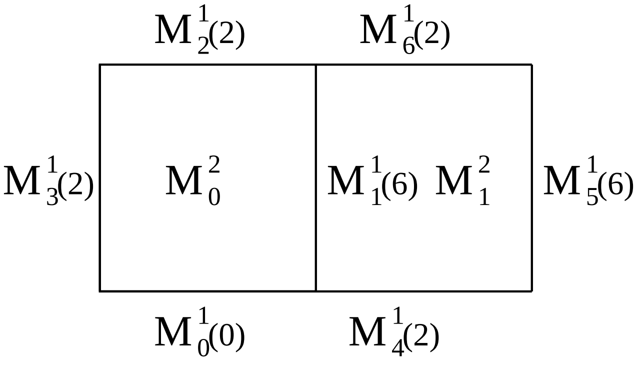


Figure 7.3: p-Order Enrichment For a Mesh Entity Connecting to Two Different Polynomial Order Elements at Direction ξ_1

CHAPTER 8

Analysis Results for p-Version Method

This chapter presents the p-version analysis results that focus on the following three aspects:

- The influence of geometric approximation on the solution accuracy of p-version method.
- The performance of the p-version meshes for models with singularities or thin sections in p-version analysis software StressCheck.
- The directional adaptive p-version analysis in Trellis.

8.1 The Influence of Geometric Approximation on the Solution Accuracy of p-Version Method

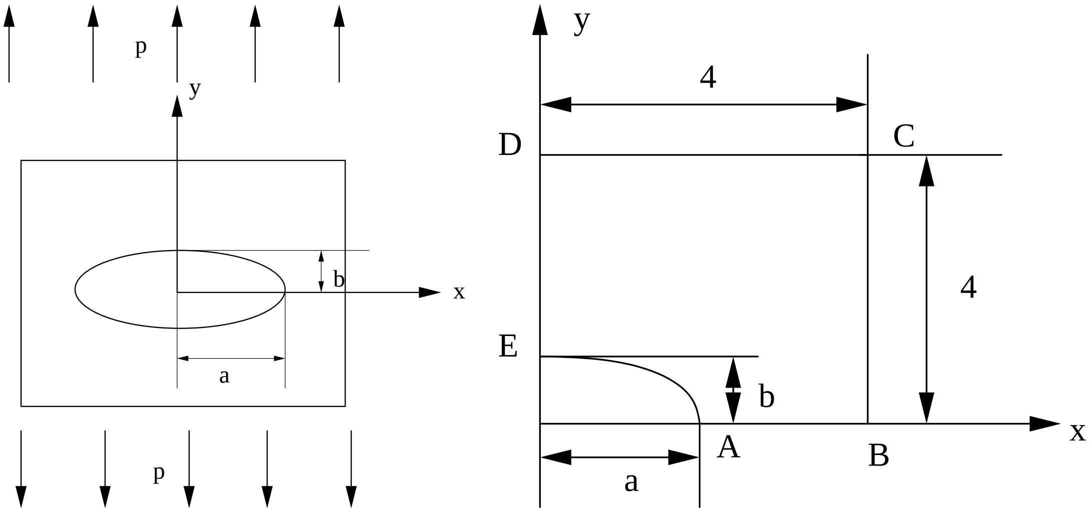
This section discusses the influence of geometric approximation on the solution accuracy of p-version finite element method by using a benchmark two-dimensional elasticity problem for which analytic expressions for the exact displacement and stress field are known.

8.1.1 Model Problem Description

An infinite plane weakened by an elliptical hole is deformed by the application of uniform tensile stress in the vertical direction at infinity as shown in Figure 8.1(a). The relevant geometric parameters are the major axis a and minor axis b of the inner ellipse. These parameters are typically related to a third parameter, m, as,

$$a = 1 + m, \quad b = 1 - m, \quad 0 \leq m < 1, \tag{8.1}$$

where $m = 0$ indicates the ellipse turn to be circle and $m = 1$ is a sharp crack. Due to the double symmetry of the problem, only one quarter of the sub-domain



(a) (b)

Figure 8.1: Elliptical Hole in an Infinite Plane Under the Uniform Tensile Stress at Infinity

ABCDE needs to be investigated as shown in Figure 8.1(b). The exact stresses for the infinite domain problem are known along edges BC and DC and are given by,

$$\sigma_x = \frac{1}{2}\{S(1-\cos 2\theta) + 2\tau_{\rho\rho}\cos 2\theta - 2\tau_{\rho\theta}\sin 2\theta\}, \tag{8.2}$$

$$\sigma_y = \frac{1}{2}\{S(1+\cos 2\theta) + 2\tau_{\rho\rho}\cos 2\theta + 2\tau_{\rho\theta}\sin 2\theta\}, \tag{8.3}$$

$$\tau_{xy} = \frac{1}{2}\{(2\tau_{\rho\rho} - S)\sin 2\theta + 2\tau_{\rho}\theta\cos 2\theta\} \tag{8.4}$$

where

$$S = \rho^4 - m^2 - 2m + 2\rho^2\cos 2\theta, \tag{8.5}$$

$\tau_{\rho\rho}$, $\tau_{\rho\theta}$ are the stress components expressed in elliptical coordinate system (ρ, θ) [88].

The mapping between Cartesian coordinate system (x, y) and elliptical coordinate system (ρ, θ) is,

$$x = R(\rho + \frac{m}{\rho})\cos\theta, \quad x = R(\rho - \frac{m}{\rho})\sin\theta, \quad R = \frac{a+b}{2}. \tag{8.6}$$

Traction (Neumann) boundary conditions are applied on edges BC and CD and symmetric Dirichlet boundary conditions are imposed on edges DE and AB.

8.1.2 Maximum Stress and Strain Energy

The maximum stress σ_{max} of this problem is concentrated at vertex A and is a function of ratio a/b being defined as [88],

$$\sigma_{max} = (1 + \frac{2a}{b})p. \tag{8.7}$$

The finite element stress is computed directly in this study. The strain vector $\boldsymbol{\varepsilon}$ is computed directly from the finite element displacement solution u^h as [124],

$$\boldsymbol{\varepsilon} = \left\{ \frac{\partial u_x^h}{\partial x}, \quad \frac{\partial u_y^h}{\partial y}, \quad \frac{\partial u_x^h}{\partial y} + \frac{\partial u_y^h}{\partial x} \right\}. \tag{8.8}$$

The finite element stress vector $\boldsymbol{\sigma}$ is then computed by applying the stress-strain relationship as [88],

$$\boldsymbol{\sigma} = \boldsymbol{E} \cdot \boldsymbol{\varepsilon}, \tag{8.9}$$

where $\boldsymbol{E}$ is the symmetric positive definite material stiffness matrix [88]. For the isotropic material under plain strain assumption used in this study, $\boldsymbol{E}$ is of the form [88],

$$\boldsymbol{E} = \begin{bmatrix} \frac{E(1-v)}{(1+v)(1-2v)} & \frac{Ev}{(1+v)(1-2v)} & 0 \\ 0 & \frac{E(1-v)}{(1+v)(1-2v)} & 0 \\ 0 & 0 & \frac{E}{2(1+v)} \end{bmatrix}, \tag{8.10}$$

where E is the Young's Module and v is Poisson's ratio. The search for the maximum computed finite element stress σ_{max}^{fem} is conducted over not only Gauss Quadrature points but also the vertices of each element [124]. The relative error in maximum stress is defined as,

$$\varepsilon_\infty = \frac{|\sigma_{max}^{fem} - \sigma_{max}|}{\sigma_{max}}\%, \tag{8.11}$$

is of the most engineering interest. The exact potential energy $\Pi(u)$ of the subdomain loaded by traction only without body force can be computed as [130],

$$\Pi(u) = -\oint (T_n u_n + T_t u_t) ds = -[\int_B^C (T_n u_n + T_t u_t) ds + \int_C^D (T_n u_n + T_t u_t) ds], \quad (8.12)$$

where T_n, T_t and u_n, u_t are the normal and tangential traction components and displacement components. The angle α is measured from the positive x axis to the normal boundaries, thus for BC and CD α is 0^0 and 90^0 respectively, so Eq. 8.12 can be simplified as [130],

$$\Pi(u) = -[\int_B^C (u_x \sigma_x + u_y \tau_{xy}) dy + \int_C^D (u_y \sigma_y + u_x \tau_{xy}) dx], \quad (8.13)$$

where σ_x, σ_y, τ_{xy} are stress tensor components and u_x and u_y are displacement components [124]. The exact potential energy is obtained by submitting the exact stress and displacement expressions defined in Eq. 8.2-8.4 into Eq. 8.13. The finite element potential energy $\Pi(u^h)$ is computed based on the definition of the potential energy in Eq. 7.3 as,

$$\Pi(u^h) = \frac{1}{2} a(u^h,\ u^h) - (u^h, f) - (u^h, h)_{\Gamma_h} = -\frac{1}{2}\left((u^h, f) + (u^h, h)_{\Gamma_h}\right), \quad (8.14)$$

that is the evaluation of the product of load vector and finite element solution over the boundary [130]. The relative error in energy norm is defined as,

$$e_r = \sqrt{\frac{\Pi(u) - \Pi(u^h)}{\Pi(u)}} \times 100\%, \quad (8.15)$$

8.1.3 Finite Element Meshes and Geometric Approximation Shapes

Two parameters $m = 0.25$ and $m = 0.9$ are selected to construct two test models. The ellipse with parameter $m = 0.25$ is smooth with a weak stress concentration 4.333333 while the ellipse with parameter $m = 0.9$ is sharp with a strong stress concentration 39. An isotropic material with Young's Module of 1.0 and Poisson's ratio of 0.3 is used under the assumption of plane strain. The stress applied at the infinite boundaries is $p = 1$. Table 8.1 provides the exact potential energy (up to 7 digits significant figures) and the exact maximum stress for selected geometric shapes.

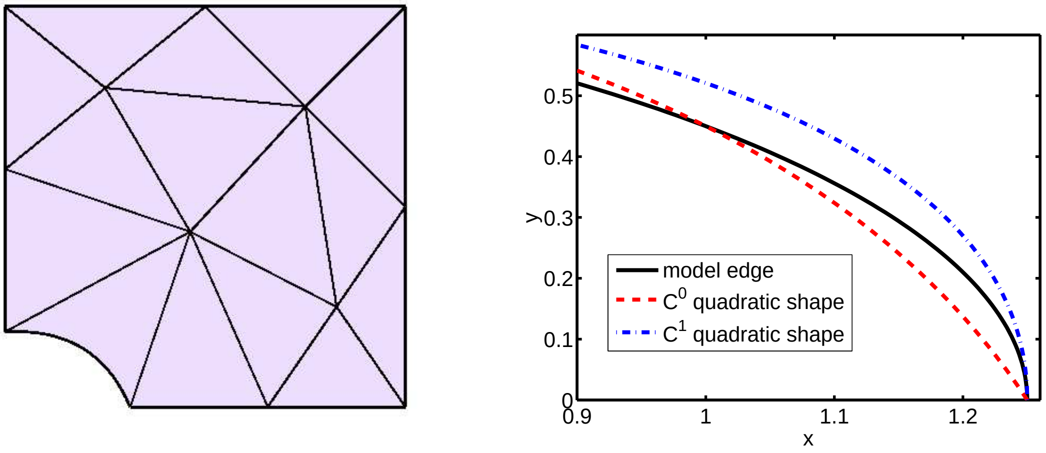
Table 8.1: Parameters for Test Problem 1

m	a	b	a/b	σ_{max}	$\Pi(u)$
0.25	1.25	0.75	1.677	13/3	-7.8462131
0.9	1.9	0.1	19	39	-8.2308397

For both of the models, a coarse mesh with only one edge classified on the ellipse AE (see Figure 8.2(a)) is used to perform the analysis and the function polynomial orders vary from $p = 2$ to $p = 10$. For the mesh edge that is used to geometrically discretize the ellipse, linear ($q = 1$), quadratic ($q = 2$), cubic ($q = 3$) and quartic ($q = 4$) geometric approximations are selected.

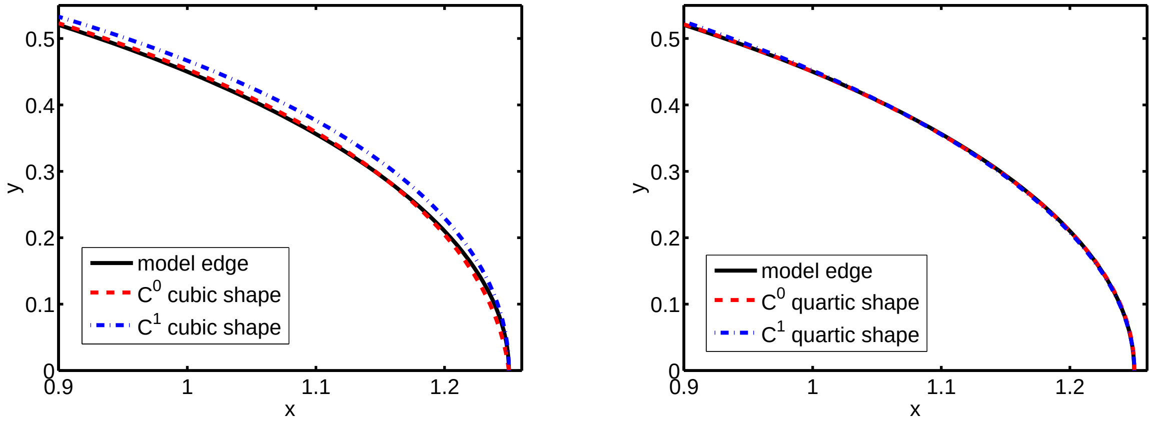
For the model with parameter $m = 0.25$, two different fitting methods are applied for the $q > 1$ cases. The first is C^0 interpolant where the interpolating points are equally spaced in the parametric space of the edge. The second enforces C^1 continuity at the vertices A and E. Close-ups of the geometric approximations in the vicinity of vertex A are shown in Figure 8.2(b)-8.2(d).

For the model with parameter $m = 0.9$, Figure 8.3(b)-8.3(d) shows the geometric approximation shapes for the mesh edge classified on the ellipse that are determined by the same two fitting methods for the model with parameter $m = 0.25$. The C^0 geometric approximation shapes defined by equal spaced interpolation provide poor geometric approximations, particular quartic shape. The quadratic and cubic C^1 geometric approximation shapes have better approximation quality than the C^0 shapes in terms of the maximum distance deviation between the geometric shapes and the model edge. However, the quartic C^1 geometric approximation shape does not show any improvement than the quartic C^0 shape. The shapes in Figure 8.3 and the analysis results in Figure 8.8 clearly demonstrate that the shapes determined by the equal spaced interpolating points in the edge parameter space are unacceptable. Alternative fitting techniques to construct geometric shapes for the sharp ellipse are needed. In this thesis, a curvature variation based method is applied in such a way that the interpolation points can be placed in the domains that have larger curvature variation.



(a) Coarse Mesh

(b) C^0 and C^1 Quadratic Approximation Shapes



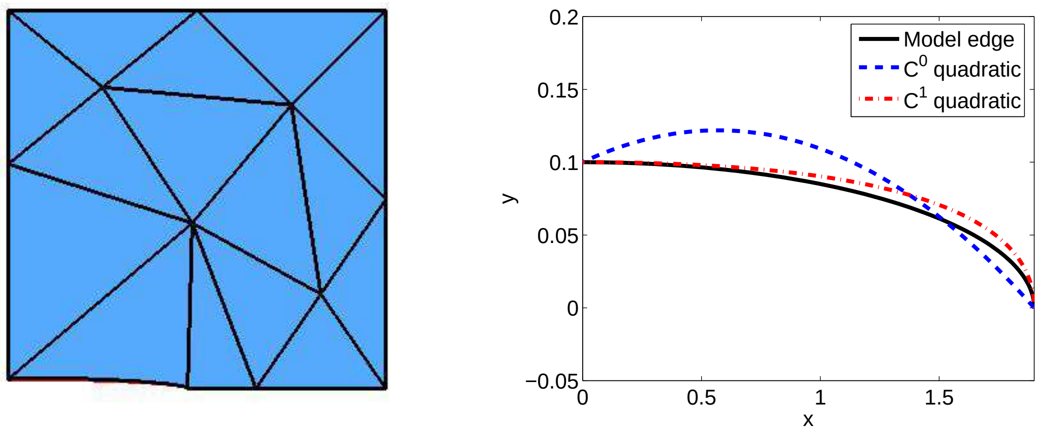
(c) C^0 and C^1 Cubic Approximation Shapes

(d) C^0 and C^1 Quartic Approximation Shapes

Figure 8.2: Mesh and Different Geometric Approximation Shapes at Point A for Model with $m = 0.25$

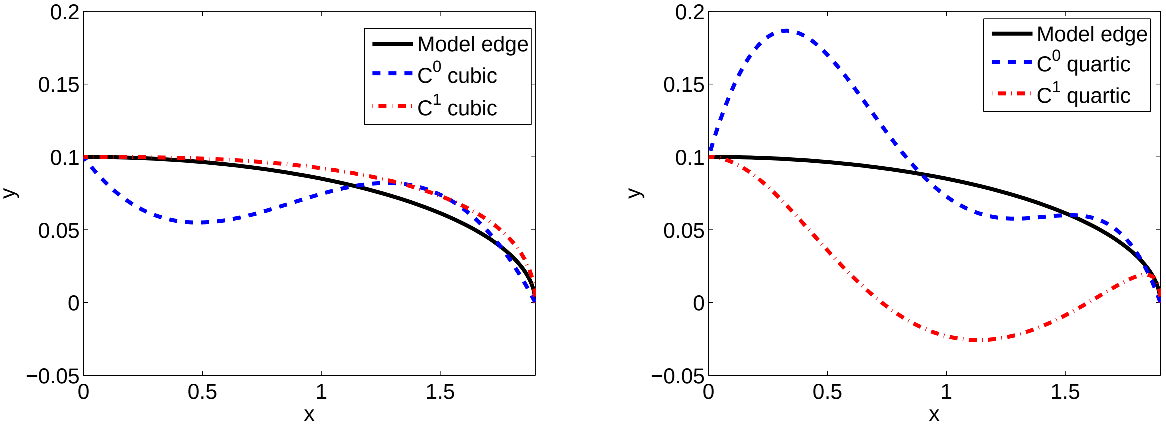
Let $S(t)$ be the curvature variation function of the model edge over the parametric space $a \leq t \leq b$ of the model edge. For the ellipse with $m = 0.9$, $S(t)$ is of the form,

$$S(t) = \frac{190(1+t^2)}{(1442t^2+1+t^4)\sqrt{\frac{(1442t^2+1+t^4)}{(1+t^2)^4}}}, \qquad 0 \leq t \leq 1. \tag{8.16}$$



(a) Coarse Mesh

(b) C^0 and C^1 Quadratic Approximation Shapes



(c) C^0 and C^1 Cubic Approximation Shapes

(d) C^0 and C^1 Quartic Approximation Shapes

Figure 8.3: Mesh and Different Geometric Approximation Shapes at Point A for Model with $m = 0.9$

For a qth order mesh edge, the set of interpolation points t_i is determined to satisfy,

$$\int_{t_i}^{t_{i+1}} S(t)dt = \frac{1}{q}\int_a^b S(t)dt, \qquad 0 \leq i \leq q-1, \tag{8.17}$$

that the curvature variation over each segment is equal. For the model with parameter $m = 0.9$, the unequal spaced interpolation points determined by the Eq. 8.17 are presented in Table 8.2 and the C^0 geometric shapes are shown in Figure 8.4 that the approximation improvements of the cubic and quartic geometric shapes to the model ellipse comparing to those C^0 shapes in Figure 8.3 are obvious. The cubic and quartic shapes are so close to the model ellipse edge that the difference between them are very small in Figure 8.4.

Table 8.2: Unequal Spaced Interpolation Points for Model with $m = 0.9$

q	t_0	t_1	t_2
q=2	0.000921		
q=3	0.000614	0.001228	
q=4	0.000461	0.000921	0.001382

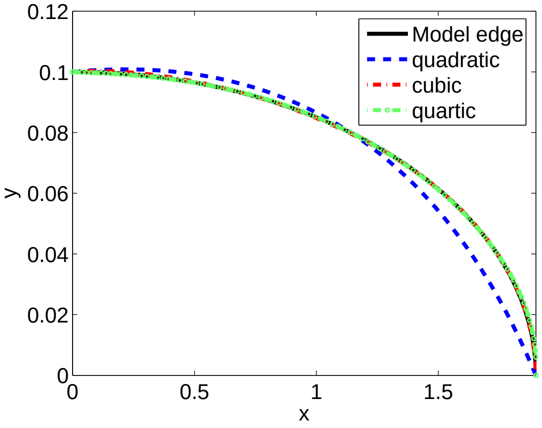
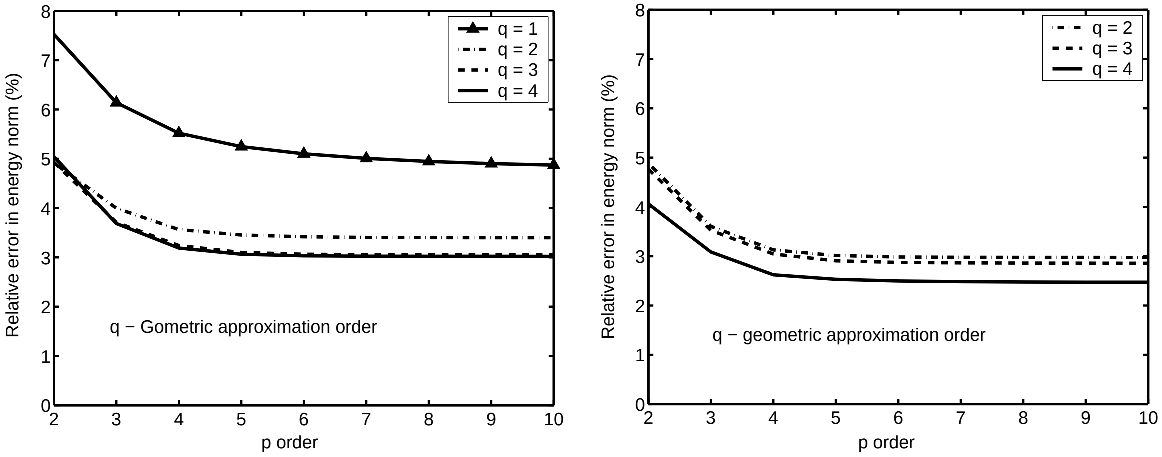


Figure 8.4: C^0 Geometric Approximation Shapes Determined by Unequal Spaced Interpolation Points for Model with $m = 0.9$

8.1.4 Analysis Results for the Model with $m = 0.25$

Convergence curves for the relative error in energy norm with respect to polynomial order p for the various geometric approximation orders q are shown in Figure 8.5(a)-8.5(b). As should be expected, the error ceases to decrease when the polynomial order increase past the geometric approximation order which is consistent with the basic theory [46]. The discretization error approaches a limit as p increases. This limit is essentially the geometric approximation error because for very high p we solve the PDE nearly exactly on an approximated geometrical domain.

The geometrical approximation error is less when the geometric approximation order increases.



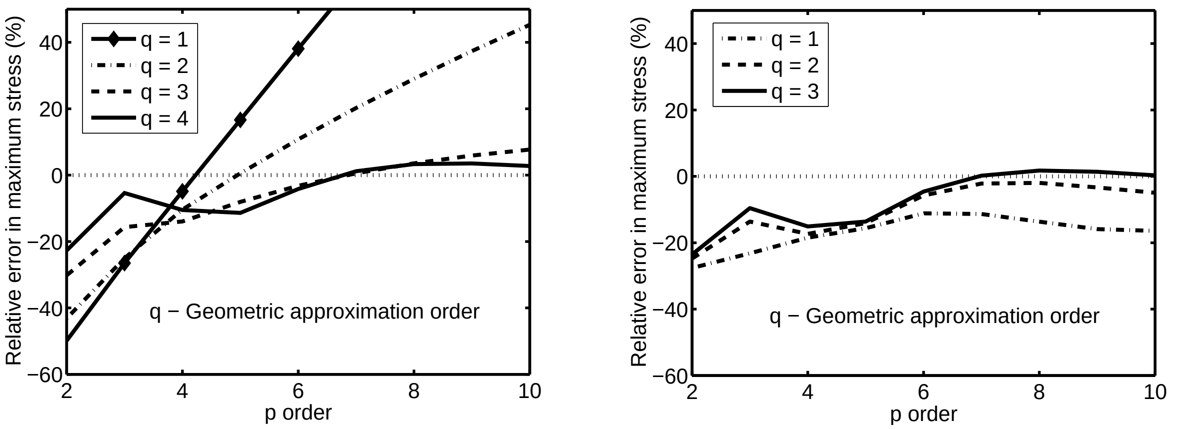
(a) Relative Error in Energy Norm for C^0 Shape

(b) Relative Error in Energy Norm for C^1 Shape

Figure 8.5: Mesh and Different Geometric Approximation Shapes

The performance of the different geometric approximations on the L^∞ norm of the maximum stress is a bit more complex. Figure 8.6(a) shows the relative error in maximum stress for the C^0 geometric approximations and Figure 8.6(b) shows it for the C^1 geometric approximations. For the linear geometric approximation ($q = 1$ for C^0) the computed maximum stress is underestimated at $p = 1$, but quickly increases past the exact value to overestimate the exact value by relative error of 122% at $p = 10$. Such behavior is expected since as p increases we are moving toward to the solution of a problem with a sharp corner at point A where the stress theoretically goes to infinity. As the geometric approximation order q is increased for the C^0 shapes, the sharpness of the slope discontinuity at vertex A is decreased and the stress results become more accurate. However, it is interesting to note that in the case of quadratic C^0 geometric approximation the stress is over-predicted by 45%.

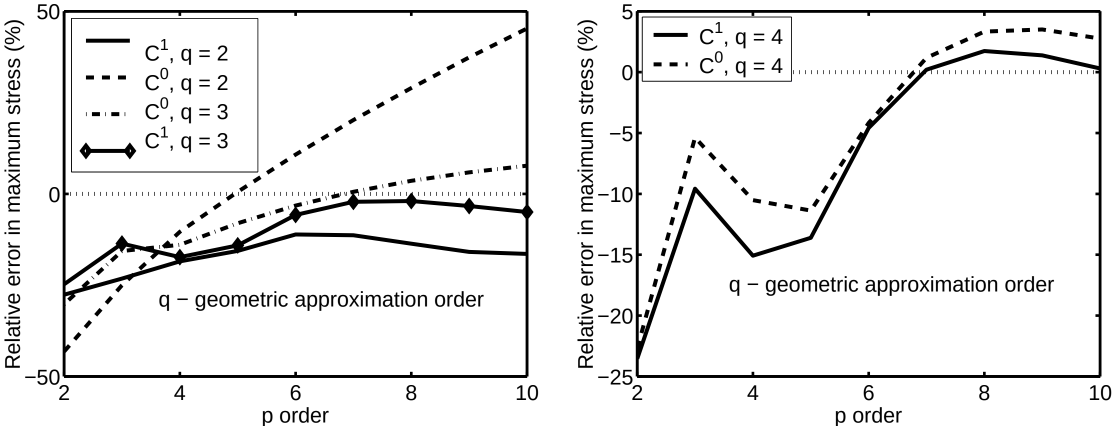
A comparison of the C^0 and C^1 geometric approximation cases indicates that both of them underestimate the exact value for low order polynomial order p. In the case of C^0 geometric approximations the stress becomes overestimated when p con-



(a) Relative Error in L^∞ Norm for C^0 Shape (b) Relative Error in L^∞ Norm for C^1 Shape

Figure 8.6: Mesh and Different Geometric Approximation Shapes

tinues to increase. In the case of C^1 geometric approximations the stress is always underestimated for $q = 2$ and $q = 3$ while the $q = 4$ does slightly overestimate the value for high p. Figure 8.7 provides a more direction comparison of the C^0 and C^1 shapes for the various geometric approximation orders.



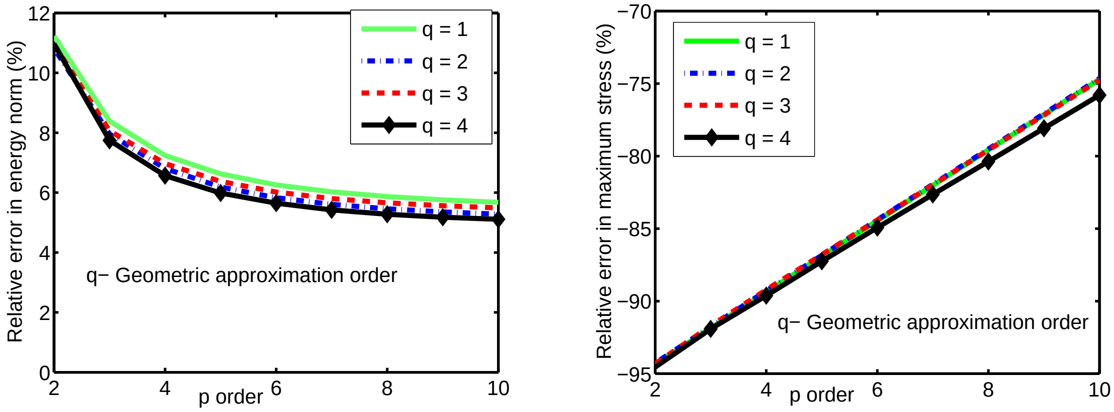
(a) Convergence Curve for $q = 2, 3$ (b) Convergence Curve for $q = 4$

Figure 8.7: Convergence Curves for C^0 and C^1 Shapes

It should noted that the geometric approximation error for the quadratic geometric approximations ($q = 2$), at $p = 10$ are substantial with an overestimate of 45% for the C^0 case and underestimate of 16% for the C^1 shape. The cubic geometric approximations ($q = 3$) yield a smaller error at $p = 10$ with an overestimate of 7.7% for the C^0 case and underestimate of 5.0% for the C^1 case. The quartic geometric approximations ($q = 4$) yield the smallest error at $p = 10$ with an overestimate of 2.8% for the C^0 case and 0.29% for the C^1 case. These results are consistent with those presented in reference [33] where the ellipse geometry was approximated using a blending function method.

8.1.5 Analysis Results for the Model with $m = 0.9$

Convergence curves for the C^0 geometric approximation shapes determined by the equal spaced interpolating points are shown in Figure 8.8. The results for the C^1 geometric approximation shapes are nearly the same that will not be repeated presented here.

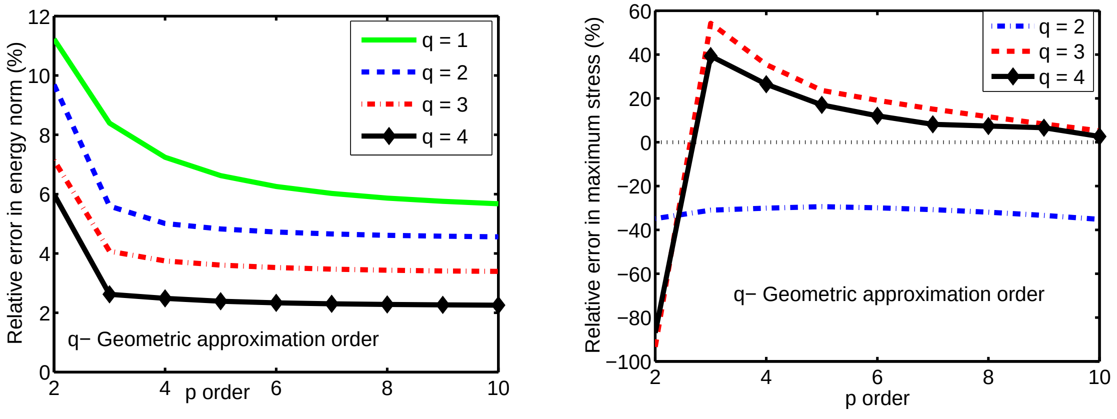


(a) Relative Error in Energy Norm

(b) Relative Error in L^∞ Norm

Figure 8.8: Convergence Curves for C^0 Shapes with Equal Spaced Interpolating Points

In the case of solution accuracy measured with the relative error in the energy norm with respect to the polynomial order, the error of the high order geometric shapes converges to the same level accuracy (5%) as the linear geometric shape. The phenomenon is expected because of the unsatisfied geometric approximations determined by the equal spaced fitting technique. In the case of the L^∞ norm, all



(a) Relative Error in Energy Norm (b) Relative Error in L^∞ Norm

Figure 8.9: Convergence Curves for C^0 Shapes with Unequal Spaced Interpolating Points

the C^0 approximation shapes underestimate the maximum stress with 75% relative error in L^∞ norm at $p = 10$.

The convergence curves for the C^1 geometric shapes determined by the curvature driven unequal spaced interpolating points are shown in Figure 8.9. The relative error in energy norm with respect to the polynomial order are shown in Figure 8.9(a). The error converges to different limits for different geometric approximation order as the polynomial order p increases, 5.8% for $q = 1$, 4.8% for $q = 2$, 3.5% for $q = 3$, and 2.2% for $q = 4$ respectively. The limits decrease as the geometric approximation order q increases that is consistent with the results for the model with $m = 0.25$.

The relative error in L^∞ norm with respect to the polynomial order are shown in Figure 8.9(b). The quadratic approximation shape always underestimates the maximum stress and the solution accuracy is -38% at $p = 10$. The cubic and quartic geometric approximation shapes have the same behavior that underestimate the exact stress at $p = 2$ then highly overestimate the exact stress at $p = 3$, 58% for the cubic shape and 40% for the quartic shape. The curves then rapidly converge to the exact stress after $p = 4$. The relative error is 6.67% for the cubic shape and 2.25% for the quartic shape at $p = 10$. Considering only one edge is used for the sharp ellipse, the results are promising.

8.2 The Application of p-Version Mesh for a 3D Model with Singularities in StressCheck

This section investigates the influence of the p-version mesh constructed using the automatic p-version mesh generation procedure presented in Part II of the thesis with the p-version analysis software StressCheck to solve 3D linear elasticity problem with singularities. The performance in terms of the relative error in energy norm with respect to the degrees of freedom and the computational cost between the p-version mesh and the conventional curved isotropic mesh is compared.

8.2.1 p-Version analysis Software StressCheck

StressCheck is a p-version finite element method based software that supports linear and nonlinear stress analysis technology. Its main features include [120]:

1. Elements - support beam, fastener, link, quadrilateral, triangle, hexahedral, prism and tetrahedral elements.

2. Error estimation - based on the extrapolation a posteriori error estimation presented in Section 7.2.2.

3. Adaptivity - StressCheck supports uniform and adaptive p-version analysis. The correction indicator is based on the stress resultant jumps between shared edges and faces defined as in Eq. 7.31-7.34.

4. Material - linear isotropic, orthotropic, anisotropic etc.; nonlinear elastic-plastic, bilinear, hyper-elastic etc.; Laminated isotropic and orthotropic layer etc..

8.2.2 Problem Description for the Model with Singularities

Figure 8.10 shows the three-dimensional model for which the top model face and the left cylindrical inner face is under uniform pressure $P = 1$ and the right cylindrical inner face is fixed. There are two singular model edges whose interior dihedral angles are 270^0 and the stress at these edges goes to infinity theoretically. The material for the problem is assumed to be linearly elastic with Young's Modulus of $2 \times 10^7 psi$ and Poisson's ratio of 0.3. Due to the symmetry, only one half of the model is solved with uniform and adaptive p-version analysis in StressCheck.

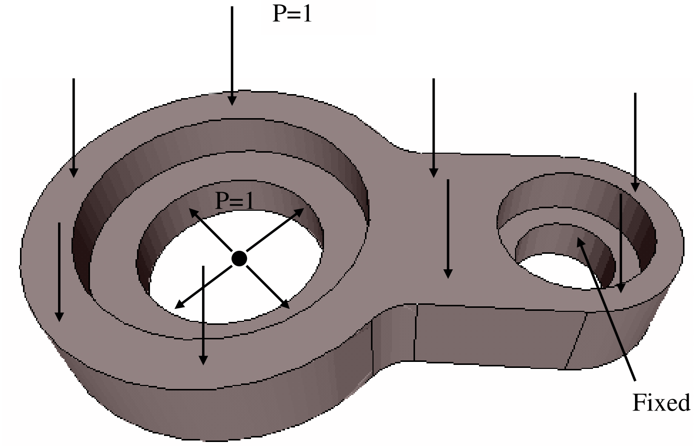


Figure 8.10: **Geometric Model Under Uniform Pressure**

Since the extrapolated solution of this problem is unknown, a finer curved graded mesh with 5481 regions and 4 graded layers towards the singular edges shown in Figure 8.11 is applied with uniform p-version method to get an estimated extrapolated potential energy $\Pi(u)$. First, the finite element potential energy $\Pi(u^h)$ at $p = 6,\ 7,\ 8$ is computed as shown in Table 8.3 and the extrapolated potential energy $\Pi(u)$ is computed using the extrapolation error estimator in Eq. 7.13.

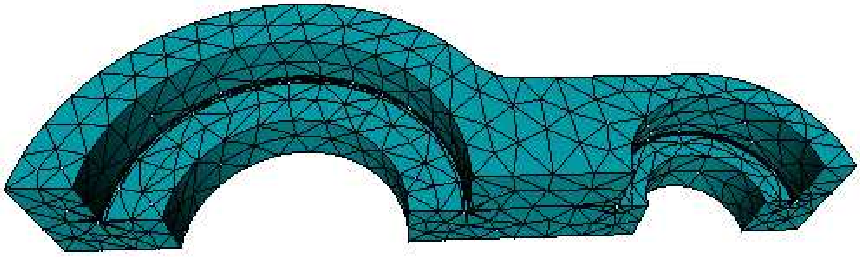


Figure 8.11: Finer Curved p-Version Mesh with 4 Layers and 5481 Regions

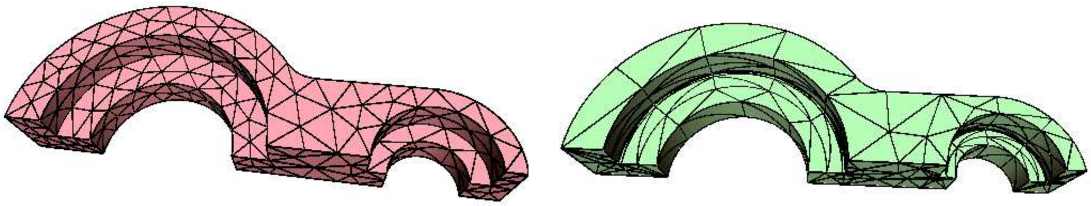
Figure 8.12 shows the two finite element meshes that will be uniformly and

Table 8.3: Computation of the Extrapolated Potential Energy $\Pi(u)$ for the Model with Singularities

	p=6	p=7	p=8
Dof	$364,254$	$623,003$	$982,085$
$\Pi(u^h)(\times 10^{-1})$	-3.73765	-3.73841	-3.73881
$\Pi(u)(\times 10^{-1})$	**-3.73954**		

adaptively solved. Figure 8.12(a) is the curved isotropic volume mesh with 1014 regions and Figure 8.12(b) is the p-version mesh with 2 graded layers and 888 regions. The problem is solved with uniform p-version method from $p = 2$ to 8 and adaptive p-version method till the maximum polynomial order of elements reaches the p-order limit 8. The solution measured in energy norm e_r is computed using the Eq. 8.15, where the extrapolated potential energy $\Pi(u)$ used is -3.73954 obtained from Table 8.3.

Both the meshes apply geometric mapping that uses a parabola to approximate the model boundaries [120]. The geometric mapping in StressCheck must obtain at least 6 sample points for each edge and 36 sample points for each face classified on the curved model boundaries [120]. Although geometric mapping approximates the model boundary more accurate than the isoparameteric mapping, geometric approximation error still exist and must be considered in analyzing the results.



(a) Curved Isotrpic Volume Mesh with 1014 Regions

(b) Curved p-version Mesh with 2 Graded Layers and 888 Regions

Figure 8.12: Finite Element Meshes

8.2.3 Analysis Results

The statistics for uniform and adaptive p-version method is shown in Table 8.4 and 8.5 respectively, where the cpu time measures the computational cost for assembling and solving the global stiffness matrix at each polynomial order or adaptive iteration step. The extrapolated potential energy $\Pi(u)$ in Table 8.4 is computed by using the data at the polynomial order $p = 6, 7, 8$ respectively for both the meshes.

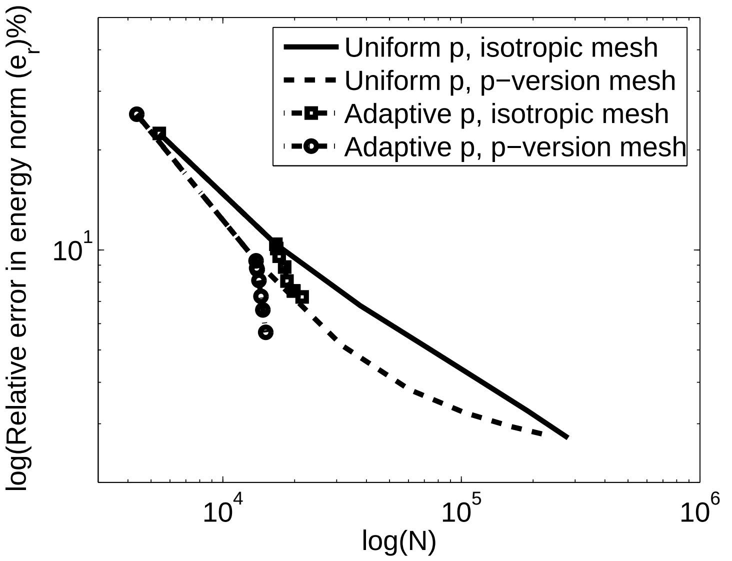
Convergence curves for the relative error in energy norm with respect to the number of degree of freedom (Dof) and computational cost (CPU) for different meshes and p-version methods are shown in Figure 8.13. The CPU time measures the computational cost starting from first step to solve the problem till the current step.

Table 8.4: Uniform p-version Analysis Results

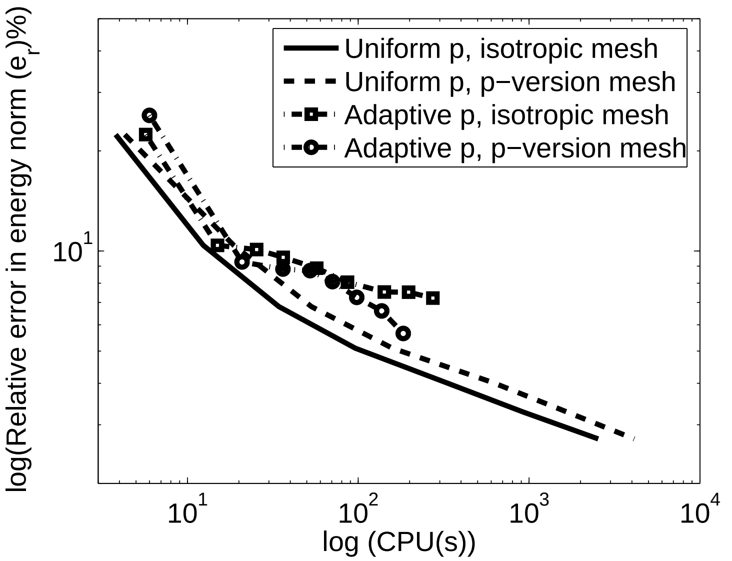
	Curved Isotropic Mesh				Curved p-version Mesh			
p	Dof	$\Pi(u^h)(-1)$	$e_r(\%)$	cpu(s)	Dof	$\Pi(u^h)(-1)$	$e_r(\%)$	cpu(s)
2	5,416	-3.55145	22.4	3.8	4,354	-3.49452	25.6	4.3
3	16,676	-3.69904	10.4	8.6	1,3781	-3.70743	9.27	14.4
4	37,693	-3.72223	6.80	21.9	31,593	-3.72957	5.16	34.5
5	71,509	-3.72983	5.10	61.8	60,454	-3.73415	3.80	105.2
6	121,166	-3.73350	4.02	230.4	103,028	-3.73562	3.24	456.4
7	189,706	-3.73551	3.28	593.1	161,979	-3.73631	2.94	1049.1
8	280,171	-3.73677	2.72	1621.6	239,971	-3.73672	2.75	2463.6
$\Pi(u)(-1)$		**-3.74032**	1.44			**-3.73769**	2.21	

The two extrapolated potential energy value computed in Table 8.4 are compared with the reference value -3.7395441 obtained from the finer mesh, the relative error is 1.44% for the curved isotropic mesh and 2.21% for the p-version mesh. Such error can be regarded as the geometric approximation error for both the meshes because the model boundaries are not represented exactly. It is expected that the curved isotropic mesh has lower relative error than the p-version mesh since the curved isotropic mesh has more mesh entities classified on the curved boundaries. The geometric approximation error is then improved.

In the case of uniform p-version method, the error for both the meshes reaches the same level of accuracy, 2.72% for the isotropic mesh and 2.75% for the p-version



(a) Relative Error in Energy Norm e_r with Respect to Dof



(b) Relative Error in Energy Norm e_r with Respect to CPU

Figure 8.13: Relative Error in Energy Norm e_r with Respect to Dof and CPU

Table 8.5: Adaptive p-version Analysis Results

	Curved Isotropic Mesh				Curved p-version Mesh			
Step	Dof	$\Pi(u^h)(-1)$	$e_r(\%)$	cpu(s)	Dof	$\Pi(u^h)(-1)$	$e_r(\%)$	cpu(s)
1	871	-3.45145	95.2	1.8	648	-3.45452	95.2	1.6
2	5,416	-3.55145	22.4	3.9	4,354	-3.49454	25.6	4.4
3	16,676	-3.69904	10.4	9.3	13,781	-3.70743	9.27	14.9
4	16,814	-3.70114	10.1	10.4	13,863	-3.71048	8.82	15.3
5	17,222	-3.70526	9.57	11.0	13,930	-3.71106	8.73	16.0
6	18,167	-3.71005	8.88	20.7	14,174	-3.71505	8.09	18.4
7	18,587	-3.71527	8.06	29.4	14,449	-3.71990	7.25	27.4
8	19,756	-3.71832	7.53	55.6	14,714	-3.72324	6.60	39.2
9	19,852	-3.71841	7.52	55.1	15,138	-3.72761	5.65	46.3
10	21,496	-3.72002	7.22	76.5				

mesh at $p = 8$. However, the p-version mesh nearly reaches its limit value 2.21% while the error for the curved isotropic mesh is still higher comparing to the limit value 1.44%. Considering the degrees of freedom used at $p = 8$, the p-version mesh uses 15% fewer DOF than the curved isotropic mesh. Convergence curve in Figure 8.13(a) indicates that the p-version mesh converges faster than the isotropic mesh and uses less than 1/2 the degrees of freedom a solution accuracy level of 5%, 3.80% at $p = 5$ with 60,454 Dof for the p-version mesh and 4.02% at $p = 6$ with 121,166 Dof for the isotropic mesh.

In the case of adaptive p-version method, the error converges very slow after 8 iteration steps for the isotropic mesh and the solution accuracy is 7.22% at the last iteration step. This phenomenon does not appear for the p-version mesh that the error keeps decreasing and terminates at a level of 5.65% at the iteration step. Considering the degrees of freedom used at the last step of the adaptive p-version analysis, the p-version mesh uses 30% fewer Dof than the isotropic mesh.

The comparison of the computational cost for the different types of meshes is a bit more complex since it depends on the software implementation, particular the solvers. The convergence curve in Figure 8.13(b) shows that the computational cost for the p-version mesh in the uniform p-version mesh is always more expensive than the isotropic mesh that is inconsistent with the convergence curve of the degrees of the freedom in Figure 8.13(a). Such contradiction is because the p-version mesh use larger aspect ratio elements to discretize the domain, particular around the singular edges. The larger aspect ratio elements worsen the conditioning of the global stiffness matrix and require the iteration solver to take longer to converge. In the case of the adaptive p-version analysis, the computational cost is consistent with the degrees of freedom curves in Figure 8.13(a) that the p-version mesh use 33% less CPU time to achieve better solution accuracy than the isotropic mesh at the last iteration step.

8.3 The Application of p-Version Mesh for a 3D Thin Section Model

StressCheck allows to independent control the polynomial order in the thickness and in-plane directions for the thin section structures discretized either by hexahedral elements or prismatic elements. In the case of hexahedral elements, any of the three independent coordinates can be selected to represent the thickness direction. In the case of prismatic elements, only the direction along the quadrilateral faces can be declared as the thickness direction [120]. In StressCheck, the polynomial order of the thickness direction is fixed at a lower value of q and the polynomial order p in the in-plane directions can be applied with uniform or adaptive p-version analysis. Since the number of degrees of freedom associated with hexahedral or prismatic elements depends on the polynomial value p and q, the different controlling of the p and q allows substantial savings in computational time without compromising the quality and accuracy of solution.

Since StressCheck does not support pyramid element, the curved 3D thin section model in this section is designed such that the p-version mesh only contains prismatic elements without long diagonal through the thickness direction. The performance of the prismatic p-version mesh when $q = p$ and $q = 1,\ 2,\ 3$ is compared with an isotropic tetrahedral mesh in terms of the solution accuracy measured in energy norm with respect to the degrees of freedom and computational cost.

8.3.1 Problem Description for the Thin Section Model

Figure 8.14 shows the curved thin section model under uniform pressure $P = 1$ while the two cylindrical holes are fixed. The material for the problem is assumed to be linear elastic with Young's Modulus of $3 \times 10^7 psi$ and Poisson's ratio of 0.3. Due to the symmetry, one half of the model is analyzed.

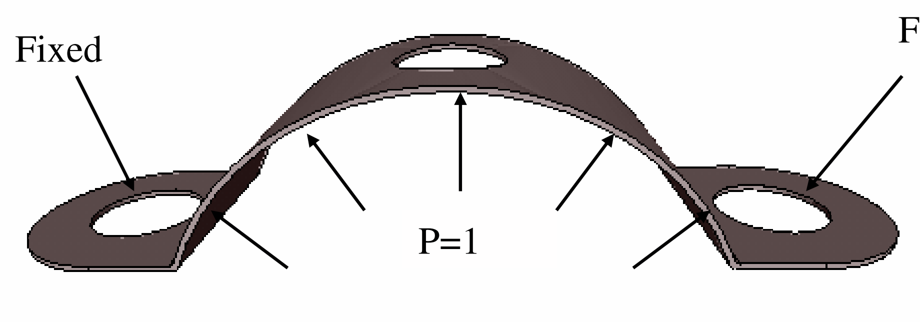


Figure 8.14: Thin Section Model

A finer curved prismatic mesh with 658 regions shown in Figure 8.15 is used to estimate the extrapolated potential energy $\Pi(u)$. Uniform p-version analysis is performed and the finite element potential energy $\Pi(u^h)$ is computed at $p = 6,\ 7,\ 8$ presented in Table 8.7. The extrapolated potential energy is computed using the extrapolation error estimator in Eq. 8.15.

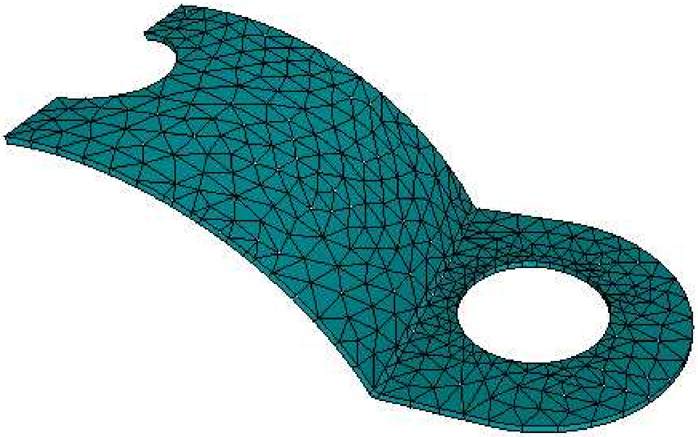


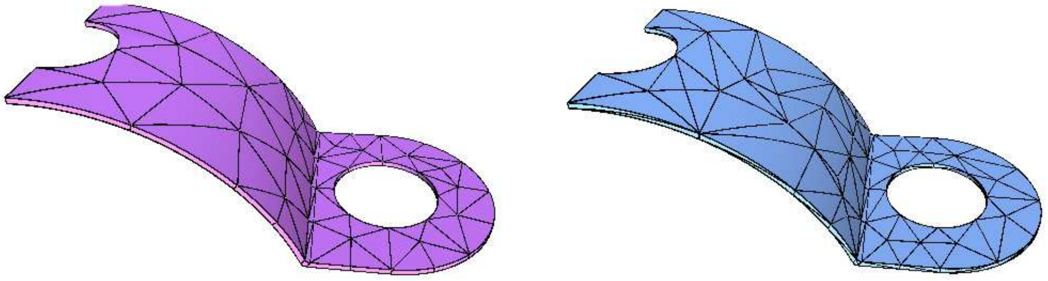
Figure 8.15: Finer Curved Prismatic Mesh with 658 Regions

Two Finite element meshes shown in Figure 8.16 are used to solve the problem. The curved prismatic mesh (left) has 80 regions generated by applying the

Table 8.6: Computation of the Extrapolated Potential Energy $\Pi(u)$ for the Thin Section Model

	p=6	p=7	p=8
Dof	$102,958$	$153,692$	$219,288$
$\Pi(u^h)$	-1.33677	-1.33773	-1.33822
$\Pi(u)$	**-1.33894**		

p-version mesh generation procedure presented in Part II of this thesis. The curved tetrahedral mesh (right) is generated by the isotropic volume mesh generator and has 297 regions.



(a) Curved Prismatic Mesh with 80 Regions

(b) Curved Tetrahedral Mesh with 297 Regions

Figure 8.16: Finite Element Meshes

The p-version analysis performed on the two meshes includes:

- Uniform p-version analysis for the tetrahedral mesh and the prismatic mesh with $q = p$ from $p = 2$ to $p = 8$.
- Fix the polynomial order $q = 1,\ 2,\ 3$ at the thickness direction for the prismatic mesh and apply uniform p-version analysis at the in-plane directions from $p = 2$ to $p = 8$.

8.3.2 Analysis Results

The statistics of the uniform p-version analysis for the tetrahedral and prismatic mesh with $q = p$ is presented in Table 8.8. Table 8.9 shows the relative error in energy norm with respect to the degrees of freedom for prismatic p-version mesh with $q = 1,\ 2,\ 3$ and Table 8.9 compares the computational costs for all of the five analysis cases. Convergence curves for relative error in energy norm with respect to the degrees of freedom and the computational cost are shown in Figure 8.17.

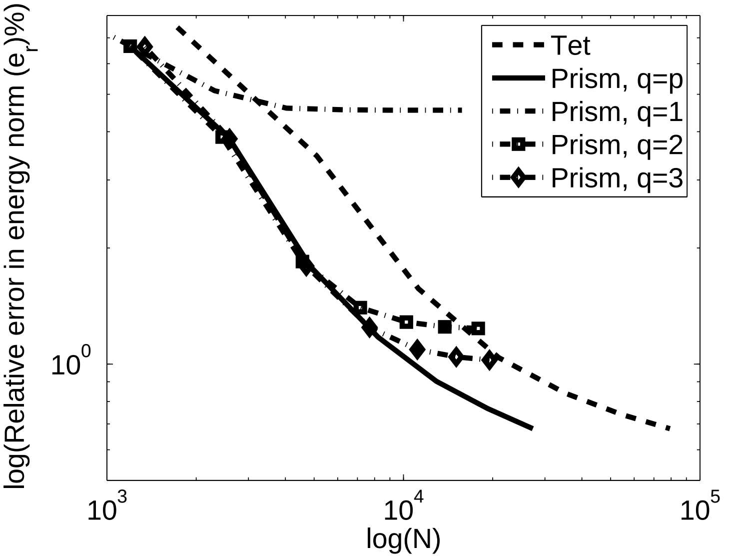
Table 8.7: Uniform p-version Analysis Results for Tet and Prism Mesh

	Curved Tetrahedral Mesh			Curved Prismatic p-version Mesh		
p	Dof	$\Pi(u^h)$	$e_r(\%)$	Dof	$\Pi(u^h)$	$e_r(\%)$
2	1,728	-0.59427	7.45	1,199	-0.74494	6.66
3	5,101	-1.17889	3.46	2,592	-1.14193	3.83
4	11,241	-1.20603	1.57	4,850	-1.29592	1.79
5	20,973	-1.32453	1.04	8,213	-1.32051	1.17
6	35,116	-1.32948	0.84	12,921	-1.32803	0.92
7	54,493	-1.33160	0.74	19,214	-1.33106	0.78
8	79,226	-1.33274	0.68	27,332	-1.33275	0.68
$\Pi(u)$		**-1.33894**			**-1.33894**	

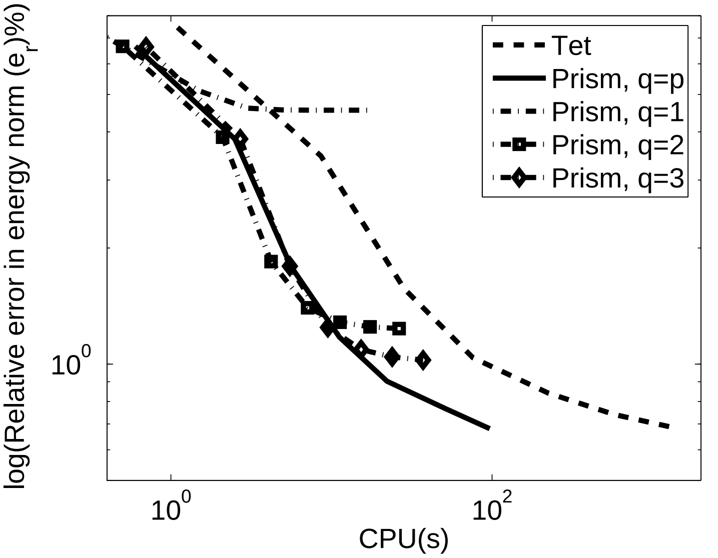
Table 8.8: Uniform p-version Analysis Results for Tet and Prism Mesh

	q=1			q=2			q=3		
p	Dof	$\Pi(u^h)$	$e_r(\%)$	Dof	$\Pi(u^h)$	$e_r(\%)$	Dof	$\Pi(u^h)$	$e_r(\%)$
2	1,056	-0.67732	7.02	1,199	-0.74494	6.66	1,342	-0.74782	6.65
3	2,306	-0.98911	5.11	2,449	-1.13765	3.87	2,592	-1.14193	3.83
4	4,036	-1.05514	4.61	4,564	-1.29338	1.84	4,704	-1.29588	1.79
5	6,246	-1.06100	4.56	7,159	-1.31274	1.40	7,687	-1.31821	1.25
6	8,936	-1.06208	4.55	10,234	-1.31686	1.28	11,147	-1.32304	1.09
7	12,106	-1.06223	4.55	13,789	-1.31802	1.25	15,087	-1.32435	1.04
8	15,756	-1.06226	4.55	17,892	-1.31846	1.24	19,507	-1.32487	1.02
$\Pi(u))$		**-1.33894**			**-1.33894**			**-1.33894**	

Since the problem does not possess any singularity caused by model geometry shape, both the tetrahedral mesh and the prismatic mesh with $q = p$ can achieve exponential convergence rate as the convergence curves shown in Figure 8.17(a) and the solution accuracy for each polynomial order p is nearly the same for these two meshes. However, the prismatic mesh uses 1/3 fewer regions than the tetrahedral mesh that substantially saves the degrees of freedom to analyze the problem. For example, the prismatic mesh use 65% fewer Dof than the tetrahedral mesh to achieve the same solution accuracy 0.68% at $p = 8$.



(a) Relative Error in Energy Norm e_r with Respect to Dof



(b) Relative Error in Energy Norm e_r with Respect to CPU

Figure 8.17: Relative Error in Energy Norm e_r with Respect to Dof and CPU for Thin Section Model

Table 8.9: The Comparison of Computational Cost (CPU(s))

	Tet Mesh		Prismatic Mesh							
			$q = p$		$q = 1$		$q = 2$		$q = 3$	
p	$e_r(\%)$	CPU	$e_r(\%)$	CPU	$e_r(\%)$	CPU	$e_r(\%)$	CPU	$e_r(\%)$	CPU
2	7.54	1.1	6.66	0.6	7.02	0.4	6.66	0.5	6.65	0.7
3	3.46	8.6	3.83	2.5	5.11	1.5	3.87	2.1	3.83	2.7
4	1.57	28.4	1.79	5.6	4.61	3.0	1.84	4.2	1.79	5.5
5	1.04	76.1	1.17	11.2	4.56	4.9	1.40	7.1	1.25	9.5
6	0.84	224.8	0.92	22.2	4.55	7.5	1.28	11.3	1.09	15.3
7	0.74	577.7	0.78	51.0	4.55	11.2	1.24	17.4	1.04	23.9
8	0.68	1450.6	0.68	96.9	4.55	17.0	1.24	26.4	1.02	37.2

As for the directional variable p-version analysis, the convergence curves in Figure 8.17(a) indicate that the error ceases to decrease when the polynomial order p past than quartic in the case that the polynomial order of the thickness direction is fixed at $q = 1$. The error remains at 4.55% that is sizable larger than the case $q = p$. In the case that the polynomial order of the thickness direction is fixed at $q = 2,\ 3$, the convergence curves have the same behavior after $p = 6$ but the convergent errors is much smaller than $q = 1$, 1.24% for $q = 2$ and 1.02% for $q = 3$. Comparing to the prismatic mesh with $q = p$, the cases with $q = 2,\ 3$ use 35% and 20% fewer degrees of freedom at $p = 8$ respectively.

The computational cost for all of the five cases is compared in Figure 8.17(b) that is consistent with the behavior of the convergence curves in Figure 8.17(a). At $p = 8$, the prismatic mesh with $q = p$ uses 95% fewer CPU time comparing the tetrahedral mesh. The prismatic mesh with $q = 2,\ 3$ save 80% and 71% fewer CPU time comparing to the case with $q = p$.

Both of the results from the models with singularity and thin sections show that the p-version mesh must be appropriately constructed and the polynomial order distribution must take the directional behavior into consideration in order to efficiently apply p-version analysis.

8.4 Directional Adaptive p-Version Method for Model with Thin Sections

This section discusses the application of directional adaptive p-version analysis in Trellis. The p-version mesh for the geometric model is generated by applying the procedure presented in Part II of this thesis.

8.4.1 Model Problem Description

The model with thin sections is shown in Figure 8.18. Due to the symmetry of the problem, only one half of the domain is analyzed by assigning properly symmetric boundary conditions. For this particular model, there are two portions of the domain can be regarded as thin sections with thickness 0.5 (marked in Figure 8.18) determined by the dimensions of the geometric model.

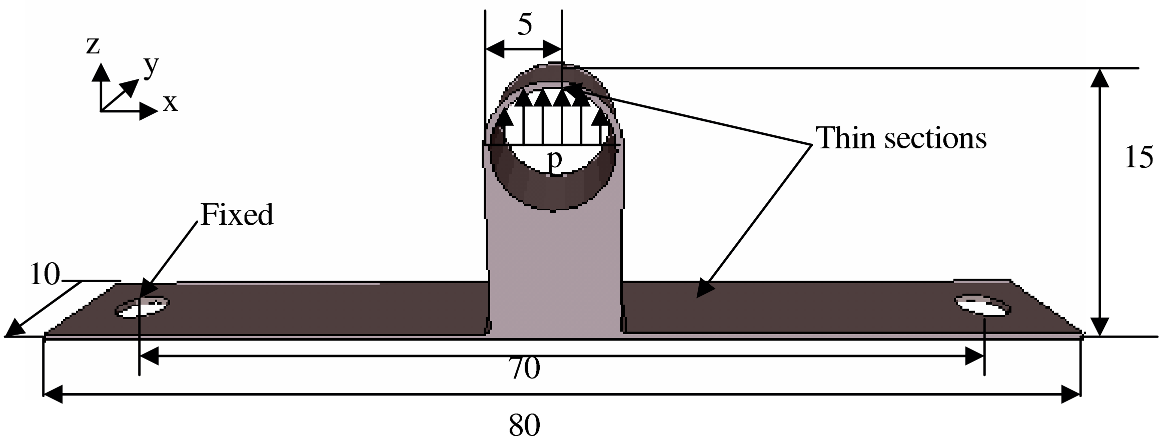


Figure 8.18: Geometric Model of the Structural Part with Thin Sections

The material is also assumed to be linearly elastic with Young's modulus $3 \times 10^7 psi$ and Poisson's ratio of 0.3. The quadratic Z directional pressure P is of the form,

$$P = 1. - (\frac{x}{4.5})^2 \tag{8.18}$$

and is applied on the inner circular surface that will cause the bending of the bottom thin sections.

Since the extrapolated solution of this problem is unknown, a finer curved mesh with 2192 tetrahedral regions is applied uniform p-version analysis in StressCheck to obtain an estimated extrapolated potential energy $\Pi(u)$. The finite element potential energy $\Pi(u^h)$ for three polynomial order $p = 6,\ 7,\ 8$ and the extrapolated potential energy computed as Eq. 8.15 is presented in Table 8.10.

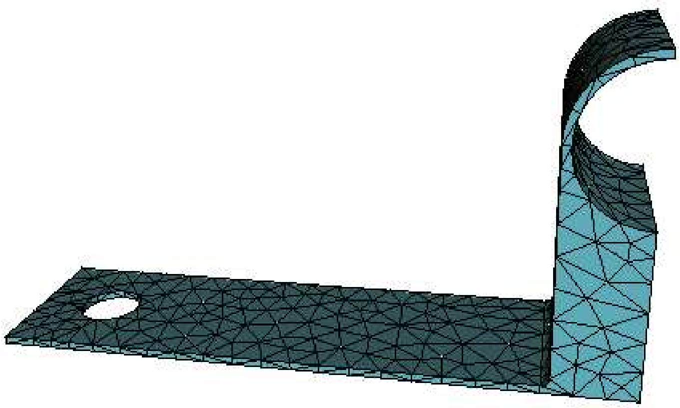


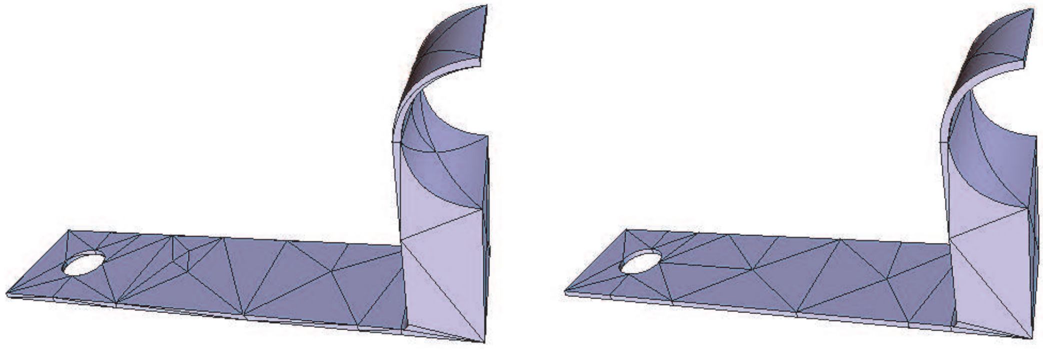
Figure 8.19: Finer Curved Mesh with 2192 Regions

Table 8.10: Computation of the Extrapolated Potential Energy $\Pi(u)$ for the Thin Section Model

	p=6	p=7	p=8
Dof	$263,281$	$411,920$	$608,026$
$\Pi(u^h)(\times 10^{-1})$	-3.78971	-3.79023	-3.79046
$\Pi(u)(\times 10^{-1})$	**-3.79062714**		

8.4.2 Finite Element Meshes for the Geometric Model

Figure 8.20 shows to curved meshes for the one half of the domain. The mesh in Figure 8.20(a) is generated by the automatic tetrahedral volume mesh generator and the mesh in Figure 8.20(b) is generated using the procedure presented in Part II that has mixed topological elements. The two thin section structures are meshed with prismatic elements without diagonal edge through the thickness directions comparing to the all tetrahedral element mesh. Table 8.11 presents the summary of these two meshes and the statistic indicates that number of elements for mixed topological mesh has been reduced almost 50% comparing to the all tetrahedral mesh.



(a) All Tetrahedral Mesh

(b) Mixed Topology Mesh

Figure 8.20: Meshes for Model with Thin Sections

The relative error e_r in energy norm is computed as,

$$e_r = \sqrt{\frac{||u - u^h||}{||u||}}\ \% = \sqrt{\frac{||e||}{||u||}}\ \% = \sqrt{\frac{\sum \eta_k}{||u||}}\ \% \tag{8.19}$$

Where η_k is the elemental residual error estimator computed based on Eq 7.28.

8.4.3 Analysis Result

The problem is analyzed using (1) Uniform p-version method that varies the polynomial order from $p = 2$ to $p = 8$ and (2) Adaptive p-version method that enriches the polynomial order independently at each coordinate direction of the elements.

Table 8.12 presents the analysis results with respect to the number of degree of freedom for uniform p-version method and 8.13 for adaptive p-version method. The convergence curves are shown in Figure 8.21

Table 8.11: Mesh Comparison for Model with Thin Sections

	Regions			Faces		Edges	Vertices
All tetrahedral	131			328		264	67
Mixed topology	66			197		194	63
	Tet	Prism	Pyramid	Tri	Quad		
	38	24	4	149	48		

In case of uniform p-version method, the mixed topology mesh can obtain more accurate solution (1.57%) than the all tetrahedral mesh (4.15%) with 45% fewer degrees of freedom. Such behavior is expected since the mixed topology mesh have 50% less elements than the all tetrahedral mesh and functional space of prismatic element is also rich than tetrahedral mesh. In case of the adaptive p-version method for the all tetrahedral mesh, the adaptive analysis stops after 6 iteration steps with an unsatisfied solution accuracy level 9.02% because some elements reached the admissible polynomial order limitation $p = 8$. For the mixed topology mesh, the directional p-version method uses 53% less degree of freedom than that used for uniform p-version method to achieve and solution accuracy that is slightly better. The highest polynomial order at the last step is $p = 7$ for the elements next to the fixed cylinder hole.

Table 8.12: Uniform p-Version Analysis Results

	All Tetrahedral Mesh		Mixed Topology Mesh	
p	Dof (N)	e_r	Dof (N)	e_r
2	890	60.42	710	47.74
3	2,566	28.86	1,699	15.39
4	5,586	17.78	2,280	12.20
5	10,343	11.61	5,939	7.39
6	17,230	7.76	9,574	4.13
7	26,640	5.56	14,483	2.16
8	38,966	4.15	20,864	1.57

Table 8.13: Directional Adaptive p-Version Analysis Results

	All Tetrahedral Mesh		Mixed Topology Mesh	
Step	Dof (N)	e_r	Dof (N)	e_r
1	890	60.42	710	47.74
2	2,635	24.02	1,925	14.35
3	2,837	22.08	2,145	10.46
4	3,371	17.06	3,536	8.87
5	4,129	11.19	5,767	4.36
6	4,432	9.02	6,768	1.97

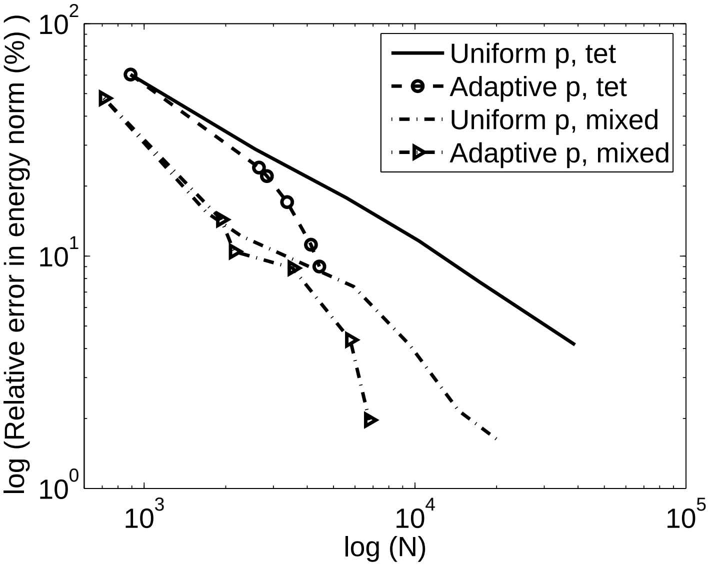


Figure 8.21: Relative Error in Energy Norm with Respect to Dof

CHAPTER 9
Conclusions

9.1 Thesis Contributions

This thesis has addressed two issues related to adaptive p-version method:

1. Automatic construction of near optimal p-version meshes characterized by (i) geometric mesh gradation towards solution singularities, (ii) controlled geometric approximation error for curved model boundaries, and (iii) Prismatic elements without long diagonals through for thin sections.
2. Directional adaptive control of polynomial order in the independent coordinates of the mesh entities.

Chapter 4 presented a procedure that automatically isolates the singular model edges and thin sections using the modeler information and a coarse surface triangulation. Cylindrical layered meshes around the singular edges and one layer prismatic thin section meshes are generated.

Chapter 5 presented a p-version mesh curving procedure that curves the mesh entities classified on the curved boundaries in an appropriate order with gradation and thin section meshes control. The procedure can produce a valid curved p-version mesh up to any required order while maintaining the geometric gradation for the singular model edges and prismatic thin section meshes. A set of curved local mesh modification operations build on the straight-sided mesh modification operations are applied in an appropriate order to incrementally correct the invalid elements caused by curving mesh entities.

The performance of the p-version meshes is investigated in the p-version analysis software StressCheck and Trellis. Chapter 6 discussed the flexible topology based directional variable p-version approximation structure that allow the assignment of different p-orders in the independent coordinates of the mesh entities in the adaptive analysis. Chapter 7 presented the automated adaptive directional variable p-version analysis procedure for 3D curved domains implemented in Trellis.

A directional correction indicator derived from the computation of the elemental residual error estimator is given such that the p-orders are adaptively enriched in the appropriate directions.

9.2 Recommendations for Future Work

Areas that can be extended from this thesis for the further research are listed as follows,

1. Efficient numerical integration schemes could be addressed for directional adaptive p-version method with curved p-version meshes.
2. The p-version adaptive analysis could be extended to fully automated hp-version adaptivity that requires a criteria to determine whether h-version or p-version refinement is performed at the adaptive analysis. In the case of h-version refinement, the maintaining of the gradation and the prismatic thin section meshes of the p-version meshes must be considered. The determination of the number of layers for the singular model edges in the adaptive analysis is also very important for the automated hp-version adaptivity.
3. The curved p-version mesh could be applied in various problems such as computational fluid dynamics and multi-scale finite element analysis.
4. The p-version mesh generator in this work only considers C^0 continuous geometric approximation shapes for the mesh entities classified on the curved model boundaries. Approaches to construct higher order continuity geometric shapes could be addressed.

Reference

1. R.L. Actis, S.P. Engelstad and B.A. Szabo, “Computational requirements in design and design certification”, *Collection of technical papers - AIAA/ASME/ASCE/AHS/ASC structures, Structural Dynamics and Materials Conference*, 1379-1389, 2002.
2. R.L. Actis and B.A. Szabo, “Analysis of bonded and fastened repairs by the p-version of the finite element method”, *Comput. Math. Applications*, v46, 1-14, 2003.
3. M. Ainsworth and J.T. Oden, ”A unified approach to a posteriori error estimation based on element residual methods”, *Numer. Math.*, v65, 23-50, 1993.
4. M. Ainsworth and B. Senior, ”Aspects of an adaptive hp-finite element method: Adaptive strategy, conforming approximation and efficient solvers”, *Comput. Methods Appl. Mech. Engrg.*, v150, 65-87, 1997.
5. M. Ainsworth and B. Senior, “An adaptive refinement strategy for hp-finite element computations”, *App. Numer. Math.*, v26, 165-178, 1998.
6. M. Anisworth. and J. T. Oden, *A posteriori error estimation in finite element analysis*, A wiley-interscience publication, 2000.
7. B. Andersson, U. Falk, I. Babuska and T.V. Petersdorff, ”Reliable stress and fracture mechanics analysis of complex components using a hp version of FEM”, *Int. J. Numer. Meth. Engng.*, v38, 2135-2163, 1995.
8. M. Aramowitz and I. A. Stegun, editors, *Handbook of Mathematical Functions*, Dover Publications Inc. New York, 1972.
9. I. Babuska and W.C. Rheinbold, “Error estimates for adaptive finite element computations”,*SIAM J. Numer. Anal.*, v15, 736-754, 1978.
10. I. Babuska and W.C. Rheinbold, ”A posteriori error estimates for the finite element method”, *Int. J. Numer. Meth. Engng.*, v12, 1597-1615, 1978.
11. I. Babuska and W.C. Rheinbold, “Analysis of optimal finite element meshes is R^1, *Mathematics of Computation*, v33(146), 435-463, 1979.
12. I. Babuska, B.A. Szabo and I.N. Katz, ”The p-version of the finite element method”, *SIAM J. Numer. Anal.*, v18(3), 515-545, 1981.

13. I. Babuska and M. Vogelius, "Feedback and adaptive finite element solution of one dimensional boundary value problem", *Numer. Math.*, v44, 75-107, 1984,

14. I. Babuska and M. Suri, "The optimal convergence rate of the p-version of the finite element method", *SIAM J. Numer. Anal.*, v24(4), 750-776, 1987.

15. I. Babuska and A. Miller, "A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimator", *Comput. Methods Appl. Mech. Engrg.*, v61, 1-40, 1987.

16. I. Babuska and B.Q. Guo, "The h-p version of the finite element method for domains with curved boundaries", *SIAM J. Numer. Anal.*, v24(4), 837-861. 1988.

17. I. Babuska and M. Suri, "The p and h-p versions of the finite element method, basic principles and properties", *SIAM Rev.*, v36(4), 578-632, 1994.

18. I. Babuska, T.V. Petersdorff and B. Andersson, "Numerical treatment of vertex singularities and intensity factor for mixed boundary value problems for the Laplace equations is $\mathbb{R}^3$", *SIAM J. Numer. Anal.*, v31(5), 1265-1288, 1994.

19. I. Babuska, G. Galoz and J.E. Osborn, "Special finite element methods for a class of second order elliptic problems with rough coefficients", *SIAM J. Numer. Anal.*, v31(4), 745-781, 1994.

20. I. Babuska, B. Andersson, B. Guo, J.M. Melenk and H.S. Oh, "Finite element method for solving problems with singular solutions", *J. Comput. Appl. Math.*, v74, 51-70, 1996.

21. I. Babuska, J.M. Melenk, "The partition of unity finite element method, *Int. J. Numer. Meth. Engng.*, v40, 727-758, 1997.

22. I. Babuska, T. Strouboulis, S.K. Gangaraj, C.S. Upadhyay, "Pollution error in the *h*-version of the finite element method and the local quality of the recovered derivatives", *Comput. Methods Appl. Mech. Engrg*, v140, 1-37, 1997.

23. I. Babuska, T. Strouboulis and K. Copps, "hp Optimization of finite element approximations: Analysis of the optimal mesh sequences in one dimension", *Comput. Methods Appl. Mech. Engrg.*, v150, 89-108, 1997.

24. I. Babuska, T. Strouboulis and S.K. Gangaraj, "A posteriori estimation of the error in the recovered derivatives of the finite element solution", *Comput. Methods Appl. Mech. Engrg.*, v150, 369-396, 1997.

25. I. Babuska, T. Strouboulis and C.S. Upadhyay, "A model study of the quality of a posteriori error estimators for finite element solutions of linear elliptic problem, with particular reference to the behavior near the boundary", *Int. J. Numer. Meth. Engng.*, v40, 2521-2577, 1997.

26. I. Babuska and B. Guo, "Optimal estimates for lower and upper bounds of approximation errors in the p-version of the finite element method in two dimensions", *Numer. Math.*, v85, 219-225, 2000.

27. U. Banerjee and M. Suri, "The effect of numerical quadrature in the p-version of the finite element method", *Mathematics of Computation*, v59(199), 1-20, 1992.

28. P.P. Basu and A. Peano, "Adaptivity in p-version finite element analysis", *J. Struct. Engng.*, v109, 2310-2324, 1983.

29. M.S. Bazaraa, H.D. Sherali and C.M. Shetty, *Nonlinear programming, theory and algorithms*, Wiley-interscience, 1993.

30. M.W. Beall and M.S. Shephard, "A general topology-based mesh data structure", *Int. J. Numer. Meth. Engng.*, v40(9), 1573-1596, 1997.

31. M.W. Beall, "An Object-Oriented Framework for the Reliable Automated Solution of Problems in Mathematical Physics", *ph.D Thesis*, Rensselaer Polytechnic Institute, 1999.

32. C. Bernardi, "Optimal finite element interpolation on curved domain", *SIAM J. Numer. Anal.*, v26(5), 1212-1240, 1989.

33. E. Bertoti and B.Szabo, "Optimal selection of polynomial degrees on a finite element mesh", *Int. J. Numer. Meth. Engng.*, v42, 561-578, 1998.

34. E. Bertoti, "Dual-mixed p and hp finite elements for elastic membrane problems", *Int. J. Numer. Meth. Engng.*, v53, 3-29, 2002.

35. D. Cardoze, G. Miller, M. Olah and T. Phillips, "A Bezier-Based Moving Mesh Framework for Simulation with Elastic Membranes", *Proceedings of the 13th Meshing Roundtable, Sandia National Laboratories*, 343-354, 2004.

36. P. Carnevali, R.B. Morris, Y. Tsuji and G. Taylor, "New basis functions and computational procedures for p-version finite element analysis", *Int. J. Numer. Meth. Engng.*, v36, 3759-3779, 1993.

37. Q. Chen and I. Babuska, "Approximation optimal points for polynomial interpolation of real functions in an interval and in a triangle", *Comput. Methods Appl. Mech. Engrg.*, v128., 405-417, 1995.

38. Q. Chen and I. Babuska, “The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron”, *Comput. Methods Appl. Mech. Engrg.*, v137, 89-94, 1996.

39. S.D. Connell and M.E. Braaten, “Semistructured mesh generation for three dimensional Navier-Stokes calculations”, *AIAA Journal*, v33(6), 1017-1024, 1995.

40. A. Cote and F. Charron, “On the selection of p-version shape functions for plate vibration problem”, *Computer & Structures*, v79, 119-130, 2001.

41. A. Csabai and P. Xirouchakis, “On the medial surface approximations of extrusions”, *Engineering with Computers*, v20, 65-74, 2004.

42. L. Demkowicz, J.T. Oden and Ph. Devloo, “On h-type mesh refinement strategy based on a minimization of interpolation error”, *Comput. Methods Appl. Mech. Engrg.*, v53, 67-89, 1985.

43. L. Demkowicz, J.T. Oden, W. Rachowicz and O. Hardy, “Toward a universal h-p adaptive finite element strategy Part 1: Constrained approximation and data structure”, *Comput. Methods Appl. Mech. Engrg.*, v77, 79-112, 1989.

44. L. Demkowicz, W. Rachowicz and Ph. Devloo, “A fully automatic hp-adaptivity”, *J. Scientific Computing*, v17(1-4), 117-142, 2002.

45. S. Dey, M. S. Shephard, and J. E. Flaherty, ”Geometry-based issues associated with p-version finite element computations”, *Comput. Methods Appl. Mech. Engrg.*, v150, 39-55, 1997.

46. S. Dey, “Geometry-Based Three Dimensional *hp* Finite Element Modelling and Computations”, *Ph.D thesis*, Rensselaer Polytechnic Institute, 1997.

47. S. Dey, R.M. O’Bara and M.S. Shephard, ”Curvilinear mesh generation in 3D”, *Computer-Aided Design*, v33, 199-209, 2001.

48. M.R. Dorr, “The approximation theory for the p-version of the finite element method”, *SIAM J. Numer. Anal.*, v21(6), 1180-1207, 1984.

49. M.R. Dorr, “The approximation of solutions of the elliptic boundary-value problems via the p-version of the finite element method”, *SIAM J. Numer. Anal.*, v21(6), 58-77, 1984.

50. R.J. Donaghy, C.G. Armstrong and M.A. Price, “Dimensional reduction of surface models for analysis, *Engineering with Computers*, v16, 24-35, 2000.

51. C.A. Duarte and J.T. Oden, “An hp adaptive method using clouds”, *Comput. Methods Appl. Mech. Engrg.*, v139, 237-273, 1996.

52. C.A. Duarte and J.T. Oden, “hp clouds - an hp meshless method” *Numer. Meth. Partial Diff. Equations*, v12, 673-705, 1996.

53. C.A. Duarte, *The hp cloud method*, PhD thesis, University of Texas at Austin, 1996.

54. C.A. Duarte, I. Babuska and J.T. Oden, “Generalized finite element methods for three-dimensional structural mechanics problems”, *Computers & Structures*, v77, 215-232, 2000.

55. C.A. Duarte and I. Babuska “Mesh independent p-orthotropic enrichment using the generalized finite element method”, *Int. J. Numer. Meth. Engng.*, v55, 1477-1492, 2002.

56. D.A. Dunavant and B.A. Szabo, “A posteriori error indicators for the p-version of the finite element method”, *Int. J. Numer. Meth. Engng.*, v19, 1851-1870, 1983.

57. A. Duster, H. Broker and E. Rank, “The p-version of the finite element method for three dimensional curved thin walled structures, *Int. J. Numer. Meth. Engng.*, v52,673-703. 2001.

58. G. Farin, *Curved and Surfaces for Computer Aided Geometric Design*, Academic press, 1992.

59. R.T. Farouki and V.T. Rajan, ”Algorithms for polynomials in Bernstein”, *Comput. Aid. Geom. Des.*, 5, 1-26, 1998.

60. R. Garimella, M.S. Shephard, ”Boundary layer mesh generation for viscous flow simulations in complex geometric domains”, *Int. J. Numer. Meth. Engng.*, v49(1-2), 193-218. 2000.

61. M. K. Georges, *Automated adaptive modeling system for the h-p version of the finite element method*, ph.D thesis, Rensselear Polytechnic Institute, 1989.

62. P.L. George, F. Hecht and E. Saltel, “Fully automatic mesh generator for 3D domains of any shape”, *Impact of Com. in Sci. and Engng*, v2, 187-218, 1990.

63. P.L. George, “Improvements on Delaunay-based three-dimensional automatic mesh generator”, v28(4), 297, *J. Raman Spectroscopy*, 1997.

64. P.L. George, H. Borouchaki and E. Saltel, “”Ultimate” robustness in meshing an arbitrary polyhedron”, *Int. J. Numer. Meth. Engng.*, v58(7), 1061-1089, 2003.

65. W.J. Gordon and C.A. Hall, “Transfinite element methods: Blending-funtion interpolation over arbitrary curved element domains”, *Numer. Math.*, v21, 109-129, 1973. .

66. W. Gui and I. Babuska, "The h, p and h-p version of finite element method in one dimension. Part 1. The error analysis of the p-version. Part 2. The error analysis of the h and h-p versions. Part 3. The adaptive h-p version", *Numer. Math.* v49, 577-683, 1986.

67. B. Guo and I. Babuska, "The h-p version of the finite element method. Part 1. The basic approximation results. Part 2. General results and applications", *Comput. Methods Appl. Mech. Engrg.*, v24. 203-220, 1986.

68. P. Hager and N.E. Wiberg, “Error estimation and h-adaptivity for eigenfrequency analysis of plates in bending numerical results”, *Computers & Structures*, v78, 1-10, 2000.

69. R. Harber, M.S. Shephard, J.F. Abel, R.H. Gallagher and D.P. Greenberg, "A General Two-Dimensional, Graphical Finite Element Preprocessor Utilizing Discrete Transfinite Mappings", *Int. J. Numer. Meth. Engng.*, v17, 1015-1044, 1981.

70. A. Huerta, A. Rodriguez-Ferran, P. Diez and J. Sarrate, “Adaptive finite element strategies based on error assessment”,*Int. J. Numer. Meth. Engng.*, v46, 1803-1818,1999.

71. T.J.R. Hughes, *The finite element method, linear static and dynamic finite element analysis*, N.J. Prentice-Hall, 2000.

72. P. Ingelstrom and A. Bondeson, “Goal-oriented error estimation and h-adaptivity for Maxwell’s equations”, *Comput. Methods Appl. Mech. Engrg.*, v192, 22-24, 2003.

73. Y. Kallinderis and S. Ward, “Prismatic grid generation for three-dimensional complex geometries”, *AIAA Journal*, v31(10), 1850-1856, 1993.

74. G.E. Karniadakis and S.J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, 1999.

75. A. Khawaja, H. McMorris and Y. Kallinders, “Hybird grids for viscous flows around complex 3-D geometries including multiple bodies”, *AIAA-95-1685*, 424-441, 1995.

76. C. G. Kim and M. Suri, “On the p-version of the finite element method in the presence of numerical integration”, *Numer. Meth. Partial. Diff. Equations*, v9, 593-629. 1993.

77. G. Kiralyfalvi and B. Szabo, “Quasi-regional mapping for the p-version of the finite element method”, *Finite Elements in Analysis and Design*, v27, 85-97, 1997.

78. J. Kurtz and C. Xenophontos, “One the effects of using curved element in the approximation of the Reissuer-Mindlin plate by the p-version of the finite element method”, *Appl. Numer. Math.*, v46, 231-246, 2003.

79. C.K. Lee and Q.X. Xu, “A new automatic adaptive 3D solid mesh generation for thin-walled structures”, *Int. J. Numer. Meth. Engng.*, v62, 1519-1558, 2005.

80. M. Lenoir, “Optimal isoparametric finite elements and error estimates for domains involving curved boundaries”, *SIAM J. Numer. Anal.*, v23(3), 562-580, 1986.

81. X.R. Li, J.F. Remacle, N. Chevaugeon and M.S. Shephard, “Ainsotropic Mesh Gradation Control”, *Proceedings of the 11th Meshing Roundtable, Sandia National Laboratorie*, 2002.

82. X.R. Li, M.S. Shephard and M.W. Beall, “Accounting for curved domains in mesh adaptation, *Int. J. Numer. Meth. Engng.*, v150, 247-276, 2003.

83. J.H. Liu and K.S. Surana, “Piecewise hierarchical p-version curved shell element for geometrically nonlinear behavior of laminated composite plates and shells”,*Computers & Structures*, v55(1), 47-66, 1995.

84. R. Lohner, “Matching semi-structured and unstructured grids for Navier-Stokes calculations”, *AIAA-93-3348-CP*, 1995.

85. X.J. Luo, M.S. Shephard, J.F. Remacle, R.M. O’Bara, M.W. Beall, B.A. Szabo, R. Actis, ”p-version mesh generation issues”, *Proceedings of the 11th Meshing Roundtable, Sandia National Laboratories*, 343-354., 2002.

86. X.J. Luo, M.S. Shephard, R.M. Obara, R. Nastasia and M.W. Beall, “Automatic p-version mesh generation for curved domains”, *Engineering with Computers*, v20, 265-285, 2004.

87. J.M. Melenk and I. Babuska, “The partition of unity finite element method: basic theory and applications”, *Comput. Methods Appl. Mech. Engrg.*, v139, 289-314, 1996.

88. N.I. Muskhelishvili, *Some Basic Problems of the Mathematical Theory of Elasticity*, Groningen, P. Noordhoff, 1963.

89. P. Nithiarasu and O.C. Zienkiewicz, “Adaptive mesh generation for fluid mechanics problems”, *Int. J. Numer. Meth. Engng.*, v 47(1-3), 629-662, 2000.

90. J.T. Oden, L. Demkowicz, W. Rachowics and T.A. Westermann , ”Toward a universal h-p adaptive finite element strategy Part 2: A posteriori error estimation”, *Comput. Methods Appl. Mech. Engrg.*, v77, 113-180, 1989.

91. J.T. Oden and L. Demkowiz, "h-p adaptive finite element methods in computational fluid dynamics", *Comput. Methods Appl. Mech. Engrg.*, v89, 11-40, 1991.

92. J.T. Oden, A. Patra and Y.S. Feng, "An adaptive h-p strategy", in A.K. Noor, ed., *Adaptive Multilevel and Hierarchical Computational Strategies*, v157, 23-46, 1992.

93. J.T. Oden, "Optimal h-p finite element methods", *Comput. Methods. Appl. Mech. Engrg.*, v112, 309-331,1994.

94. J.T. Oden, W. Wu and M. Anisworth, "Three step h-p adaptive strategy for the incompressible Navier Stokes equations", in I. Babuska et al., eds, *Modeling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations*, 347-366, 1995.

95. J.T. Oden, C.A. Duarte and O.C. Zienkiewics, "A new cloud-based hp finite element method", *Comput. Methods. Appl. Mech. Engrg.*, v153, 117-126, 1998.

96. N.M. Patrikalakis and H.N. Gursory, "Shape interrogation by medial axis transform, *ASM Advances in Design Automation, 16th design automation conf., Chicago*, v23, 77-88,1990.

97. S. Pirzadeh, "Viscous unstructured three dimensional grids by the advancing-layers method", *Proceedings of the 32nd Aerospace Sciences Meething and Exhibit, AIAA-94-0417*, 1994.

98. M.A. Price, C.G. Armstrong and M.A. Sabin, "Hexahedral mesh generation by medial surface subdivision: I. Solids with convex edges, *Int. J. Numer. Meth. Engng.*, v38, 3335-3359, 1995.

99. M.A. Price and C.G. Armstrong, "Hexahedral mesh generation by medial surface subdivision: II. Solids with flat and concave edges, *Int. J. Numer. Meth. Engng.*, v40, 111-136, 1997.

100. W.R. Quadros and K. Shimada, "Hex-layer:layered all-hex mesh generation on thin section solid via chordal surface transformation", *Proceeding of the 11th International Meshing Roundtable*, 169-180, 2002.

101. W. Rachowics, J.T. Oden and L. Demkowicz, "Toward a universal h-p adaptive finite element strategy Part 3: Design of h-p meshes", *Comput. Methods Appl. Mech. Engrg.*, v77, 181-212, 1989.

102. E. Rank and I. Babuska, "An expert system for the optimal mesh design in the hp-version of the finite element method", *Int. J. Numer. Meth. Engng.*, v24, 2087-2016,1987.

103. E. Rank, R. Krause and K. Preusch, "On the accuracy of p-version elements for the Reissner-Mindlin plate problem", *Int. J. Numer. Meth. Engng.*, v43,51-67. 1998.

104. E. Rank, A. Duster, V. Nubel, K. Preusch and O.T. Bruhns, "High order finite elements for shells", *Comput. Methods Appl. Mech. Engrg.*, v194, 2494-2512, 2005.

105. J.N. Reddy, *Theory and analysis of elastic plates*, Taylor & Francies, 1999.

106. J.F. Remacle, S.S. Frazao, X.R. Li and M.S. Shephard, "Adaptive discountinous Galerkin method for the shallow water equations", *Int. J. Numer. Meth. Fluids*, v00, 1-20, 2003.

107. W.C. Rheinboldt, "On a theory of mesh-refinement processes", *SIAM J. Numer. Anal.*, v17(6), 766-788, 1980.

108. A.Z. Salem, S.C. Canann and S. Saigal, "Mid-Node admissible spaces for quadratic triangular 2D finite elements with one edge curved", *Int. J. Numer. Meth. Engng.*, v50, 181-197, 2001.

109. A.Z. Salem, S. Saigal and S.A. Canann, "Mid-Node admissible space for 3D quadratic tetrahedral finite elements", *Engineering with Computers*, v17, 39-54, 2001.

110. A. Schmidt and K.G. Siebert, "A posteriori estimators for the h-p version of the finite element method in 1D", *Appl. Numer. Math.*, v35, 43-66, 2000.

111. D. Schotzau and C. Schwab and R. Stenberg, "Mixed hp-FEM on anisotropic meshes II: Hanging nodes and tensor products of boundary layer meshes", *Numer. Math.*, v83, 667-697, 1999.

112. C. Schwab, M. Suri and C. Xenophontos, "The hp finite element method for problems in mechanics with boundary layers", *Comput. Methods Appl. Mech. Engrg.*, v157, 311-333, 1998.

113. D. Sharov and K. Nakahashi, "Hybird prismatictetrahedral grid generation for viscous flow applications, *AIAA Journal*, v36(2), 157-162, 1998.

114. M.S. Shephard and M.K. George, "Reliability of automatic 3D mesh generation", *Comput. Methods Appl. Mech. Engrg.*, v101, 443-462, 1992.

115. M.S. Shephard, S. Dey, and J.E. Flaherty, "A straight forward structure to construct shape functions for variable p-order meshes", *Comput. Methods Appl. Mech. Engrg.*, v147, 209-233, 1997.

116. M.S. Shephard, M.B. Beall, R.M. O'Bara, B.E. Webster, "Toward simulation-based design", *Finite Elements in Anal. Des.*, v40, 1575-1598, 2004.

117. S.J. Sherwin and G.E. Karniadakis, “A triangular spectral element method: applications t incompressible Navier-Stokes equations”, *Comput. Methods Appl. Mech. Engrg.*, v123, 189-229, 1995.

118. S.J. Sherwin and G.E. Karniadakis, “Tetrahedral hp finite elements: algorithms and flow simulations”, *J. Comput. Phys.*, v124, 14-45, 1996.

119. S. J. Sherwin and J. Peiro, ”Mesh generation in curvilinear domains using high order elements”, *Int. J. Numer. Meth. Engng.*, v53, 207-223. 2002.

120. StressCheck User’s Manual, ESRD, 2005.

121. T. Strouboulis, I. Babuska, D.K. Datta, K. Copps and S.K. Gangaraj, “A posteriori estimation and adaptive control of the error in the quantity of interest. Part I. A posteriori estimation of the error in the von Mises stress and stress intensity factor”, *Comput. Methods Appl. Mech. Engrg.*, v181, 261-294, 2000.

122. B.A. Szabo, “Mesh design for the p-version of the finite element method”, *Comput. Methods Appl. Mech. Engrg.*, v55, 181-197, 1986.

123. B.A. Szabo and G.J. Sahrmann, “Hierarchic plate and shell models based on p-extension”, *Int. J. Numer. Meth. Engng.*, v26, 1855-1881, 1988.

124. B.A. Szabo and I. Babuska, *Finite Element Analysis*, John Wiley & Sons New York, 1991.

125. B.A. Szabo and Z. Yosibash, “Superconvergent extraction of flux intensity factors and first derivatives from finite element solution”, *Comput. Methods Appl. Mech. Engrg.*, v129, 349-370, 1996.

126. B.A. Szabo and G. Kiralyfalvi, “Linear models for buckling and stress-stiffening”, *Comput. Methods Appl. Mech. Engrg.*, v171, 43-59, 1999.

127. N.E. Wiberg, P. Hager, “Error estimation and adaptivity for *h*-version eigenfrequency analysis”, *Stud. Appl. Mech.*, v47, 461-475, 1998.

128. S. Yamakawa and K. Shimada, “ Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells”, *Int. J. Numer. Meth. Engng.*, v57(15), 2099-2129, 2003.

129. L.Z. Yin, X.J. Luo and M.S. Shephard, “Identifying and meshing thin sections of 3-D cuSrved domains”, *submit to Proceedings of the 14th Meshing Roundtable*, 2005.

130. Z. Yosibash and B. Szabo, “Convergence of Stress Maxima in Finite element computations”, *Comm. Numer. Meth. Engng.*, v10. 683-697, 1994.

131. Z. Yosibash and B. Szabo, "The solution of axisymmetric problems near singular points and computation of stress intensity factors", *Finite Element in Analysis & Design*, v19, 115-129, 1995.

132. Z. Yosibash, "On solutions of two-dimensional linear elastostatic and heat-transfer problems in the vicinity of singular points", *Int. J. Solids Structures*, v34(2), 243-274, 1997.

133. Z. Yosibash, "Computing singular solutions of elliptic boundary elliptic boundary value problems in polyhedral domains using the p-FEM", *Appl. Numer. Math.*, v33, 71-93, 2000.

134. F.X. Zgainski, J.L. Coulomb and Y. Marechal, "A new family of finite elements: The pyramidal elements", *IEE Transactions on Magnetics*, v32(3), 1393-1396. 1996.

135. J.Z. Zhu and Z. Zhang, "The relationship of some a posteriori estimators", *Comput. Methods Appl. Mech. Engrg.*, v176, 463-475, 1999.

136. O.C. Zienkiewicz and J.Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part 1. The recovery technique", *Int. J. Numer. Meth. Engng.*, v33, 1331-1364, 1992.

137. O.C. Zienkiewicz and J.Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part 2. Error estimates and adaptivity", *Int. J. Numer. Meth. Engng.*, v33, 1365-1382, 1992.