

ADAPTIVE ANALYSIS OF THIN-WALLED STRUCTURES USING THREE DIMENSIONAL SOLID ELEMENTS



XU QIANG XUN

**SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING
NANGYANG TECHNOLOGICAL UNIVERSITY**

Adaptive Analysis of Thin-Walled Structures Using Three Dimensional Solid Elements

XU QIANG XUN

School of Civil and Environmental Engineering

**A thesis submitted to Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy**

2006

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my supervisor, Associate Professor Lee Chi King, for his continuous and patient instruction and encouragement during the entire period of my research study at Nanyang Technological University. His guidance, diligence, patience and hard work helped me to bring this research work to the final form.

Deep gratitude is given to Associate Professor Lee Chi King and Dr. Lee Young Kyu for their computer programs on surface mesh generation. Special thanks are also given to Profs. S. H. Lo, W. J. Chen and K. Y. Sze for their help on the special elements.

Many thanks are given to my senior, Mr. Shuai Ying Yong, for his beneficial suggestions and warm help in my research.

Great appreciation is given to the author's family and his friends for their constant support.

Finally, thanks are also given to the School of Civil & Environmental Engineering, Nanyang Technological University for providing the conducive research environment and financial support to carry out this study.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
TABLE OF CONTENTS	ii
SUMMARY	vii
LIST OF FIGURES	ix
LIST OF TABLES	xv
LIST OF SYMBOLS	xviii
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Objective	3
1.3 Scope	6
1.4 Computational environment	9
1.5 Organization	10
CHAPTER 2 LITERATURE REVIEW	12
2.1 A Brief Review of the Finite Element Method	12
2.1.1 History of FEM	12
2.1.2 Basis of FEM	13
2.2 A Brief Review of the Automatic Mesh Generation	15
2.2.1 Surface mesh generation	15
2.2.1.1 Parametric mapping	16
2.2.1.2 Direct 3D generation	17
2.2.2 3D mesh generation	18
2.2.3 Mesh post-processing	19
2.2.3.1 Mesh smoothing	20
2.2.3.2 Mesh cleanup	20
2.2.3.3 Mesh refinement	20

2.3	A Brief Review of the Error Estimation and Adaptive Refinement	22
2.3.1	A priori error estimation	23
2.3.2	A posteriori error estimation	24
2.3.3	Adaptive Mesh Refinement	25
2.4	Closure	27
CHAPTER 3 SURFACE MESH EXTRUSION		28
3.1	Some reviews on surface mesh generation	29
3.1.1	A brief review on NURBS surface	29
3.1.2	A brief review on metric specification	32
3.2	Extrusion for Single Surface	33
3.3	Treatments for Surfaces Intersections	38
3.3.1	Simple surface intersections	39
3.3.1.1	Re-definition of surface normal vectors for simple surface intersection	41
3.3.1.2	Adjustment of normal sides near surface intersections	45
3.3.2	Multi-cross intersections	49
3.3.3	Corner point intersections	50
3.4	Mesh Extrusion Examples	53
3.5	Closure	58
CHAPTER 4 SOLID MESH GENERATION		59
4.1	Volume Mesh Refinement	60
4.1.1	Refinement parameters used	60
4.1.2	Refinement procedures along the thickness direction	62
4.1.2.1	Connection of dividing nodes for adjacent normal sides	62
4.1.2.2	Formation of 3D solid elements	64
4.2	Mesh Quality Measurement and Enhancement	72
4.2.1	Some reviews on mesh measurement	72
4.2.2	Volume mesh quality measurement and normal sides smoothing	75

4.2.2.1	Facet shape quality measurement	75
4.2.2.2	Normal direction quality measurement	76
4.2.2.3	Overall shape quality measure and normal side smoothing	76
4.2.3	Dividing node smoothing	78
4.2.3.1	Face shape quality measurement for 3D solid elements	78
4.2.3.2	Face normal vector quality measurement	79
4.2.3.3	Overall shape quality measure and dividing node smoothing	80
4.3	Solid Mesh Generation Examples	81
4.4	Closure	88
CHAPTER 5 STRESS RECOVERY AND ERROR ESTIMATION		90
5.1	Selection of Elements	90
5.2	Basic Notation	95
5.2.1	Nodal normal vector	95
5.2.2	Coordinate Systems	97
5.2.2.1	Nodal coordinate system (NCS)	99
5.2.2.2	Convective coordinate system (CCS)	100
5.2.3	Energy norm and error norm	102
5.2.4	Separation of stress and energy norm	104
5.3	Stress Recovery for TWSs	107
5.3.1	A brief review on the SPR technique	108
5.3.2	Modified SPR technique for single surface structures	110
5.3.2.1	Selection of appropriate polynomial terms	110
5.3.2.2	Local patch coordinate system	111
5.3.2.3	Scaling of patch coordinates	115
5.3.2.4	Boundary patch enhancement	117
5.3.3	Stress recovery for structures with junction	120
5.4	A Prior Error Estimation for TWSs	122
5.5	A Posterior Estimation for TWSs	123
5.5.1	A brief review on Z-Z error estimator	124

5.5.2	Error estimation for TWSs	125
5.5.2.1	Conjoint polynomial interpolation	126
5.5.2.2	Estimated error for TWSs	127
5.6	Error Estimation Examples	129
5.7	Closure	139
CHAPTER 6 ADAPTIVE REFINEMENT		140
6.1	Overview of the General Adaptive Refinement Scheme	140
6.2	Adaptive Refinement Schemes for TWS	142
6.2.1	Notations used	142
6.2.2	Target error distribution	145
6.2.3	Selection of the adaptive scheme	147
6.2.3.1	The immediate target error	148
6.2.3.2	Selection of refinement directions	150
6.2.4	Algorithm to define the new element size	151
6.2.4.1	Element sizes determination	152
6.2.4.2	Element sizes adjustment	154
6.2.5	Mesh refinement	156
6.2.5.1	Formation of nodal element sizes	156
6.2.5.2	Adjustment of nodal element size in thickness direction	158
6.2.5.3	Generation of the new mesh	159
6.3	Retrieve of the Stress from the Reference Mesh	160
6.4	Adaptive Refinement Examples	162
6.4.1	Examples of structures formed by single surfaces	162
6.4.2	Examples of structures formed by multi-surface	188
6.5	Closure	215
CHAPTER 7 CONCLUSION AND FUTURE WORK		217
7.1	Conclusions for Current Work	217
7.1.1	Conclusion of the auto mesh generation step	217
7.1.2	Conclusion of the error estimation step	219

7.1.3	Conclusion of the adaptive refinement scheme design step	220
7.2	Plans for Further Research Work	221
	REFERENCE	227
	APPENDIX A FORMATION OF THE HIGH-PERFORMANCE ELEMENT	240
A.1	Weaker continuity condition	240
A.2	Refined nonconforming displacement function	241
	APPENDIX B SEARCH FOR THE ELEMENT ENCLOSING A GIVEN POINT	247

SUMMARY

Analysis of thin-walled structures is often encountered in many fields of engineering applications. Due to the zero normal stress assumption, traditional shell theories are not suitable for the analysis of junction between two thin-walled structures or shells. So using refined 3-dimensional (3D) elements to analyze the thin-walled structures is needed.

In this report, a novel algorithm to generate 3D mesh for thin-walled shell structures is proposed. In the proposed algorithm, the mesh generation procedure is divided into two distinct steps. In the first step, a surface mesh will be generated to form the middle surface of the thin-walled shell structure. In the second step, a specially designed algorithm is used to convert the surface mesh to a 3D solid mesh by extruding the elements along the normal direction of the surface. Any existing surface mesh generator can be used to generate the surface mesh and either a triangular or a quadrilateral mesh can be accepted as input for the extrusion. In addition, the extrusion procedure development is able to handle complex joints generated by the intersection of different shell surfaces.

A suitable *a posteriori* error estimator has also been developed to provide convincing error estimation for thin-walled structures either in the surface direction or in the normal direction. In order to separate the stresses into the components along the surface direction and normal direction, local coordinate systems are established at different locations on the shell to get the approximate shell surface direction and normal direction. Finally, an adaptive refinement scheme based on the estimated error in surface direction and normal direction is designed to perform the mesh refinement in two separate steps. The first step is carried out in the surface direction, which is controlled by the surface mesh generator. The second step is implemented in the normal direction, which is controlled by the layers used in

extrusion scheme. Furthermore, the adaptive scheme also can adjust the accuracies in surface direction and normal direction according to the energy contribution in the corresponding directions, which will greatly reduce the total number of degrees of freedom and save the computational time.

Numerical tests have been done to test the effectivity of the suggested adaptive strategies by comparing with the results obtained using uniform mesh refinement.

LIST OF FIGURES

Figure 1.1	Two analysis approaches for thin-walled structures	5
Figure 1.2	Flow chart of the whole analysis procedure	9
Figure 3.1	The process of surface extrusion	28
Figure 3.2	Extrusion along normal direction	34
Figure 3.3	Extrusion of surface elements to volume elements	34
Figure 3.4	The process of normal offsetting	35
Figure 3.5	Normal vector at a node	36
Figure 3.6	Degenerated cases for the fitting procedure: (a) only one distinct normal $\mathbf{n}_1$ exist; (b) two distinct normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ exist	37
Figure 3.7	Three types of structures with surface intersections: (a) a simple surface intersection; (b) a multi-cross intersection; and (c) a corner point intersection	38
Figure 3.8	Overlapping elements, 2D-cross curve and 3D-cross face	39
Figure 3.9	Intersection angles between two intersecting surfaces	39
Figure 3.10	Definitions of φ , θ_I and θ_{II}	40
Figure 3.11	The situation when φ is small	41
Figure 3.12	Surface node layers near the 2D-cross curve	42
Figure 3.13	Location of perpendicular point A on the 2D-cross curve: (a) $\angle \mathbf{BN}_1\mathbf{N}_3 > 90^\circ$ and $\angle \mathbf{BN}_1\mathbf{N}_2 < 90^\circ$; (b) $\angle \mathbf{BN}_1\mathbf{N}_3 < 90^\circ$ and $\angle \mathbf{BN}_1\mathbf{N}_2 < 90^\circ$; (c) $\angle \mathbf{BN}_1\mathbf{N}_3 > 90^\circ$ and $\angle \mathbf{BN}_1\mathbf{N}_2 > 90^\circ$	42
Figure 3.14	Re-definition of normal sides for nodes: (a) plan view; (b) section view	43
Figure 3.15	Upper half of an invalid volume element formed after extrusion	45
Figure 3.16	Adjustment of normal side VB	46
Figure 3.17	Treatment of volume meshes for two intersecting surface meshes	48
Figure 3.18	Division of a 3D-cross face and volume mesh to two separate parts	49

Figure 3.19	Multi-cross and the normal directions	49
Figure 3.20	Calculation the normal side through the corner point	51
Figure 3.21	Determine the final normal side for node A near a corner point	52
Figure 3.22	Meshes for Example 3.1	53
Figure 3.23	Meshes for Example 3.2	54
Figure 3.24	Meshes for Example 3.3	55
Figure 3.25	Meshes for Example 3.4	55
Figure 3.26	Meshes for Example 3.5	56
Figure 3.27	Meshes for Example 3.6	57
Figure 4.1	Layering the volume mesh	59
Figure 4.2	Division of a volume element, $NEL = 3$	60
Figure 4.3	Rectification of input NEL values: (a) input NEL values; (b) modification starts at the normal sides with highest NEL values; (c) after first layer is modified; (d) after two layers are modified; (e) after three layers are modified; and (f) final NEL values	62
Figure 4.4	Connection of adjacent normal sides: (a) case of $NL_A = NL_B = 3$; (b) case of $NL_A = 5 = NL_B + 2$; (c) case of $NL_A = 5 = NL_B + 1$; (d) case of $NL_A = 4 = NL_B + 1$; and (e) case of $NL_A = 6 = NL_B + 2$	63
Figure 4.5	Formation of 3D solid elements from an assembly of prism volume elements: (a) an assembly of prism volume elements and NEL at normal sides; (b) connections of normal sides; and (c) formation of 3D solid elements	65
Figure 4.6	A hexahedral volume element E such that $NEL_{max}=3$, $NELD=3$ and $NMS=1$	66
Figure 4.7	Splitting of a hexahedral volume element into two prism volume elements	67
Figure 4.8	Examples (set 1) for refinement of hexahedral volume element when $NELD=2$: (a) $NMS=1(2-1-1-1)$; (b) $NMS=2(2-2-1-1)$; and (c) $NMS=3(2-2-1-2)$ (Numbers inside brackets are values of $NELA-NELB-NELC-NELD$ respectively)	68
Figure 4.9	Examples (set 2) for refinement of hexahedral volume element when $NELD=2$: (a) $NMS=1(3-1-1-1)$; (b) $NMS=2(3-3-1-1)$; and (c) $NMS=3(3-3-3-1)$ (Numbers inside brackets are values of $NELA-NELB-NELC-NELD$ respectively)	69
Figure 4.10	Examples (set 3) for refinement of hexahedral volume element when $NELD=2$: (a) $NMS=1(3-3-3-2)$; (b) $NMS=2(3-3-2-2)$;	

	and (c) NMS=3 (3-3-2-3) (Numbers inside brackets are values of NELA-NELB-NELC-NELD respectively)	70
Figure 4.11	Examples (set 3) for refinement of hexahedral volume element when NELD=3: (a) NMS=1 (3-2-1-1); (b) NMS=1 (3-2-2-1); and (c) NMS=2 (3-3-2-1) (Numbers inside brackets are values of NELA-NELB-NELC-NELD respectively)	71
Figure 4.12	Measures of triangle and quadrilateral	74
Figure 4.13	Normal side smoothing	77
Figure 4.14	Face shape quality measure for 3D solid elements: (a) tetrahedron; (b) pyramid; (c) prism; and (d) hexahedral	78
Figure 4.15	Face normal vector quality measure for interior node Q: (a) an assembly of 3D solid elements; (b) elements connectivity and (c) faces associated with interior node Q	80
Figure 4.16	Meshes for Example 4.1	81
Figure 4.17	Meshes for Example 4.2	82
Figure 4.18	Meshes for Example 4.3	83
Figure 4.19	Meshes for Example 4.4	84
Figure 4.20	Meshes for Example 4.5	85
Figure 4.21	Meshes for Example 4.6	86
Figure 4.22	Meshes for Example 4.7	87
Figure 4.23	Meshes for Example 4.8	88
Figure 5.1	Quadric elements used in the FE analysis	91
Figure 5.2	Cantilever plate of 10-node tetrahedron elements	92
Figure 5.3	Cantilever plate of 13-node pyramid elements	92
Figure 5.4	Cantilever plate of 15-node prism elements	93
Figure 5.5	Cantilever plate of 20-node hexahedron elements	93
Figure 5.6	Boundary condition for the cantilever plate example: (a) boundary condition for Figure 5.2; (b) boundary condition for Figure 5.3, Figure 5.4 and Figure 5.5	93
Figure 5.7	Normal of dividing node	96
Figure 5.8	Normal vector of a point on boundary surface	96
Figure 5.9	Coordinate systems for solid elements of the TWS	98
Figure 5.10	Convective coordinate system	100
Figure 5.11	Local stress components	104

Figure 5.12	Superconvergent points within the 8-node quadrilateral elements	108
Figure 5.13	Element patch of node A	108
Figure 5.14	A simple 2D patch for the comparison of SPR1 and SPR2	112
Figure 5.15	SPR1 method	113
Figure 5.16	SPR2 method	113
Figure 5.17	Comparison of SPR1 and SPR2	115
Figure 5.18	Maximum boundary distances of the patch	116
Figure 5.19	Patch of a boundary element which does not consist of interior node: (a) boundary triangular element in a 2D mesh (b) boundary tetrahedron element in a 3D mesh	118
Figure 5.20	Local coordinate system used for SPR-R technique: (a) 2D mesh; (b) 3D mesh for TWS	120
Figure 5.21	Junction zone and shell zone of a TWS with surface intersections	121
Figure 5.22	Junction patch and shell patch in a structure shown in Figure 5.20	122
Figure 5.23	Conjoint polynomial interpolation at point A: (a) patch for node 1; (b) patch for node 2 and (c) patch for node 3	126
Figure 5.24	Example 5.1: plate with central circular hole	129
Figure 5.25	Example 5.1: meshes for triangular input elements	131
Figure 5.26	Example 5.1: meshes for quadrilateral input elements	131
Figure 5.27	Example 5.1: convergence rate for triangular input elements	132
Figure 5.28	Example 5.1: convergence rate for quadrilateral input elements	133
Figure 5.29	Example 5.2: spherical container subjected to uniform internal pressure	133
Figure 5.30	Example 5.2: boundary conditions for one-eighth of the shell	134
Figure 5.31	Example 5.2: meshes of refinement for case $t = 2$	135
Figure 5.32	Example 5.2: meshes of refinement for case $t = 1$	135
Figure 5.33	Example 5.2: meshes of refinement for case $t = 0.5$	135
Figure 5.34	Example 5.2: comparison of the convergence rate	138
Figure 6.1	Main steps in the adaptive refinement scheme	141
Figure 6.2	Determine the nodal element size in surface direction	157
Figure 6.3	The nearest belonging patch of point P	161

Figure 6.4	Example 6.1: adaptive refinement	163
Figure 6.5	Example 6.1: uniform refinement	164
Figure 6.6	Example 6.1: comparison of convergence rate	165
Figure 6.7	Example 6.2: region near a crack tip	166
Figure 6.8	Example 6.2: adaptive refinement	167
Figure 6.9	Example 6.2: uniform refinement	168
Figure 6.10	Example 6.2: comparison of convergence rate	169
Figure 6.11	Example 6.3: a square plate under uniform pressure with soft-simple on all sides: (a) whole structure; (b) a quadrant of the structure and the corresponding boundary conditions	170
Figure 6.12	Example 6.3, Case 1: 2D view of the reference mesh	171
Figure 6.13	Example 6.3, Case 1: adaptive refinement	171
Figure 6.14	Example 6.3, Case 1: uniform refinement	172
Figure 6.15	Example 6.3, Case 1: comparison of convergence rate	173
Figure 6.16	Example 6.3, Case 2: 2D view of the reference mesh	174
Figure 6.17	Example 6.3, Case 2: adaptive refinement	175
Figure 6.18	Example 6.3, Case 2: uniform refinement	175
Figure 6.19	Example 6.3, Case 2: comparison of convergence rate	177
Figure 6.20	Example 6.4: L-shaped plate	177
Figure 6.21	Example 6.4: 2D view of the reference mesh	178
Figure 6.22	Example 6.4: adaptive refinement	179
Figure 6.23	Example 6.4: uniform refinement	180
Figure 6.24	Example 6.4: comparison of convergence rate	182
Figure 6.25	Example 6.5: cylindrical shell roof under self-weight	183
Figure 6.26	Example 6.5: boundary conditions.	183
Figure 6.27	Example 6.5: reference mesh	184
Figure 6.28	Example 6.5: adaptive refinement	185
Figure 6.29	Example 6.5: uniform refinement	186
Figure 6.30	Example 6.5: comparison of convergence rate	187
Figure 6.31	Example 6.6: two intersected plates under horizontal pressure	189
Figure 6.32	Example 6.6: boundary conditions	189
Figure 6.33	Example 6.6: reference mesh	190
Figure 6.34	Example 6.6: adaptive refinement	191

Figure 6.35	Example 6.6: uniform refinement	193
Figure 6.36	Example 6.6: comparison of convergence rate	195
Figure 6.37	Example 6.7: two intersected plates under horizontal pressure	196
Figure 6.38	Example 6.7: boundary conditions	197
Figure 6.39	Example 6.7: reference mesh	197
Figure 6.40	Example 6.7: adaptive refinement	199
Figure 6.41	Example 6.7: uniform refinement	200
Figure 6.42	Example 6.7: comparison of convergence rate	203
Figure 6.43	Example 6.8: two intersected plates under horizontal pressure	204
Figure 6.44	Example 6.8: boundary conditions of a quarter of the structure	204
Figure 6.45	Example 6.8: reference mesh	205
Figure 6.46	Example 6.8: adaptive refinement	206
Figure 6.47	Example 6.8: uniform refinement	206
Figure 6.48	Example 6.8: comparison of convergence rate	209
Figure 6.49	Example 6.9: two intersected plates under horizontal pressure	210
Figure 6.50	Example 6.9: boundary conditions of a quarter of the structure	210
Figure 6.51	Example 6.9: reference mesh	211
Figure 6.52	Example 6.9: adaptive refinement	212
Figure 6.53	Example 6.9: uniform refinement	212
Figure 6.54	Example 6.9: comparison of convergence rate	215
Figure 7.1	Example of TWS with complex surface intersection	222
Figure 7.2	Adjust the algorithm on NEL redefinition	223
Figure B.1	Volume of the tetrahedron	249
Figure B.2	Split of a hexahedron element	249
Figure B.3	Spilt of a prism element	249
Figure B.4	Split of a pyramid element	250
Figure B.5	Split the element that has quadrilateral facet	250

LIST OF TABLES

Table 2.1	Some techniques of mesh cleanup for triangular elements	21
Table 3.1	Normal vectors of surfaces for different 2D-cross curve	50
Table 4.1	Number of different types of solid elements in Figure 4.5b	65
Table 4.2	Number of different types of solid elements in Figure 4.8	68
Table 4.3	Number of different types of solid elements in Figure 4.9	69
Table 4.4	Number of different types of solid elements in Figure 4.10	70
Table 4.5	Number of different types of solid elements in Figure 4.11	71
Table 5.1	Element types in the final solid mesh	91
Table 5.2	Coordinates of the corner nodes of P13 element	92
Table 5.3	load values for different plate thickness	94
Table 5.4	Deflections for a cantilever plate	94
Table 5.5	Number of integration points used for the quadratic solid elements	125
Table 5.6	Equations in a posteriori error estimation for junction zone and shell zone	129
Table 5.7	Example 5.1: Results of triangular input elements	131
Table 5.8	Example 5.1: Results of quadrilateral input elements	131
Table 5.9	Example 5.2: results for case $t = 2$	136
Table 5.10	Example 5.2: results for case $t = 1$	136
Table 5.11	Example 5.2: results for case $t = 0.5$	137
Table 5.12	Example 5.2: comparison of different cases of thickness	137
Table 6.1	Notations of superscripts and subscripts	143
Table 6.2	Example 6.1: results of the adaptive refinement scheme	164
Table 6.3	Example 6.1: results of the uniform refinement scheme	164

Table 6.4	Example 6.2: results of the adaptive refinement scheme	168
Table 6.5	Example 6.2: results of the uniform refinement scheme	169
Table 6.6	Example 6.3, Case 1: details of the reference mesh	171
Table 6.7	Example 6.3, Case 1: results of the adaptive refinement scheme	172
Table 6.8	Example 6.3, Case 1: results of the uniform refinement scheme	172
Table 6.9	Example 6.3, Case 2: details of the reference mesh	174
Table 6.10	Example 6.3, Case 2: results of the adaptive refinement scheme	176
Table 6.11	Example 6.3, Case 2: results of the uniform refinement scheme	176
Table 6.12	Example 6.4: details of the reference mesh	178
Table 6.13	Example 6.4: results of the adaptive refinement scheme	181
Table 6.14	Example 6.4: results of the uniform refinement scheme	181
Table 6.15	Example 6.5: details of the reference mesh	184
Table 6.16	Example 6.5: results of the adaptive refinement scheme	186
Table 6.17	Example 6.5: results of the adaptive refinement scheme	186
Table 6.18	Example 6.6: details of the reference mesh	190
Table 6.19	Example 6.6: results of the adaptive refinement scheme	193
Table 6.20	Example 6.6: results of the uniform refinement scheme	194
Table 6.21	Example 6.6: comparison of computational time (time unit: second)	194
Table 6.22	Example 6.7: details of the reference mesh	198
Table 6.23	Example 6.7: results of the adaptive refinement scheme	201
Table 6.24	Example 6.7: Results of the uniform refinement scheme	201
Table 6.25	Example 6.7: comparison of computational time (time unit: second)	201
Table 6.26	Example 6.8: details of the reference mesh	205
Table 6.27	Example 6.8: results of the adaptive refinement scheme	207
Table 6.28	Example 6.8: results of the uniform refinement scheme	207
Table 6.29	Example 6.8: comparison of computational time (time unit: second)	207
Table 6.30	Example 6.9: details of the reference mesh	211
Table 6.31	Example 6.9: results of the adaptive refinement scheme	213
Table 6.32	Example 6.9: results of the uniform refinement scheme	213

Table 6.33	Example 6.9: comparison of computational time (time unit: second)	213
Table A.1	Functions of M1 for different elements	246

LIST OF SYMBOLS

$\{ \}$	Vector
$[\]$	Matrix
$\{ \}^T, [\]^T$	Transposed vector or matrix
$\ \ $	Norm
Ω	Domain
$\int_{\Omega}$	Domain integration
$\sum$	Summation
$\mathbf{M}$	Metric tensor
Φ	Measure of facet shape quality for normal side smoothing
Ψ	Measure of normal direction quality for normal side smoothing
Θ	Measure of the overall shape quality for normal side smoothing
$\tilde{\Phi}$	Measure of face shape quality for dividing node smoothing
$\tilde{\Psi}$	Measure of face normal vector quality for dividing node smoothing
$\tilde{\Theta}$	Measure of the overall shape quality for dividing node smoothing
E	Young's module
ν	Poisson ratio
$\mathbf{u}$	Displacement vector
$\boldsymbol{\sigma}$	Stress vector
σ	Normal stress component
τ	Shear stress component
$\boldsymbol{\varepsilon}$	Strain vector
ε	Normal strain component
γ	Shear strain component
$\mathbf{D}$	Strain-stress matrix
$\mathbf{e}$	Pointwise error in stress

η	Relative error
λ	Strength of singularity
θ	Effectivity index
R	Convergence rate
t	Thickness of structure
h	Element size
x	x coordinate
y	y coordinate
z	z coordinate

CHAPTER 1

INTRODUCTION

1.1 Background

With the fast development of computational hardware, the *finite element method* (FEM) is becoming more and more widely used by the engineers for the analysis of many varieties of problems. The basic idea of this method is to break down a complicated problem into a sequence of simpler discretized problems and then integrate their solutions. As this kind of simplification is an approximation of the original problem, the solution obtained will not be exact and discretization error exists. A too coarse discretized model will lead to inaccurate or even erratic solutions which may result in wrong judgments. Hence, it is necessary to make an evaluation on how good the model is and how accurate and reliable the obtained FEM solutions are. Such a requirement has led to the studies and researches on the adaptivity of the finite element method. For a general *adaptive finite element procedure*, the solution is first started from a relatively coarse model. According to the analysis of the FEM solution of the current model, an error estimator will estimate the error of the current solutions and provide a guide for the creation of a better discretized model. The current model is then refined by increasing the number of *degrees of freedom* (DOF) where the previous model is inadequate. The estimation and refinement procedures will be repeated until a solution with the desired accuracy is obtained. A well designed adaptive refinement scheme will help engineers to obtain an accurate and reliable FEM solution of the modelled problem at less time cost. Furthermore, it will also assist the engineers to get a clear picture

Introduction

about the distribution of error and the variation of the solution of the problem.

Analysis of *thin-walled structures* (TWS) is frequently encountered in many fields of engineering applications. In most engineering disciplines, a structure is considered as a TWS if its first two characteristic dimensions, (length/width) are much larger (e.g. one order of magnitude higher) than its third characteristic dimension (thickness). However, it should be noted that, among different disciplines, different values of the length to thickness ratio could be used. For example, in structural and offshore engineering, the value of such ratio could be as small as 10 while in the application of plastic injection molding modelling, the ratio adopted is usually bigger (e.g. 50). Traditionally, in structural engineering, plate and shell elements are used for analyzing of TWSs and the *finite element* (FE) meshes needed are often generated by an automatic surface mesh generator. Even through many efficient shells elements are available, due to the zero normal stress assumption, they are unable to predict the normal stress variation along the intersection of two shell structures. Thus, the use of shell elements will not be adequate during the fatigue analyses of thin-walled structures joints. Examples of such type of structures included all kinds of tubular structures which are commonly used in the construction offshore platforms [1, 2]. Even in principle, accurate *stress concentration factor* (SCF) predictions and fatigue analyses can be obtained by using 3D FE method, the use of full 3D FE models for the study of TWSs is so far uncommon. The two main difficulties faced by many engineers who tried to use 3D FE models for the study of TWS joints are

- (i) Most of the conventional 3D mesh generation schemes available to date are primary not designed for the discretization of TWS in which the aspect ratio of the problem domain in the thickness direction is at least one order less than that of the other dimensions.
- (ii) Even a conventional 3D mesh generator is able to discretize the TWS into a

solid 3D mesh, since the aspect ratio of elements are similar in all directions, it will result in impractical number of degrees of freedoms.

Duo to the difficulties mentioned above, the analysis of TWSs using 3D finite element method appears to be a relatively new and unexplored topic.

1.2 Objective

The main objective of this study is to suggest a new adaptive refinement procedure for TWSs using 3D solid elements. In particular, the current algorithm is mainly designed for TWSs encountered in structural and offshore engineering where the length to thickness ratio of the structure is within the range of [10, 100]. As mentioned in the previous section, this is a relatively untouched topic so that the study will cover all the procedures of an adaptive refinement cycle. A failure or mistake coming from any procedure may result in a bad performance of the final scheme. In order to develop a robust and reliable adaptive refinement scheme, the following requirements are essential for the researcher:

- (i) A clear awareness of the mechanical properties of the problem that need to be solved. This requirement will assist the researcher to make correct hypotheses which will ensure that the focus can be devoted to the most critical part of the problem in the real application.
- (ii) A thorough understanding of the mathematical model of the problem. This requirement will help the researcher to find the correct model to convert the complex physical engineering problem to simplified mathematical problem.
- (iii) A comprehensive command of the whole *finite element analysis* (FEA) procedure. This requirement will make sure that every procedure of the finite element analysis is correctly developed and they are also successfully integrated.

Introduction

- (iv) A good mastery of programming techniques. This requirement will guarantee that all the algorithms are implemented accurately and efficiently. Besides that, a well designed program can reduce the numerical error as much as possible.

As mentioned before, in order to design a successful adaptive refinement procedure for thin-walled structures, the mechanical properties of this kind of structures must be studied first. According to the basic theory of structural mechanics, stresses of thin-walled structures are dominant in the shell surface direction and the energy contribution in shell normal direction is trivial. This phenomenon is also the reason why shell elements are usually used for the analysis of shell structures. Based on this property, the main idea of this research is to separate the analysis of thin-walled structures into two directions: one is the shell surface direction and the other is the shell normal direction. The reasons why such a treatment can be taken are:

- (i) Stresses are mainly distributed along the surface (in-plane) direction, the energy contribution in shell normal direction is small.
- (ii) Only near the joint between two shell surfaces, there is no obvious surface direction and normal direction. For most parts of the structure, the surface direction and normal direction can be clearly identified by the definition of the geometry.
- (iii) General 3D mesh generators are not suitable for thin-walled structures, while many robust surface mesh generators are available.

Based on the above arguments, the whole mesh generation procedure will be broken down into two steps. In the first step, the thin-walled structure is reduced to its middle surface form and a surface mesh generator will be employed to discretize the middle surface into surface mesh. In the second step, a specially designed algorithm

Introduction

will convert the surface mesh to a solid mesh for the modeling of the thin-walled structures. During the error estimation procedure, the energy norm and error norm in surface direction and normal direction will be calculated separately. Hence the mesh refinement work can be carried out in surface direction and normal direction respectively.

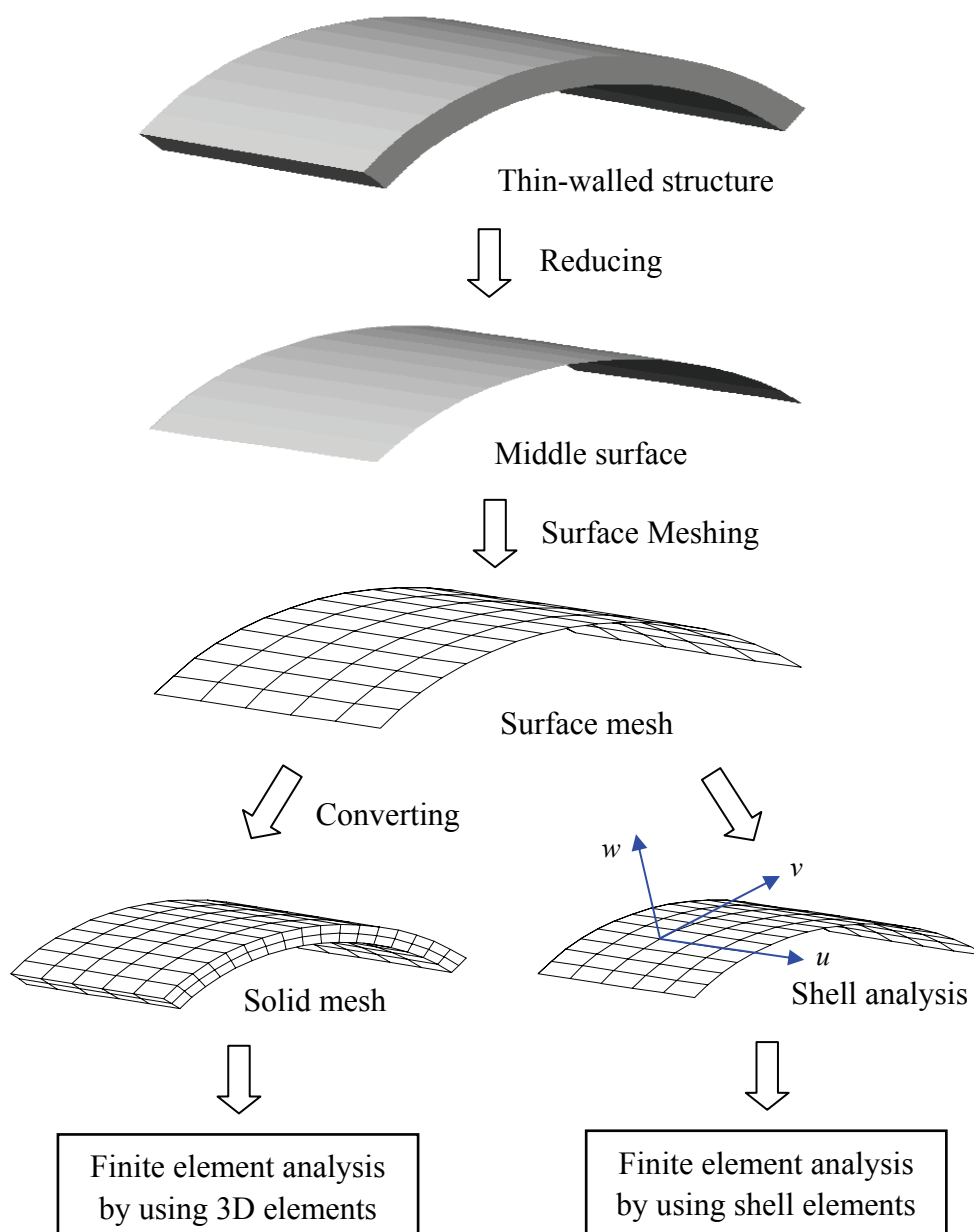


Figure 1.1 Two analysis approaches for thin-walled structures

Figure 1.1 illustrates the difference of treatments for thin-walled structures between the method used in this report and the method of shell analysis. For the shell analysis approach, after the surface mesh is formed, shell theory (and hence shell elements) will be used to analyze the problem; for the present analysis approach, the surface mesh will be further converted to solid 3D mesh first, and then the 3D finite element method will be used to analyze the problem.

1.3 Scope

Adaptive finite element procedure is a general method which can be applied to a lot of structures and many mechanical problems. In the current research, only the linear elastic problems of the thin-walled structures are studied. This is because linear elastic problems are the most commonly used formulations for structural mechanics, and also the most widely used formulations by many structural and civil engineers. A successful adaptive refinement scheme for this class of problem will provide a valuable basis to tackle the extensions to more complicated formulations in future research.

In the current study, attentions will be focused on the following critical components for the analysis procedure:

1) *Automatic mesh generation*

In this part, the middle surface of the thin-walled structure will be meshed by an existing surface mesh generator. In this study, the mesh generator developed by Lee [3] will be adopted to generate the triangular surface mesh. If quadrilateral mesh is needed, a particular algorithm developed by Lee [4] can be used to transfer the triangular mesh to quadrilateral mesh. After the surface mesh is obtained, a new program will be developed to convert the surface mesh to solid mesh specially.

2) *Selection of the element types*

It should be noticed that after the mesh generation part is finished, the final solid mesh will consist of four different element types (brick, prism, tetrahedron and pyramid). All these elements are linear solid elements. Most of them are rather thin, some may even be distorted. A thin or distorted linear solid element may not perform well in the finite element analysis according to the experience of FEM analysts. To get a better FEA performance, quadratic elements will be used here so that the linear solid mesh will be further converted to quadratic solid mesh [5]. Although some high-performance elements [6-8] have been developed, due to the difficulty of the implementation of the algorithms, they are not used in the current study, so only conventional quadratic elements are used here.

3) *Finite element analysis*

For the core finite element analysis part, since there are many commercial FEA software package available, there is no need to develop a separate program to repeat all the matured algorithms such as forming the stiffness matrices and solving linear equations. This part of work can be assigned to existing FEA software. In the current study, ABAQUS, the commercial FEA software package, is used to carry out the standard finite element analysis process. The advantages of this decision are: 1) a matured FEA software package can provide more robust and more reliable FEM solution than the user developed program; 2) A commercial software package is well optimized on the balance between the memory cost and computational speed; 3) Using existing software can help the author put more concentration on the development of the new algorithms; 4) Pyramid element will exist in the solid mesh obtained by the mesh generation procedure, while most of the commercial FEA software do not have this kind of element in its element library. ABAQUS has provided the powerful user-defined subroutine (UEL), and it can allow users to define their own elements. So the pyramid element can be easy developed.

4) *Stress recovery and error estimation*

In this part, *a priori* error estimation will be analyzed to assess the convergence rate of the FEM solution. Based on the famous Z-Z error estimator [9, 10], a new *a posteriori* error estimator will be specially designed for thin-walled structures. In order to take the error in surface direction and normal direction into consideration separately, local coordinate systems will be defined at different positions within the mesh. For the structures with joints, there is no obvious surface direction or normal direction at the junction part, so this zone will be treated specially. Considering that smaller surface elements are needed in the region near the junction because of stress concentration, the solid element size in surface direction does not differ much to the size along the normal direction. Therefore, in the region near the junction, the ratio of the element size between different directions will satisfy the element quality requirement for a standard 3D finite element analysis. Hence, in the current research, the joint part is regarded as a general 3D zone and the error will be only measured in the total sense.

5) *Mesh refinement*

In this part, a separate module will be designed to use the information given by the error estimator to predict the new element size of the next refined mesh. According to the different accuracy requirements of the users (some may only care about the error in total sense, some may concern more on the error of particular components such as the surface direction error, normal direction or the error of the joint part), the program must be able to select different schemes to control and arrange the distribution of the element size automatically.

Figure 1.2 is the flow chart for the procedures in the whole analysis. The adaptive scheme will combine the steps of mesh generation, FEM analysis, stress recovery and error estimation into a continuous procedure. It will keep refining the mesh until the final error is within the given range.

Introduction

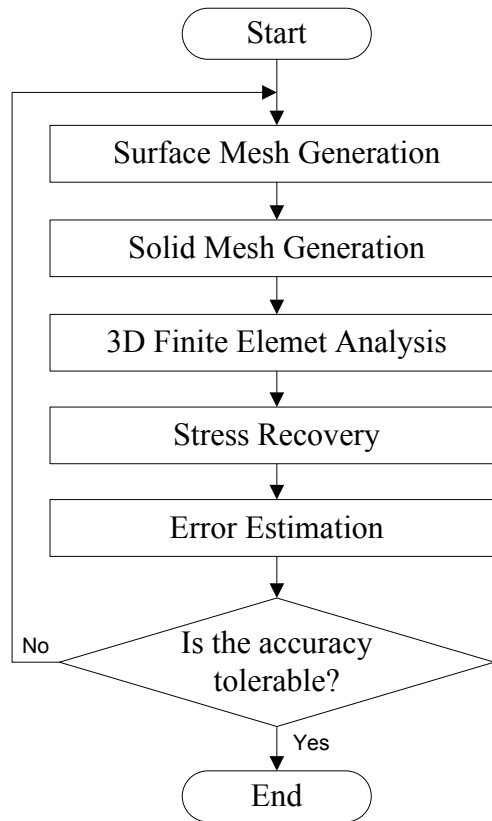


Figure 1.2 Flow chart of the whole analysis procedure

1.4 Computational environment

The computational hardware used in the current study for the development and running most of the examples is a common PC compatible machine equipped with an Intel XEON 2400GHz processor and 2048MB onboard memory and with 160GB hard-drive storage space. Due to the upgrade of the hardware during the later part of the PhD study, some examples are run on a higher end computer. The configuration of this computer is an AMD ATHLON 64 3800+ X2 processor, 4096MB on board memory and 160GB hard-drive storage. It should be mentioned here that a computer with such a hardware configuration is not a high end machine for engineering computation. By the time when this thesis was completed, the AMD ATHLON 64 6000+ X2 processor had been released. There is no doubt that with the aid of latest

Introduction

processor models, the computational speed will be greatly improved. This also means that all the numerical results obtained in the current study are representative in the sense of CPU time.

Programming languages used for the development of the algorithms are FORTRAN 77 and C++. For the surface mesh generator, it is modified based on the program developed by Lee [11], which is written in FORTRAN 77. For all the other codes, they are written in C++. The reason for the selection of this language is that an *Object Oriented Programming* (OOP) language can provide a better organization for the program structure and C++ is proved to be a matured and efficient OOP language. Another reason is that, the author is more familiar with C++, so both FORTRAN 77 and C++ are used in the current work to write the codes. The software environment for the program development is Compaq Visual Fortran 6.6 and Microsoft Visual Studio 6 with SP6. The operating system is Microsoft Windows XP Professional with SP2 (32-bit version).

The finite element analysis process is carried out by ABAQUS 6.5. This version of ABAQUS can only access 3GB memory when the operating system is a 32-bit system. Due to such a restriction of ABAQUS, it has been found that the upper limitation of the DOFs used for the problem is about 1 million.

1.5 Organization

After this introduction, an overview of automatic surface mesh generation, solid mesh generation, error estimation and adaptive mesh refinement will be provided in Chapter 2. It will only give a brief investigation for the related background; some theories and algorithms that are closely related to this study will be explained in details in the following chapters.

Introduction

In Chapter 3, methods used in the adopted surface mesh generator will be first introduced. After that, a special designed surface extrusion algorithm will be presented. Emphases are given to the treatments for structures with multi surfaces.

Chapter 4 will be devoted to algorithms designed for dividing the volume mesh, which is generated by the surface extrusion procedure, into layered solid mesh. A volume mesh refinement scheme will be presented first. After that, some solid mesh enhancement techniques will be discussed.

In Chapter 5, *a priori* error estimator in form of convergence rate prediction will be discussed first. Then, the superconvergent patch recovery algorithm will be presented in details. Based on *Z-Z* method, a new *a posteriori* error estimator which is specially designed for the proposed mesh generator will be developed in the last.

An appropriate adaptive refinement scheme will be brought forth in Chapter 6. Focus will be given on how to select the schemes to control the accuracy in different parts or in different directions according to the different requirements asked by the user. Numerical examples are provided to show the efficiency of the developed adaptive refinement scheme.

In Chapter 7, the last chapter of the thesis, conclusions of the current work are summarized. In addition, plans for future work will also be presented.

CHAPTER 2

LITERATURE REVIEW

In Chapter 1, a brief introduction of the research work is given. In this chapter, some related theories will be reviewed, which are listed as:

- 1) Finite element method
- 2) Mesh generation
- 3) Error estimation
- 4) Adaptive mesh refinement

2.1 A Brief Review of the Finite Element Method

2.1.1 History of FEM

The finite element method is an analysis tool for numerically analyzing problems described by partial differential equations or can be formulated as functional minimization. It was first proposed by Courant [12] as a method of stress analysis. He defined piecewise linear polynomials over a triangularized region to solve the torsion problem in elasticity. Shortly thereafter, a paper published in 1956 by Turner et al. [13] established a broader definition of numerical analysis. This paper centered on the "stiffness and deflection of complex structures". The early research work on FEM was limited in the aeronautics, automotive, defense, and nuclear fields due to the huge expense of computation. The name "finite element method" was coined by Clough [14] in 1960. In 1965 Zienkiewicz and Cheung [15] reported that this method is applicable to all field problems that can be cast into

variational form. After that, the finite element method aroused the broad attention of researches and became a hot research topic. With the development of computer technology, the finite element method has become the most popular method for engineering analysis. It has been successfully applied to a wide variety of fields such as solid mechanics, heat transfer, fluid mechanics, acoustics, electro-magnetism and quantum mechanics. Those problems can be boundary-value, eigen-value or initial-value problems, described by differential, integral or variational equations.

2.1.2 Basis of FEM

For most practical problems, it is almost impossible to find the exact solution for the unknown to satisfy the governing equations and boundary conditions. The aim of the FEM is to find an explicit expression for the unknown, in terms of known functions, which approximately satisfies all the governing equations and initial conditions. In order to simplify the problem, the original continuous domain is divided into finite discrete sub-domains (elements) and then the analysis is carried out within each domain individually. As the element is formed by simple geometry, it is much easier to analyze than the actual structure. It needs to be mentioned that there are two valuable features of the finite element method: i) Piece-wise approximation of physical fields on finite elements is able to provide good precision even with simple approximate functions; ii) Locality of approximation leads to sparse equation systems for a discretized problem. This helps to solve problems with very large number of nodal unknowns.

A standard FEM analysis procedure consists of seven major steps:

- (1) *Discretize the continuum.* A solution region is divided into finite elements so that the original domain is represented by a mesh. This step is also referred as

the mesh generation process. The description of mesh consists of the information of nodal coordinates and element connectivity.

- (2) *Select element interpolation functions.* Values inside an element can be recovered using the interpolation of element nodal values. Usually, polynomials are selected as interpolation functions. The degree of the polynomial depends on the number of nodes assigned to the element.
- (3) *Formulate the element properties.* The matrix equation for the finite element should be established which relates the nodal values of the unknown function to other parameters. If the formulation of the physical problem is given as a differential equation, the *Method of Weighted Residuals*, such as the famous *Galerkin Method*, can be used. If the physical problem can be formulated as minimization of a function, then the *Variational Method*, also known as the *Ritz Method*, is usually used.
- (4) *Assemble the element equations.* According to the information of element connectivity, equations of all the discretized elements are assembled to global equation system for the whole solution region.
- (5) *Apply the known loads and boundary conditions.* Using stress analysis problem for example, the loads are the nodal forces or moments. The boundary conditions can be the constraints or the displacements that are already known.
- (6) *Solve the global equation system.* The finite element global equation system is typically sparse, symmetric and positive definite. Either direct or iterative methods can be used for solution. The nodal values of the sought function are produced as a result of the solution.
- (7) *Calculate additional results.* In many cases the nodal values obtained by solving the global equation system are not the results the user really cares, so addition computation must be done for those parameters interested. For instance, in mechanical problems strains and stresses are of interest in addition to displacements, so further calculation needs to be carried out according to the

relationship between stress/strain and displacement.

It must be repeated that the FEM can only produce an approximate solution for the original program. Theoretically, the FEM solution can be improved to achieve any required accuracy by increasing the number of elements, but in the application, this is impractical because the computation time is also quite important to the engineering analysis. Hence, an error estimate for the FEM solution needs to be designed. This topic will be introduced later in Section 2.3.

2.2 A Brief Review of the Automatic Mesh Generation

Mesh generation can be formally defined as the process of breaking up a physical domain into small sub-domains (elements) in order to facilitate the numerical solution. It is usually the first step of finite element method, also referred as the preprocessing of the FEM. With the popularity of computers, automatic mesh generation algorithms have become the hot research topics and many algorithms have been presented. According to the dimensions of the domain to be meshed, mesh generation can be classified to 2D mesh generation, surface mesh generation and 3D mesh generation. For 2D mesh generation and surface mesh generation, triangular elements and quadrilateral elements are the main element types; for 3D mesh generation, tetrahedron elements and hexahedron elements are the main element types. Since automatic generation is a very wide topic in computational geometry and computational mechanics, in this chapter, the scope of review on mesh generation will not cover the algorithms for all difficult approaches but only be concentrated on some theories related to the current work.

2.2.1 Surface mesh generation

Many mesh generation problems such as shell analysis involve the formation of

elements on 3D surfaces. The resulting surface mesh can also be used as input data for a volume mesh generator. Thus, meshing on the 3D surfaces, which is referred as surface mesh generation, has become an important approach of mesh generation. Surface mesh generation algorithms are classified to parametric mapping [16-18] and direct 3D generation [19, 20].

2.2.1.1 Parametric mapping

The basic idea of this algorithm is simplifying the 3D problem to 2D problem by the way of *mapping*. First, every point on the 3D surface will be mapped to a parametric space. Then all the mesh generation procedures are carried out in the parametric space by a 2D mesh generator. Finally, the generated 2D mesh in the parametric space must be re-mapped back to the 3D space to form the 3D surface mesh. The main advantage of this algorithm is that any 2D mesh generator can be used during the step of meshing in parametric space. Since the research on 2D meshing algorithms is quite matured and a lot of robust 2D mesh generators have been developed, the shape qualities of the elements in parametric space can be guaranteed. The drawback of this kind of algorithm is that the well-shaped elements formed in parametric space may not always form well-shaped elements in 3D space once mapped back to the curved surface. In order to solve this problem, two methods are used to control the element size and shape in the parametric space:

- 1) Modify the parametric representation to ensure that there is a reasonable mapping from parametric space back to 3D space. This method requires that the surface derivatives should not vary widely in the domain. To solve this, some algorithms such as the arc-length reparameterizations [21] and scaled parametric coordinates [22] were proposed. Although these techniques are useful in many cases, they are still not adequate enough.

2) Modify the mesh generation algorithm so that stretched or anisotropic elements meshed in parametric space will map back to well-shaped and isotropic elements in 3D space. This method is proved to be effective by using *Non-uniform rational B-spline* surface and *metric specification*. *Non-uniform rational B-spline* (NURBS) [23] is a numerical method to use the interpolation of polynomials to approximately represent complex curves and curved surfaces. Metric specification is first proposed by Borouchaki [24, 25]. It is a technique to use metric tensor, a nodal matrix, to control the element size in the directions defined by the user. Lee [11, 3] combined these two techniques and successfully applied them in the process of surface mesh generation. Details of the definition for NURBS surface and metric tensor will be introduced in Chapter 3.

2.2.1.2 Direct 3D generation

In this approach, elements are formed directly on the 3D surfaces without regarding to the parametric representation of the underlying geometry. Direct 3D surface mesh generators are particularly useful in some cases where a parametric representation is not available or where the surface parameterization is very poor. Lau and Lo [19] applied *advancing front technique* in 3D surface meshing. In this method new nodes and elements are first generated on the tangential planes of the surfaces and then projected back on to the 3D surfaces. The advantage of this approach is that as the positions of the new nodes are not computed by mapping, the method can be used for the discretization of highly folded surfaces. However, as an iterative projection procedure is used to ensure that new nodes remain on the surface, a significant number of surface projections are required. In addition, special intersection calculations are required to avoid elements overlapping. So this algorithm is more complicated and computationally intensive. Another direct 3D surface meshing approach is the paving algorithm presented by Cass et al. [26]. This

algorithm is suitable for quadrilateral surface meshing.

2.2.2 3D mesh generation

As an important approach of mesh generation, 3D meshing is much more complicated and time consuming than 2D meshing. A lot of work has been done to develop the mesh generation algorithms in 3D cases. As the main techniques of 2D meshing, *Delaunay Triangulation Algorithm* and *Advancing Front Technique* have both been extended to 3D situation. The main characteristics of these two algorithms are listed as following:

1) *Delaunay Triangulation Algorithm*

This algorithm is derived from a mathematic rule referred as the *Delaunay criterion* [27]. The criterion defines a best connection way for a set of given points in the researching 2D domain. It was first applied into the field of automatic mesh generation by Lawson [28] in 1977. Later, Watson [29] extended it into 3D domain. For 2D (surface) domains, the Delaunay criterion requires that any node must not be contained within the *circumcircle* of any triangle within the mesh; while for 3D (volume) domains, it is required that any node must not be contained within the *circumsphere* of any tetrahedron within the mesh. Because Delaunay Triangulation Algorithm is created based on the strict mathematic definition, it is accepted as a robust method to generate the unstructured mesh and has been widely used in the application. This algorithm is very suitable for meshing a convex object, but for a concave object, it becomes troublesome to mesh it directly. Therefore, for a concave domain, most of the research in this approach has been given to the study on how to maintain the boundary geometry [30-32].

2) *Advancing Front Technique*

This algorithm is a relatively new approach in the field of automatic mesh

generation. It was first developed for 2D mesh generation by Lo [33] in 1985. Lohner and Parikh [34] extended this algorithm to 3D cases later. In this technique, the elements are built progressively inward from the boundary surface. An active front is maintained where new elements are formed. For each triangular facet on the front, an ideal location for a new fourth node is computed. As the algorithm progresses, the front will advance to fill the remainder of the area (volume in 3D cases) with formed elements until the whole domain is full of meshes. This algorithm is flexible because it has no requirement for the shape of the boundary so that it is suitable for meshing objects with complex geometries. Another advantage is that the qualities of the elements formed are good. The drawback of the algorithm is that it needs to create an assistant background mesh to help itself to search for the location of the new node that will be inserted. Hence, it increases the computational cost. Recently, some researchers developed combined algorithms by mixing Delaunay Triangulation Algorithm and Advancing Front Technique together [35-37]. Such algorithms combine the advantages of efficiency and nice mathematical properties of a Delaunay approach with the advancing front high quality point-placement strategy.

Other than the Delaunay Triangulation Algorithm and Advancing Front Technique, the *recursive domain decomposition* method [38-41] is another major branch of the 3D mesh generation schemes.

2.2.3 Mesh post-processing

Sometimes when the mesh is generated, it is not as good as been expected either in the distribution of node density or in the formation of element shape. Hence it is necessary to take some further procedures to optimize the mesh. This step is referred as *mesh post-processing*.

2.2.3.1 Mesh smoothing

Mesh smoothing is a process to improve the local quality of elements by repositioning individual nodes. Among the wide variety of smoothing algorithms, the simplest and most straightforward one is Laplacian smoothing [42]. In this method, an internal node in the mesh is repositioned by directly averaging the coordinates of the neighboring nodes. Similar to Laplacian smoothing, some other *averaging methods* [43-45] were developed and the differences among them mainly involve the use of different weighted parameters to calculate the average location of the node. Rather than relying on heuristic averaging methods, some other techniques such as optimization-based method [46, 47] and physically-based method [48, 49] were also developed. Although these techniques provide better quality than simple averaging methods, their implementation is much more time consuming.

2.2.3.2 Mesh cleanup

Mesh cleanup is another basic approach of mesh enhancement. The cleanup methods will make local changes to the element connectivity. For 2D cases, some simple mesh cleanup techniques for triangular meshes are briefly presented and listed in Table 2.1. For quadrilateral meshes, a technique named as *Topological Improvement* [50, 51] can be used by optimizing the number of edges sharing a single node. For 3D cases, the cleanup methods used for tetrahedral meshes and hexahedron meshes can be found in references [52] and [53].

2.2.3.3 Mesh refinement

Mesh refinement is a process to increase the number of elements locally or globally. It is needed when the mesh is too coarse to represent the geometry of the

Literature review

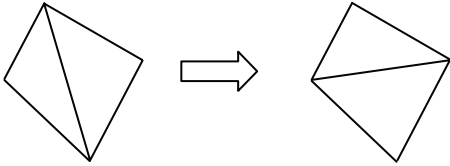
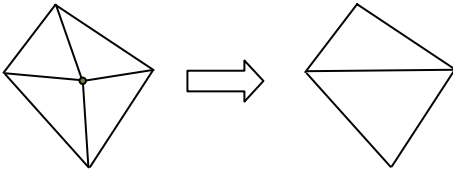
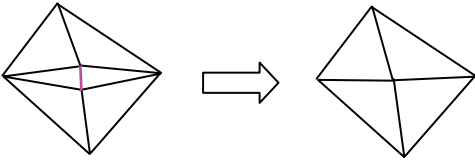
Diagonal swapping	
Description	Illustration
Swap the diagonals of the quadrilateral formed by two neighboring triangular elements	
Node removing	
Description	Illustration
Remove the node shared by several elements	
Edge Collapsing	
Description	Illustration
Remove the short edge which is shared by two neighboring elements	

Table 2.1 Some techniques of mesh cleanup for triangular elements

target object or the mesh is not fine enough for the finite element analysis to get acceptable results. There are two main kinds of mesh refinement techniques: *node insertion* and *edge bisection*. In the first method, new nodes will be added to the current mesh and then they will be connected to the existing nodes to form new elements. This approach can be also regarded as the extension of the process of mesh generation. In the second method, there is no new node that will be created inside the element. Only some or all the existing elements are divided into two or more elements by splitting their sides. The most famous method in this approach is

the *longest edge bisection technique* proposed by Rivara [54] in 1984. In this technique, elements that need to be refined will be split at the midpoint of the longest edge of this element so that the element will be divided into two new elements. The advantage of this method is that it can easily restrict the mesh refinement locally in a small region with relatively less affection to the mesh quality.

In some references, the technique to enhance the order of the element shape function is also regarded as a way of mesh refinement, which will be introduced in the later sections.

2.3 A Brief Review of the Error Estimation and Adaptive Refinement

As mentioned in Section 2.1, finite element method can only provide an approximate solution for the problem that needs to be solved. At the element boundaries, only C_0 continuity is satisfied. Hence discretization error will be generated and it can be defined as the difference between the exact solution and the numerical approximation. For most of the engineering problem, no exact solution is available, hence in practice, it is quite necessary to develop a reliable error estimation system to guarantee a certain level of accuracy of the numerical solution.

There are two categories of error estimators, namely, the *a priori error estimator* and the *a posteriori error estimator*. An *a priori* error estimator is made before the FE analysis is carried out. It is usually devised based on the mathematical formulation of the FEM and also any *a priori* knowledge of the exact solution of the problem. An *a priori* error estimator does not provide the exact magnitude of the error of the FEM solution for an individual element but only provides qualitative

information about the convergence rate of the solution when the mesh is under a given refinement. This type of estimate is useful for finite element design, and is valuable for determining the convergence rate of the method as well as for finding dependencies on problem parameters. By contrast, an *a posteriori* error estimator, is computed after the FEM solution is obtained. Therefore, it can provide both global and local error information for the FEM solution. This type of estimate is most useful for adaptivity and control of solution error.

2.3.1 *A priori* error estimation [55-60]

As in finite element analysis, piecewise polynomials are employed to represent the variation of both displacements and stresses within the finite elements, for a given point within the finite element, the convergence rate of the stresses at that point will be related to the order of approximation used. According to Taylor expansion, it is proved that if a k order polynomial is used to represent the exact solution, d , within an element of size h , then the rate of convergence of d within this element will be of order $O(h^{k+1})$. It can be further deduced that if p order polynomials are used as the element shape functions, the error of the finite element solution will be bounded by a function

$$\|\mathbf{e}_u\|_{\Omega} \leq Ch^p \quad (2.1)$$

where $\|\mathbf{e}_u\|_{\Omega}$ is the error norm and C is a constants independent of h . Equation (2.1) also shows that the FEM error is bounded by a function of the element size, h . By refining the mesh to get smaller elements, a less FEM error will be achieved.

A priori error estimate has only theoretical meaning that the approximation method (FEM) is correct in principle. In practice, the most popular error control approach is the *a posteriori* error estimation method.

2.3.2 *A posteriori* error estimation

A posteriori estimators are expressed in terms of the known finite element solution. They can be categorized into two main groups [60]:

1) *Residual type error estimator*

The residual type error estimator is first introduced by Babuska and Rheinboldt [61]. This kind of error estimator is based on evaluating equilibrium residuals, inter-element traction jumps and surface traction equilibrium. In practical application, the residuals of the FEM solution can be computed implicitly or explicitly. The implicit residual error estimators [62-65] are determined by solving local boundary value problems for the error, while the explicit residual error estimations [66-69] are expressed directly by the residuals of the discretization solution.

2) *Recovery type error estimator*

The recovery type error estimator is another approach to estimate the error by post-processing technique of the FEM solution such as the recovered gradient values from the original solution. Babuska and Miller [70] first brought forth this idea by designing a method for post-processing a FEM solution to obtain higher accuracy approximations for displacements, stresses, etc. The post-processing technique was also applied to calculate generalized stress intensity factors near corner points [71] and deal with the questions of adaptive mesh selection and *a posteriori* error estimation [72]. In 1987, Zienkiewicz and Zhu [73] created the famous Zienkiewicz-Zhu (Z-Z) error estimator. In their method, a projected stress field is constructed by using fitting approximation to the FEM solutions at some special positions within the mesh. The energy norm of the difference between the projected stress and the FEM stress is then used as an estimate of the error. The major step involved in Z-Z error estimation is the development of the

Superconvergent Patch Recovery (SPR) method [74], which will be particularly introduced in Chapter 5. Another well known recovery technique is the *Recovery by Equilibrium in Patches* (REP) [75]. The difference of these two methods is that SPR uses super-convergent points to construct the smoothed stresses while REP recovers the stresses by a weighted form of equilibrium satisfaction within a patch. Lee et al. [76] combined both the SPR and REP techniques to develop the LP procedure, which outperforms the SPR for the problems with smooth exact solution.

It was found that there is a relationship between the two types of error estimators mentioned above [77]. Both residual type error estimator and recovery type error estimator are equivalent for one-dimensional problems analytically and such equivalence also exists for two-dimensional problems numerically.

2.3.3 Adaptive Mesh Refinement

The objective of an adaptive mesh refinement program is to control the discretization error by means of increasing the number of DOFs in regions where the previous FE model is not adequate and to achieve the specified accuracy in a most economic manner. These are mainly three versions of adaptive refinement schemes, which are listed as following:

- (1) *h-version* [78-81]. This is the most common approach, which refines the mesh by increasing the number of nodes and elements, while the element types used are kept unchanged.
- (2) *p-version* [82-86]. In this approach, the mesh is kept fixed and the accuracy is improved by increasing hierarchically the order of the approximation used (i.e. the polynomial order for element shape functions).
- (3) *h-p-version* [87-91]. This approach is the combination of *h*- and *p*- refinement.

For the above three refinement strategies (h -, p - and h - p - version), there is no answer about which one is the best for general applications. This is because that the performance of different refinement strategies will be greatly influenced by the mathematical and numerical models used in the analysis. For a given application one particular type of refinement strategy can be more effective than the others and vice versa for another type of application. Generally, the h -version refinement is accepted as the most robust one and it has a very wide application range, while the p -version scheme usually requires new code development but can be very efficient especially if a high accuracy is desired and a series of different order elements is available for the problem under consideration. The h - p version is the most effective in both cost and time measures but is more difficult to implement. In this study, the h -version adaptive refinement strategy is adopted.

Generally, an h -version adaptive refinement procedure involves three essential components: i) Estimate of the error resulting from an analysis to the current mesh; ii) Prediction of a new refinement required to satisfy the user specified accuracy; and iii) Implementation of the predicted refinement by a mesh regeneration process. These three stages can be applied into computer programs and then they will be executed repeatedly until the estimated error falls within the specific accuracy range. Since the adaptive refinement can provide approximately maximum convergence rate, it is much more efficient than the uniform mesh refinement approach, especially for the analysis for some complicated problems where steep stress variations exist.

Different from the above three approaches, there is another adaptive refinement strategy referred as r -version [92-94], which means relocating the mesh points in order to get a better resolution of the solution with fixed amount of unknowns. A survey of the general ideas and effort in adaptive analysis is given in

reference [95].

2.4 Closure

The above reviewed topics are some theories related to the work of adaptive finite element analysis in this thesis. Due to the length limitation of the chapter, only a brief introduction was given here. For the algorithms and theories closely associated with the research, they will be presented in details in the following chapters.

CHAPTER 3

SURFACE MESH EXTRUSION

In Chapter 2, a brief review of the theory related to the current work is introduced. The corresponding research development is also presented. In this chapter, a new algorithm named as extrusion to convert the surface mesh to volume mesh will be explained. The process of surface mesh extrusion is illustrated in Figure 3.1.

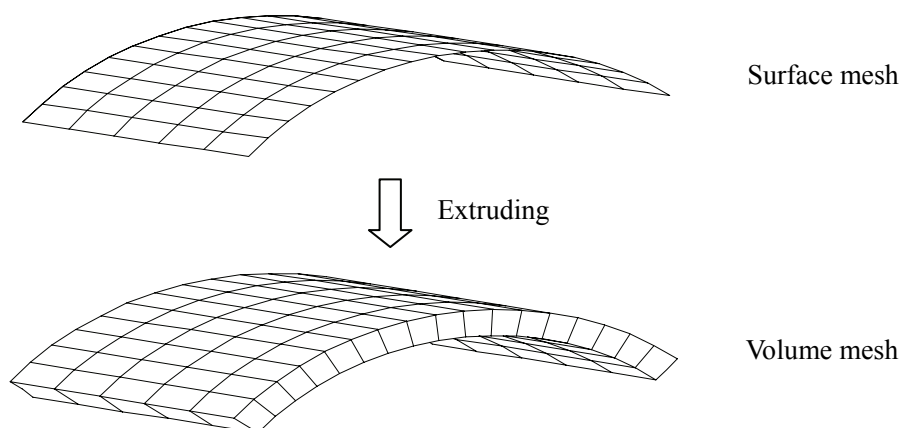


Figure 3.1 The process of surface extrusion

As an input, a surface mesh must be prepared before the extrusion step. Since an existing surface mesh generator will be used to form the surface mesh, for a better understanding of the generated surface mesh, some critical algorithms and techniques used in this surface mesh generator need to be reviewed first. After that, details of the surface mesh extrusion algorithm will be presented. How to extrude the structure with multi surfaces is the emphasis of this Chapter. Finally, some

numerical examples will be provided to show the validity of this algorithm.

3.1 Some reviews on surface mesh generation

In the current research, a triangular surface mesh generator developed by Lee [3] will be used to generate the surface mesh. This mesh generator is using Advancing Front Technique as the basic algorithm to generate the mesh. Besides that, NURBS surfaces are employed to represent the curved surfaces with complex geometries, and a technique named as Metric Specification is used for the generation of anisotropic meshes. If quadrilateral elements are required, a program developed by Lee [4] can be further applied to convert the triangular mesh into unstructured quadrilateral mesh. NURBS and Metric Specification are also employed in this program.

3.1.1 A brief review on NURBS surface

When using parametric mapping algorithm to generate surface meshes, it is needed to calculate the surface derivatives. If the surface shape equation is known, the surface derivatives can be easily obtained. But not all the curves and surfaces have their shape functions, so the use of polynomial functions based on interpolation for representing curves and curved surfaces becomes widespread. Among all the interpolation methods, NURBS curves and surfaces are the most popular ones. Now NURBS has been taken as an Initial Graphics Exchange Specification.

1) *Rational B-spline curve*

The NURBS curves and surfaces are the interpolation of the given set of points. The interpolation is carried out using *B-Spline basis* which is defined as following:

Surface mesh extrusion

Let

$$\mathbf{T} = \{t_0, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_m\} \equiv \text{knot vector} \quad (3.1)$$

where $\mathbf{T}$ is a non-decreasing sequence of real vector (*knot vector*). The i th normalized B-spline function of degree p (order $p+1$) is

$$\begin{cases} N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \end{cases} \quad (3.2)$$

where $N_{i,p}(t)$ is a p th degree piecewise polynomial function. The interval $[t_i, t_{i+1})$ is called the i th *knot span*.

Suppose the degree p has been fixed, the knot vector $\mathbf{T} = \{t_0, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_m\}$ is called *non-periodic* if the first and last knots are repeated with multiplicity $p+1$, i.e. $t_0 = t_1 = \dots = t_p$ and $t_{m-p} = t_{m-p+1} = \dots = t_m$. Furthermore, it is assumed that $t_0 = 0$ and $t_m = 1$. Hence the knot vectors have the form:

$$\underbrace{\{0, \dots, 0\}}_{p+1}, t_{p+1}, \dots, t_{m-p-1}, \underbrace{\{1, \dots, 1\}}_{p+1}.$$

If there exists a positive real number d , such that $t_{j+1} - t_j = d$ for all $p \leq j \leq m - p - 1$ (equally spaced knots), then $\mathbf{T}$ is a *uniform* knot vector; otherwise, it is called *non-uniform*. The use of non-uniform knot vectors allows better shape control and can model a much larger class of shapes than the uniform knot vectors.

2) Rational B-spline curve

The varieties of the objective model sizes cause some numerical problems. In order to remove these problems, a positive weight factor named w will be used to

Surface mesh extrusion

represent points in 3D space in terms of points in 4D space. Any non-infinite point (x, y, z) in 3D space will be mapped to the point (wx, wy, wz, w) in 4D space. Normalization map for the 4D point is defined by

$$H\{(wx, wy, wz, w)\} = \begin{cases} (wx/w, wy/w, wz/w) & \text{if } w \neq 0 \\ \text{point at infinity on the line from the} & \text{if } w = 0 \\ \text{origin through } (x, y, z) & \end{cases} \quad (3.3)$$

4D points are denoted as $\mathbf{p}^w$, and $\mathbf{p} = H\{\mathbf{p}^w\}$. (x, y, z) and $(x, y, z, 1)$ are considered to be identical. The polynomial interpolation in NURBS curve is expressed as:

$$C(t) = H\{C^w(t)\} = H\left\{\sum_{i=0}^n N_{i,p}(t) \mathbf{P}_i^w\right\} = \frac{\sum_{i=0}^n N_{i,p}(t) w_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(t) w_i} = \sum_{i=0}^n R_{i,p}(t) \mathbf{P}_i \quad (3.4)$$

where $\mathbf{P}_i$ is the i th 3D control point and $\mathbf{P}_i^w$ is the corresponding 4D control point with the weight factor w_i .

3) Rational B-spline surface

The NURBS surface is the bi-directional extension of the NURBS curve, so the polynomial interpolation is:

$$\begin{aligned} S(u, v) &= H\{S^w(u, v)\} = H\left\{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{ij}^w\right\} \\ &= \frac{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) w_{ij}} = \sum_{i=0}^m \sum_{j=0}^n R_{i,p;j,q}(u, v) \mathbf{P}_{ij} \end{aligned} \quad (3.5)$$

where $\mathbf{P}_{ij}$ is the 3D control point and

$$R_{i,p;j,q}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{ij}}{\sum_{r=0}^m \sum_{s=0}^n N_{r,p}(u) N_{s,q}(v) w_{rs}} \quad (3.6)$$

With this function, tangent vectors and normal vectors of the surface can be easily obtained by calculating the derivatives.

3.1.2 A brief review on metric specification

Metric specification is proposed by Borouchaki et al. [25, 24] to control the element size and shape for anisotropic mesh generation. It can also be used in surface meshing to control the element shape in parametric space [3]. A brief summary of the metric specification is given as the following.

In 2D space, a *metric tensor* that defines the user specification of element size characteristics is written as

$$\mathbf{M} = [\mathbf{e}_1 \quad \mathbf{e}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2]^T \quad (3.7)$$

$(\mathbf{e}_i, \lambda_i)$, $i = 1, 2$ are the *eigenpairs* of $\mathbf{M}$ and $\lambda_i = \frac{1}{(h_i)^2} > 0$ where h_i is the *principal element size* in $\mathbf{e}_i$ direction. The eigenpair, $(\mathbf{e}_i, \lambda_i)$, will determine the stretching ratio and direction of the mesh.

The distance between two points $\mathbf{P}_1$ and $\mathbf{P}_2$, $\tilde{l}(\mathbf{M}, \mathbf{P}_1 \mathbf{P}_2)$, is defined as

$$\tilde{l}(\mathbf{M}, \mathbf{P}_1 \mathbf{P}_2) = \int_0^1 \sqrt{\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t)^T \cdot \mathbf{M}(\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t)) \cdot \mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t)} dt \quad (3.8)$$

where

$$\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t) = \mathbf{P}_1 + t(\mathbf{P}_2 - \mathbf{P}_1) \quad 0 \leq t \leq 1 \quad (3.9)$$

$\mathbf{M}(\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t))$ is the interpolation of $\mathbf{M}_1$ and $\mathbf{M}_2$, which are the metric tensors defined at $\mathbf{P}_1$ and $\mathbf{P}_2$. Details of the interpolation can be found in reference [11, 25].

In the 3D space, a metric tensor is written as

$$\mathbf{M}_{3D} = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3]^T \quad (3.10)$$

During surface mesh generation procedure, a surface metric tensor in parametric space is defined as [3]

$$\mathbf{M}_{sur} = (\mathbf{r}_{,u} \quad \mathbf{r}_{,v})^T \mathbf{M}_{3D} (\mathbf{r}_{,u} \quad \mathbf{r}_{,v}) \quad (3.11)$$

where $(\mathbf{r}_{,u} \quad \mathbf{r}_{,v})$ is a bi-variate mapping with the form

$$(\mathbf{r}_{,u} \quad \mathbf{r}_{,v}) = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix} \quad (3.12)$$

(u, v) are the parametric coordinates of the node mapped from 3D space. The surface metric tensor combines the effects of the user specification and the surface mapping.

With the surface metric tensor defined by Equation (3.11) and the length measurement defined by Equation (3.8), the element shape and quality can also be measured. The method to define the shape and quality measures will be reviewed in next chapter.

3.2 Extrusion for Single Surface

It is considered that the thickness of the researching object is much smaller than the sizes in surface direction, so the thin-walled structure can be formed by the process of extrusion along the surface normal direction of the input surface (Figure 3.2).

Surface mesh extrusion

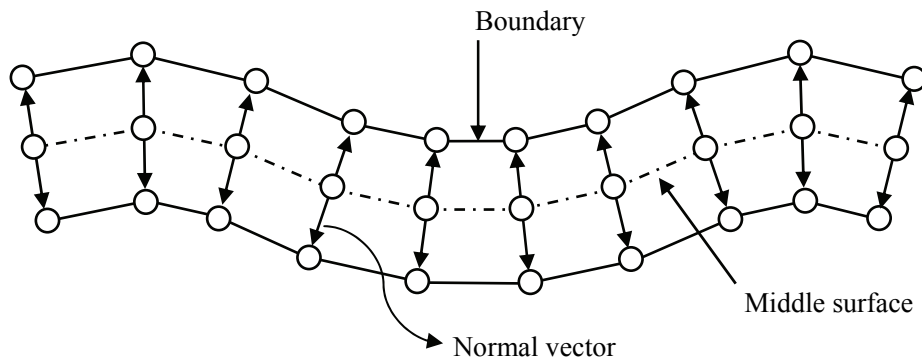


Figure 3.2 Extrusion along normal direction

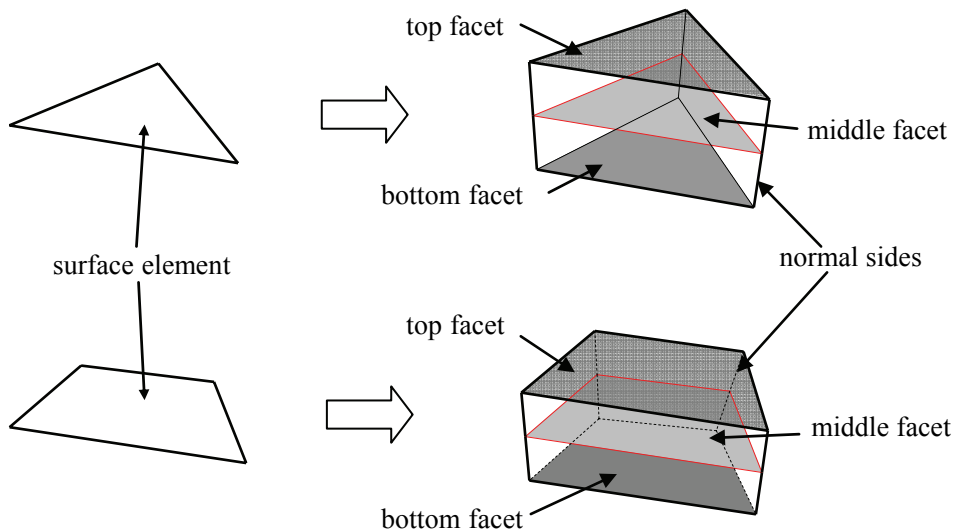


Figure 3.3 Extrusion of surface elements to volume elements

After the extrusion, each mid-surface element will be converted to a volume element. Triangular elements are converted to prism volume elements and quadrilateral elements are converted to hexahedral volume elements (Figure 3.3). Figure 3.3 also illustrates that after extrusion, the original mid-surface element becomes a *middle facet* and locates inside the volume element.

By contrast, the top and bottom surfaces of the volume element are defined as the *top facet* and the *bottom facet* respectively. In addition, *normal sides*, which

Surface mesh extrusion

defined the extrusion directions at the corner nodes, are generated during the extrusion process.

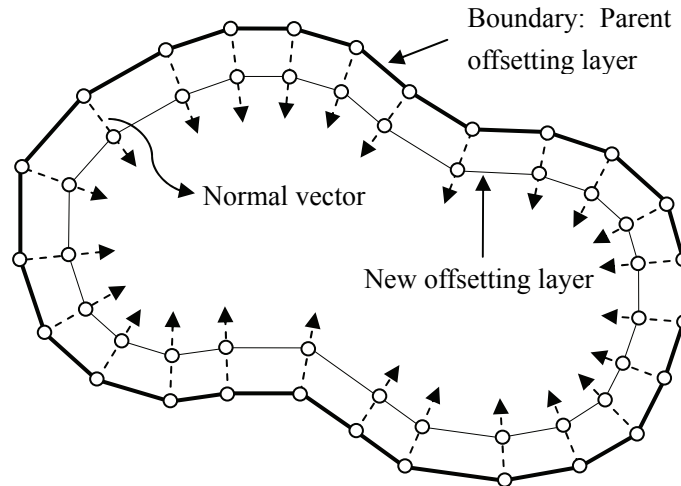


Figure 3.4 The process of normal offsetting

Note that the extrusion process used here is somehow similar to the normal offsetting technique presented by Johnston [96, 97]. Normal offsetting technique is used for meshing a closed domain and the offsetting process is advancing inward from the boundary (Figure 3.4). The *offsetting technique* was also used by Lohner et. al. [98] and Marcum [99] for volume meshing in *Computational Fluid Dynamics* studies. Different from offsetting techniques, the extrusion technique is used for the meshing of solid domain of the TWS and the extruding process is executed along two opposite directions normal to the middle surface (Figure 3.2). Furthermore, a similar approach for generating boundary layer mesh for solid domain was purposed by Athanasiadis and Deconinck [100]. However, in reference [100], the algorithm developed is mainly for the generation of boundary mesh with uniform layers of elements and no detailed discussion for the treatment of surface intersections were given. Recently, another remarkable approach for generating 3D solid meshes for thin-walled solids was purposed by Quadros and K. Shimada [101]. The algorithm purposed in reference [101] uses a special algorithm to extract the

Surface mesh extrusion

chordal surface from the 3D solid model. An all quad-element mesh generator is then applied to discretize the surface model before the final all-hex mesh with uniform layers of elements along the surface direction is generated by simple extrusion process.

In the current implementation, the extrusion algorithm is designed in such a way that it will make use of the following input data:

- (1) A surface mesh of the mid-surface of the TWS.
- (2) The thickness, h , and the surface normal vectors, $\mathbf{b}$, of the TWS at the nodal points of the surface mesh.
- (3) The nodal number of layers of elements, NEL, required to be generated in the thickness direction of the elements of the surface mesh.

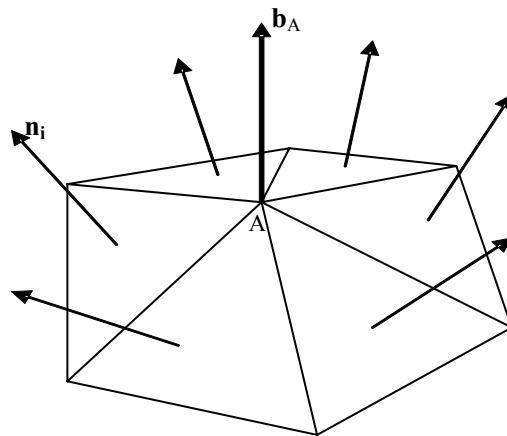


Figure 3.5 Normal vector at a node

It should be mentioned that the nodal surface vector is only an optional input for the conversion process as it is possible to compute them based on the coordinates of the nodal points and the connectivity of the surface mesh. For a given surface node A , the *optimal unit normal vector*, $\mathbf{b}_A$, at that node is calculated by first retrieving the normal vectors of its adjacent elements (Figure 3.5). The normal vector $\mathbf{b}_A$ can then be obtained by applying the *least-squares fitting* (LSF)

Surface mesh extrusion

method. During the fitting process, the error function contribution due to the i th adjacent elements [99] used is defined as

$$\mathbf{e}_i = 1 - \mathbf{b}_A \cdot \mathbf{n}_i \quad (3.13)$$

where $\mathbf{n}_i = [n_i^x, n_i^y, n_i^z]^T$ is the surface normal vector of the i th adjacent element. By using the LSF technique, the vector $\mathbf{b}_A$ can be obtained by minimizing the sum $\sum_{i=1}^{NB_i} \mathbf{e}_i^2$ and the resulting equations are:

$$\begin{cases} \sum_{i=1}^{NB_i} (\mathbf{b}_A \cdot \mathbf{n}_i) \cdot n_i^x = \sum_{i=1}^{NB_i} n_i^x \\ \sum_{i=1}^{NB_i} (\mathbf{b}_A \cdot \mathbf{n}_i) \cdot n_i^y = \sum_{i=1}^{NB_i} n_i^y \\ \sum_{i=1}^{NB_i} (\mathbf{b}_A \cdot \mathbf{n}_i) \cdot n_i^z = \sum_{i=1}^{NB_i} n_i^z \end{cases} \quad (3.14)$$

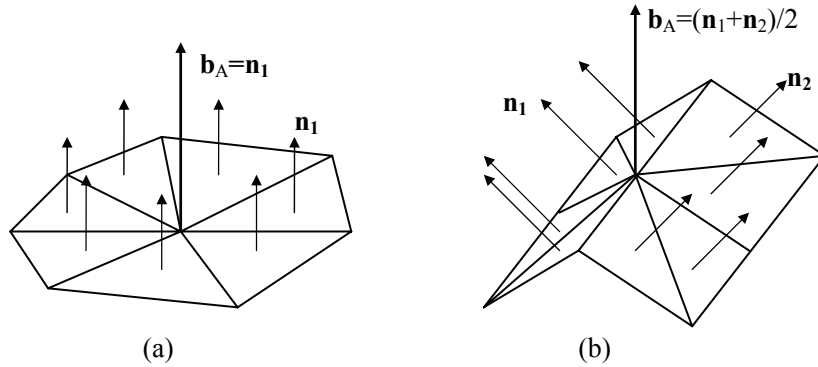


Figure 3.6 Degenerated cases for the fitting procedure:

(a) only one distinct normal $\mathbf{n}_1$ exist; (b) two distinct normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ exist

After the $\mathbf{b}_A$ is computed it will be normalized and volume elements will be created by extrusions along the $+\mathbf{b}_A$ and $-\mathbf{b}_A$ directions for a distance equal to $h_A/2$ where h_A is the thickness of the TWS at node A. It should be remarked that Equation (3.14) will only have unique solution provides that there exists at least three distinct normal vectors $\mathbf{n}_i$. In such cases that only one (Figure 3.6a) or two (Figure 3.1b) distinct normal vectors exist, the only normal vector $\mathbf{n}_1$ or the average of the two distinct normal vectors $(\mathbf{n}_1 + \mathbf{n}_2)/2$ will be used as the optimal unit vector

$\mathbf{b}_A$, respectively.

Note that for the computation of the normal vectors, LSF is used rather than the *weighted least-squares fitting* (WLSF) (e.g. used the areas of adjacent elements as weights) or the *simple averaging* (SA) method because it is well-known that for a given surface model, the WLSF and SA procedures could produce variations in normals due to purely topology (connectivity of elements surrounding a node) or local face areas difference. A simple LSF approach eliminates such variations.

3.3 Treatments for Surfaces Intersections

In order to generate volume meshes for practical engineering applications, it is essential that the surface extrusion procedure should be extended to cover surface intersections. In this study, special algorithms are developed for the treatments of the following surface intersection situations:

- (i) *Simple surface intersections* in which only two surfaces are involved (Figure 3.7a)
- (ii) *Multi-cross intersections* in which more than two surfaces are involved (Figure 3.7b)
- (iii) *Corner point intersections* in which a corner point is created (Figure 3.7c).

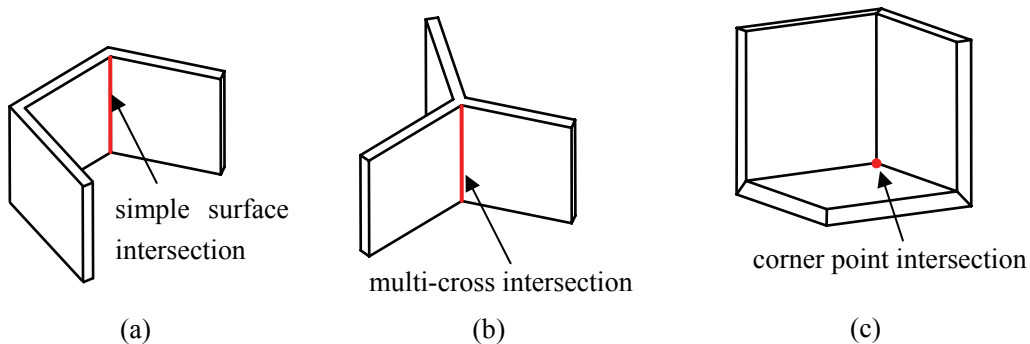


Figure 3.7 Three types of structures with surface intersections: (a) a simple surface intersection; (b) a multi-cross intersection; and (c) a corner point intersection

3.3.1 Simple surface intersections

For the treatment of simple surface intersection involving two surfaces, surface I and surface II (Figure 3.8), the final solid mesh will be formed in two steps. In the first step, the simple extrusion algorithm will be applied to the two intersecting surfaces independently. In the second step, modification algorithms are employed to treat those overlapping elements formed near the junction of the TWS.

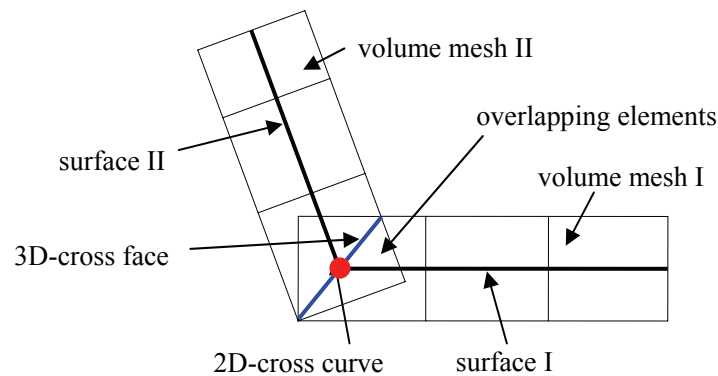


Figure 3.8 Overlapping elements, 2D-cross curve and 3D-cross face

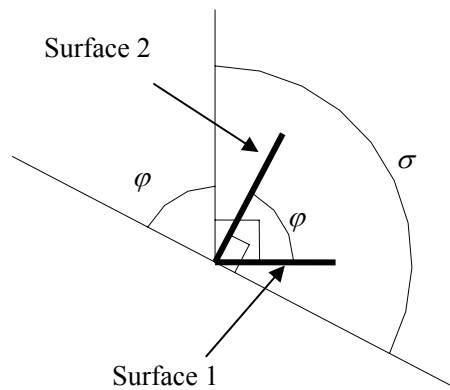


Figure 3.9 Intersection angles between two intersecting surfaces

In order to explain the algorithm used, two definitions will be given first. The intersection curve of the two mid-surfaces is defined as the *2D-cross curve* while the curve surface formed by the intersection of the two face walls of the two volume

Surface mesh extrusion

meshes is referred as the *3D-cross face* (Figure 3.8). In addition, the angle between the two intersection surfaces is denoted as φ (Figure 3.9). In general, the value of φ varies along the 2D-cross curve and it will only be constant when the two surfaces are both planar. The values of φ at all the nodes lying on the 2D-cross curve could be computed straightforwardly using the surface mesh information. For each node on the 2D-cross curve, the angle between the two normal vectors respect to each surface is defined as σ (Figure 3.9) such that

$$\varphi = \pi - \sigma \quad (3.15)$$

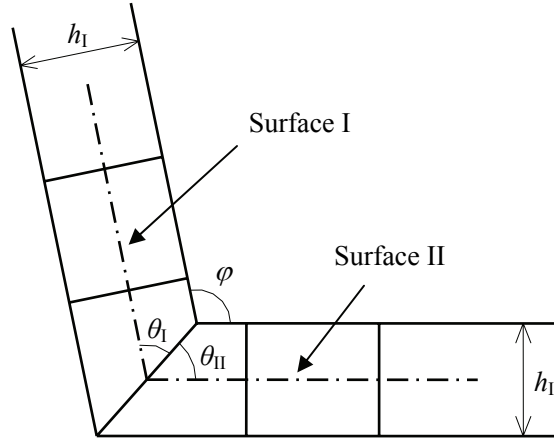


Figure 3.10 Definitions of φ , θ_I and θ_{II}

Let θ_I and θ_{II} be the angles between the 3D-cross face and surfaces I and II, respectively (Figure 3.10). In addition, the thickness of the surfaces at the nodes lying on the 2D-cross curve will be retrieved. As shown in Figure 3.10, if the thicknesses of the two surfaces at the nodes under consideration are equal to h_I and h_{II} , respectively, then θ_I and θ_{II} can be computed as:

$$\left\{ \begin{array}{l} \theta_I + \theta_{II} = \varphi \\ \frac{h_I}{2\sin\theta_I} = \frac{h_{II}}{2\sin\theta_{II}} \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \theta_I = \tan^{-1}\left(\frac{h_I \sin\varphi}{h_I \cos\varphi + h_{II}}\right) \\ \theta_{II} = \tan^{-1}\left(\frac{h_{II} \sin\varphi}{h_{II} \cos\varphi + h_I}\right) \end{array} \right. \quad (3.16)$$

By using Equation (3.16), the normal sides for all the nodes lying on the 2D-cross curve can be modified such that they will be conformal with the 3D-cross face of the

intersection.

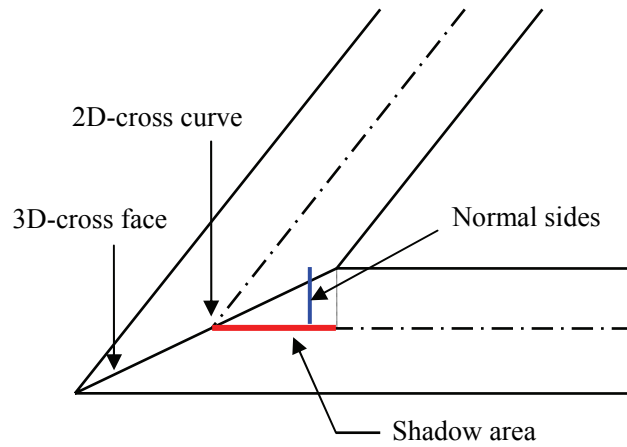


Figure 3.11 The situation when φ is small

3.3.1.1 Re-definition of surface normal vectors for simple surface intersection

When the angle φ is small and the thickness of the surfaces I and II are larger than the element size near the intersection, some normal sides of the volume elements near the intersection may cut into the 3D-cross face (Figure 3.8). In general, when projecting the 3D-cross face onto the intersecting surfaces, a *shadow area* is generated (Figure 3.11). Note that both intersection surfaces have their own shadow areas. If there are surface nodes located inside these shadow areas, it can be affirmed that the normal sides passing through these nodes will intersect with the 3D-cross face and their directions must be re-defined. Furthermore, it is found that if the normal side of a given node is re-defined, it is often required to modify the normal sides associated with all the nodes adjacent to it. As a result, all the surface nodes that are located inside the shadow area and their adjacent nodes will be collected into a set denoted as $\mathcal{J}$. All the nodes belong to $\mathcal{J}$ will be checked and their original optimal surface normal vectors (Equation (3.14)) may be re-defined. Nodes belong $\mathcal{J}$ will be grouped in layers according to their connectivity with the 2D-cross curve (Figure

Surface mesh extrusion

3.12). By starting from the first node in the first layer, nodes are checked layer by layer until no normal side is modified in the current layer.

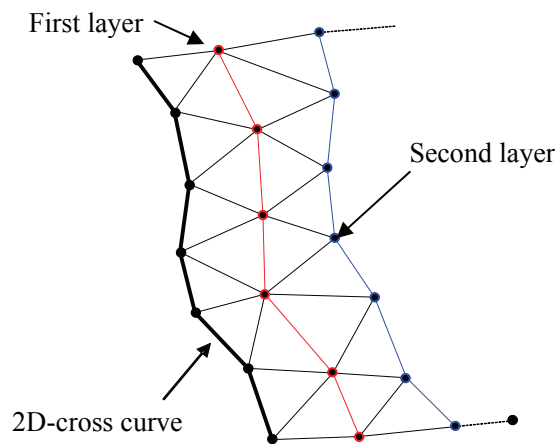


Figure 3.12 Surface node layers near the 2D-cross curve

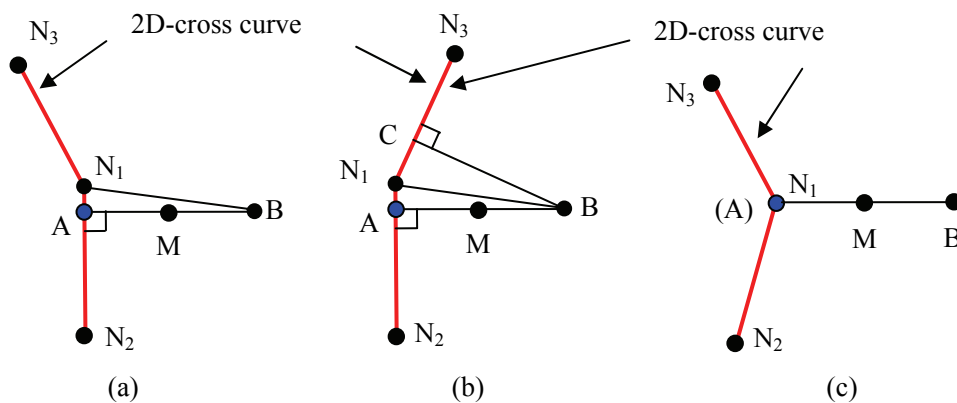


Figure 3.13 Location of perpendicular point A on the 2D-cross curve:

- (a) $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 < 90^\circ$; (b) $\angle BN_1N_3 < 90^\circ$ and $\angle BN_1N_2 < 90^\circ$; (c) $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 > 90^\circ$

For the given node $B \in \mathcal{J}$ to be checked, a connection vector will be drawn from node B to a point A such that BA is perpendicular to the 2D-cross curve at A (Figure 3.13 and Figure 3.14a). Since the 2D-cross curve is discretized into line segments, the following three situations may occur:

Surface mesh extrusion

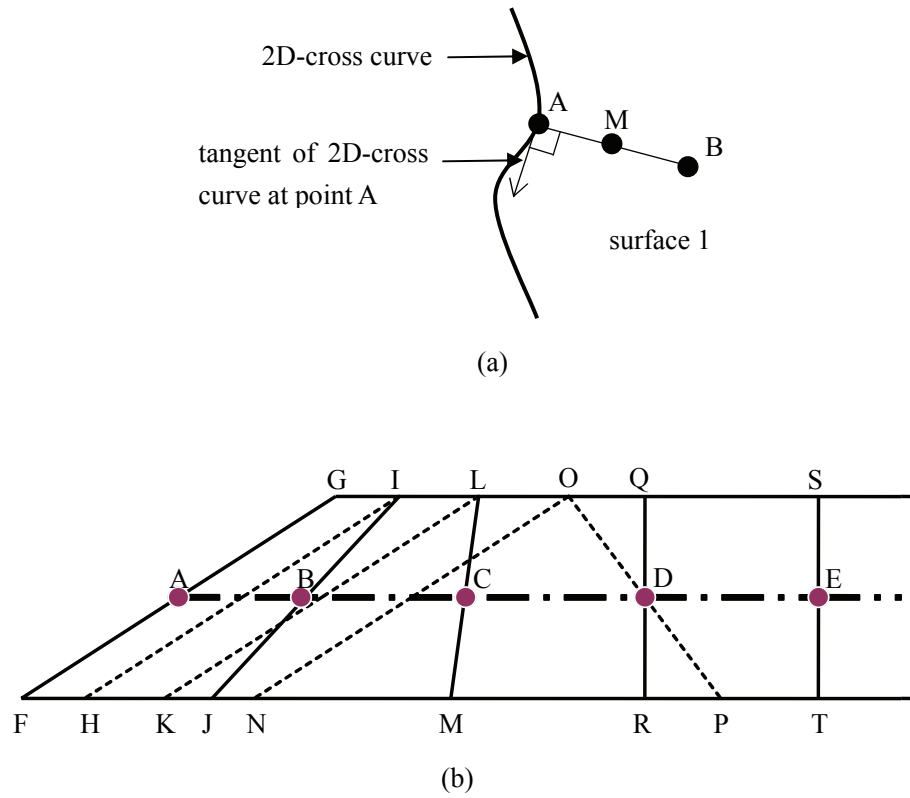


Figure 3.14 Re-definition of normal sides for nodes: (a) plan view; (b) section view

Situation I: $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 < 90^\circ$ (Figure 3.13a). In this case, the perpendicular point A is located on the segment N_1N_2 .

Situation II: $\angle BN_1N_2 < 90^\circ$; (b) $\angle BN_1N_3 < 90^\circ$ (Figure 3.13b). In this case, two perpendicular points A and C can be found on the segments N_1N_2 and N_1N_3 respectively, and any one of them could be used. In the current implementation, point A will be selected if the numbering of N_2 is less than that of N_3 and vice versa.

Situation III: $\angle BN_1N_3 > 90^\circ$ and $\angle BN_1N_2 > 90^\circ$ (Figure 3.13c). In this case, no perpendicular point can be found and the nearest point to B, N_1 will be selected.

After the location of point A is defined, the surface normal vector at A, $\mathbf{b}_A$ will be calculated. Assume that the optimal surface normal vector at node N_1 and N_2 are $\mathbf{b}_{N_1}$ and $\mathbf{b}_{N_2}$ respectively. For situations I and II, $\mathbf{b}_A$ can be obtained by linear

Surface mesh extrusion

interpolation as

$$\mathbf{b}_A = \frac{\mathbf{b}_{N_1}|AN_2| + \mathbf{b}_{N_2}|N_1A|}{|N_1N_2|} \quad (3.17)$$

while $\mathbf{b}_A$ will be taken as $\mathbf{b}_{N_1}$ for situation III.

After $\mathbf{b}_A$ is determined, the re-defined normal direction for node B will be re-calculated according to the algorithm given in Box 3.1 using point B in Figure 3.14 as an example.

- (i) A connection vector BA will be drawn from node B to point A.
- (ii) The original normal direction, $\mathbf{b}_B$, and thickness, h_B , at node B and the original normal side at B, $\mathbf{V}_B$ will be retrieved.
- (iii) The angle between the line AB and $\mathbf{V}_B$ will be computed. If this angle is less than $\pi/2$, then the normal $\mathbf{V}_B$ will be kept unchanged and goto (vii). Otherwise, the angle between $\mathbf{b}_A$ and $\mathbf{b}_B$ is computed and denoted as δ .
- (iv) The location of point M, the midpoint of BA, will be found.
- (v) From M a line HI will be drawn along the $\mathbf{b}_A$ direction such that $|IH| = h_B/\cos\delta$.
- (vi) The line IB is then connected and extended to point J (Figure 3.14b). The line JI will then be re-defined as the new normal side for node B by setting $\mathbf{V}_B \leftarrow \vec{JI}$. The new optimal normal direction will be re-defined as $\mathbf{b}_B \leftarrow \vec{JI}/|\vec{JI}|$.
- (vii) Quit with the new normal side $\mathbf{V}_B$ and new optimal normal direction $\mathbf{b}_B$.

Box 3.1. Re-definition of normal side and surface normal direction at node B

As shown in Figure 3.14b, the new normal sides for all other nodes belong to $\mathcal{J}$ will be obtained in a similar way. Note that when the node under consideration is

close to the 2D-cross curve, the angle between the connecting line and the final normal side will be an obtuse angle (e.g. nodes B and C in Figure 3.14b). When the node is far from the 2D-cross curve, the value of this angle will decrease. Once this angle becomes an acute angle, the original normal side of the node will not be changed (e.g. node D in Figure 3.14b).

3.3.1.2 Adjustment of normal sides near surface intersections

It is found that even though the normal sides near the 2D-cross curve are modified according to the method described in the last section, invalid volume elements may still be formed after the extrusion process. Such situation is more common when the surface mesh is highly anisotropic and contains a large number of highly stretched elements near the surface intersection. A typical case of an invalid volume element is shown in Figure 3.15 in which the mid-surface element ABC is extruded along directions AA', BB' and CC' at nodes A, B and C respectively. It can be seen that due to the rapid change in the surface normal directions, the normal vectors for the middle facet ABC (N) and upper facet A'B'C' (N') are opposite to each other such that $N \cdot N' < 0$ and resulted in an invalid volume element.

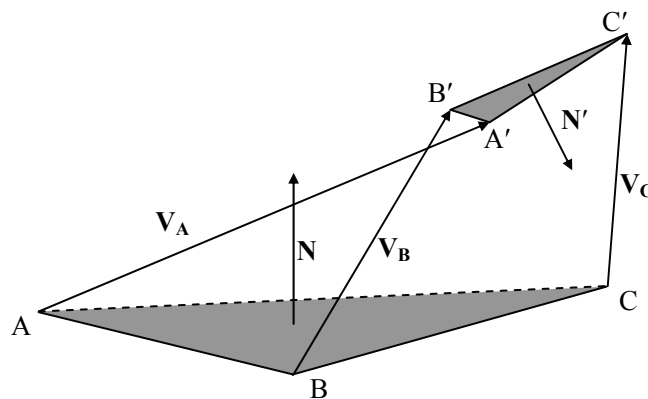


Figure 3.15 Upper half of an invalid volume element formed after extrusion

Surface mesh extrusion

In order to avoid the formation of invalid volume elements, an additional step is implemented to further adjust the normal sides of nodes belong to the set $\mathcal{J}$. A normal side that requires further adjustment is called an *invalid normal side*. For a given normal side $\mathbf{V}_B$ (which may already be re-defined according to the procedure described in the previous section) associated with a node $B \in \mathcal{J}$, its validity will be checked by the procedure shown in Box 3.2.

- (i) Retrieve all the volume elements adjacent to the node B .
- (ii) For a given element retrieved, if all its normal sides have been checked according to the procedure described in Box 3.1, the validity of the element will be checked by computing the dot product $\mathbf{N} \cdot \mathbf{N}'$ where $\mathbf{N}$ and $\mathbf{N}'$ are the normal vectors corresponding to the middle and top facets of the volume element respectively (Figure 3.14). A volume element is invalid if $\mathbf{N} \cdot \mathbf{N}' \leq 0$.
- (iii) If there exists at least one invalid element adjacent to node B , $\mathbf{V}_B$ is considered as invalid and it will be further adjusted, otherwise, $\mathbf{V}_B$ is valid and no adjustment is required.

Box 3.2. Checking the validity of a given normal side $\mathbf{V}_B$

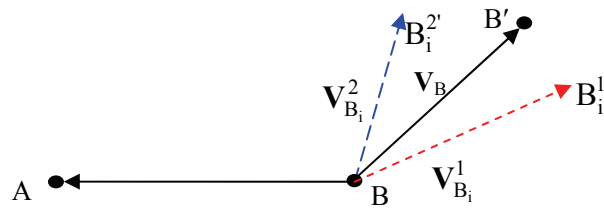


Figure 3.16 Adjustment of normal side $\mathbf{V}_B$

Assume that now the normal side $\mathbf{V}_B$ is invalid. The new direction of the adjusted normal sides will be established by perturbing it about its original direction as shown in Figure 3.16. In Figure 3.16, $\mathbf{BA}$ is the connecting vector to the 2D-cross curve (Box 3.1) such that it is perpendicular to the 2D-cross curve at A (Figure 3.13).

Surface mesh extrusion

The new direction of the normal side will be determined by perturbing the vector $\mathbf{VB}$ on the plane defined by the points A, B and B' (Figure 3.15). Two series of perturbed vectors $\mathbf{V}_{B_i}^j$ will be generated such that

$$\mathbf{V}_{B_i}^j = \mathbf{V}_B \left(1 - \frac{i}{ND}\right) + (-1)^j \frac{i}{ND} (\overrightarrow{BA}) \quad \text{for } j = 1, 2 \text{ and } i = 1, \dots, ND \quad (3.18)$$

In Equation (3.18), ND is the maximum number of divisions used for the generation of perturbed vectors. From Equation (3.18), it can be seen that for the first series, $\mathbf{V}_{B_i}^1$, the normal side direction is shifted away from the surface intersection and the value of the angle $\angle ABB'$ is increased and vice versa for the second series $\mathbf{V}_{B_i}^2$. For any normal vector $\mathbf{V}_{B_i}^j$, the corresponding normal side for the upper half of the volume element, $BB_i^{j'}$, will be generated in such a way that when it is projected back to the original optimal normal direction $\mathbf{b}_B$ (defined in Equation(3.14)), its length will be equal to half of the thickness at the node B. In practice, in order to minimize the change in the normal side direction during the adjustment, the series $\mathbf{V}_{B_i}^1$ will be employed to generate the new normal side first. The generation sequence of the perturbed vectors for node B is shown in Box 3.3.

- (i) Loop over the index j from 1 to 2.
- (ii) Loop over the index i from 1 to ND.
- (iii) Determine the vector $\mathbf{V}_{B_i}^j$ using Equation (3.18) and the corresponding new normal side.
- (iv) Test whether the new normal side generated is a valid normal side or not. If it is a valid normal side, quit with the valid normal side.
- (v) End for loop i
- (vi) End for loop j

Box 3.3. Adjustment of invalid normal side $\mathbf{V}_B$

Surface mesh extrusion

The above adjustment procedure will be applied to all normal sides belong to J in a layer by layer manner similar to the case of normal side re-definition. Numerical tests show that by setting $ND=20$ in Equation (3.18), the adjustment procedure would normally remove all invalid normal sides even when the thickness of the surfaces is much larger than the element size along an acute intersection for a highly anisotropic mesh. In case that there are still invalid normal sides after the first pass of the adjustment procedure, it will be repeated until all the invalid normal sides are corrected. Numerical experiences done show that it is rarely required more than one iteration to remove all invalid normal sides even for the most adverse case tested.

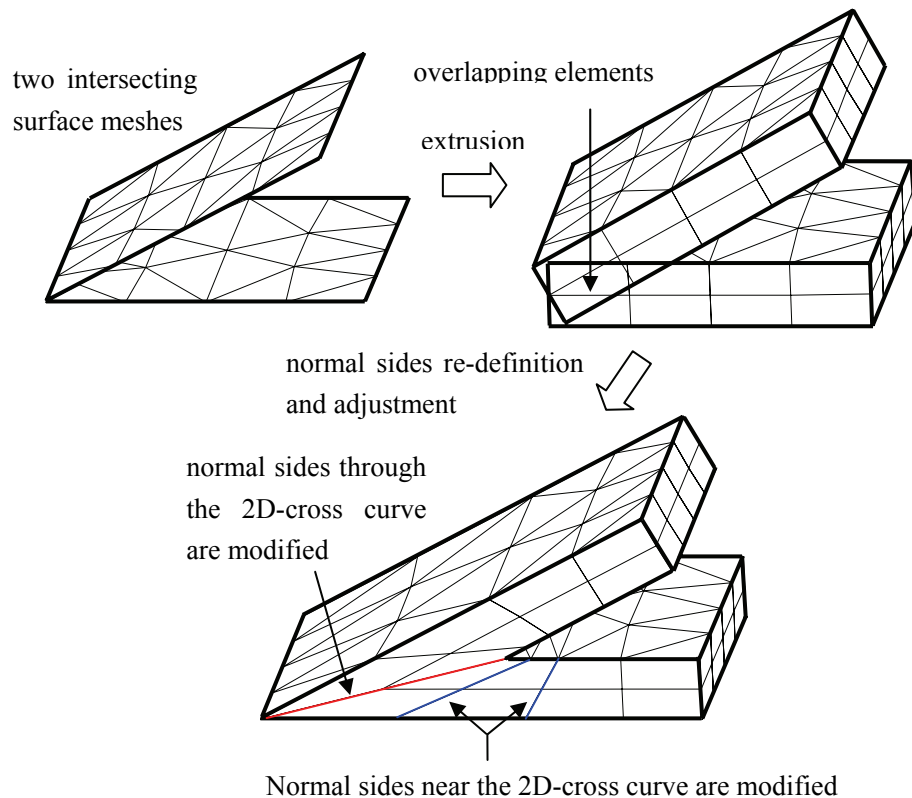


Figure 3.17 Treatment of volume meshes for two intersecting surface meshes

A simple example to demonstrate the actions for the normal side re-definition and adjustment process is given in Figure 3.17. In this example, the structure modeled consists of two planar surfaces intersecting at 30° and the thicknesses of

shells equal to the average element size near the intersection. As a result, nine normal sides near the 2D-cross face are modified.

3.3.2 Multi-cross intersections

During the treatment of multi-cross intersection such as the one shown in Figure 3.7b, each 3D-cross face and the volume elements generated for the two surfaces will be divided into two separate parts by its mid-surface and considered separately (Figure 3.18). As a result, on each surface, two layers of volume elements will be formed.

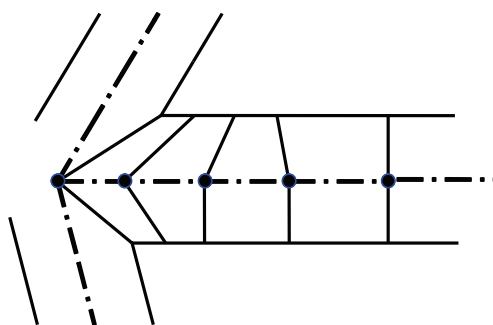


Figure 3.18 Division of a 3D-cross face and volume mesh to two separate parts

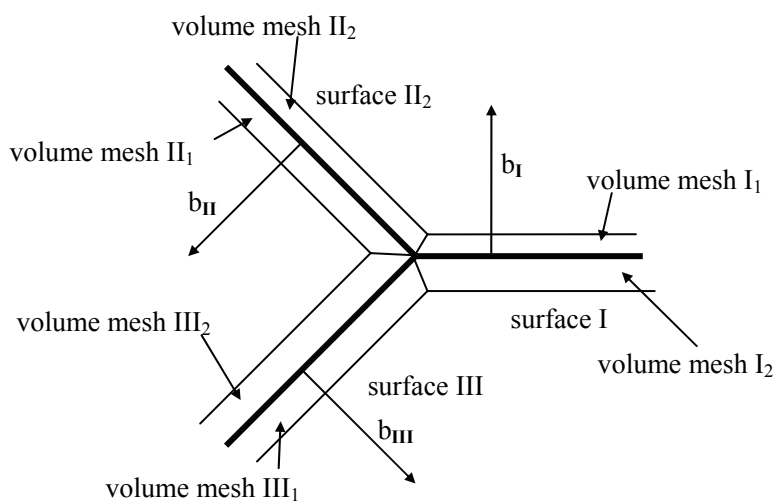


Figure 3.19 Multi-cross and the normal directions

Surface mesh extrusion

In order to use Equation (3.15) and (3.16) to calculate the normal sides on the 2D-cross curve, normal vectors of the intersection surfaces will be reversed. For example, for a joint formed by the intersection of three surfaces I, II and III as shown in Figure 3.19, only one unique 2D-cross curve is formed. However, during the treatment of this multi-cross intersection, three separated 2D-cross curves (and hence, 3D-cross faces) and six volume meshes will be created. After the surface normal vectors are defined, volume meshes can be generated by applying the procedure described in Section 3.2 on appropriate pairs of normal vectors in turns. The different surfaces and normal vectors used for the formation of 3D-cross curves and volume meshes for the multi-cross intersection shown in Figure 3.19 are listed in Table 3.1. In general, for a surface intersection involving NS surfaces, NS 3D-cross faces and $2 \times NS$ surface normal vectors and volume meshes will be created.

2D-cross curves/ 3D-cross face	Surfaces used	Surface normal vectors used	volume mesh formed
1	I, II	$b_I, -b_{II}$	I_1, II_2
2	II, III	$b_{II}, -b_{III}$	II_1, III_2
3	III, I	$b_{III}, -b_I$	III_1, I_2

Table 3.1. Normal vectors of surfaces for different 2D-cross curve

3.3.3 Corner point intersections

For a multi-cross intersection involving a *corner point* which is formed by the intersection of three 2D-cross curves (Figure 3.7c). The normal side for the node at the corner point is defined as the intersection line of three 3D-cross faces and its direction is determined by the surface normal vectors and the thicknesses of the three intersection surfaces.

Surface mesh extrusion

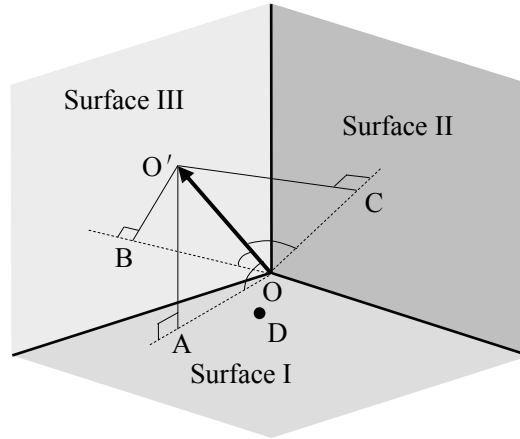


Figure 3.20 Calculation of the normal side through the corner point

As shown in Figure 3.20, the surface node corresponding to the corner point is denoted as O and OO' is the normal side to be determined. Let points A, B and C are located on the bottom facets of the volume meshes corresponding to surfaces I, II and III respectively. The lines $O'A$, $O'B$ and $O'C$ are constructed so that they are perpendicular to surfaces I, II and III respectively. If the thicknesses of the TWS at point O for the three surfaces are equal to h_I , h_{II} and h_{III} respectively, then the coordinates of point O' , $(x_{O'}, y_{O'}, z_{O'})$, can be calculated by solving the following equations:

$$\begin{aligned} |OO'| \cos(\angle OO'A) &= h_I \\ |OO'| \cos(\angle OO'B) &= h_{II} \\ |OO'| \cos(\angle OO'C) &= h_{III} \end{aligned} \quad (3.19)$$

Equation (3.19) can be rewritten as

$$\begin{aligned} (x_{O'} - x_O) \cdot b_{I,x} + (y_{O'} - y_O) \cdot b_{I,y} + (z_{O'} - z_O) \cdot b_{I,z} &= h_I \\ (x_{O'} - x_O) \cdot b_{II,x} + (y_{O'} - y_O) \cdot b_{II,y} + (z_{O'} - z_O) \cdot b_{II,z} &= h_{II} \\ (x_{O'} - x_O) \cdot b_{III,x} + (y_{O'} - y_O) \cdot b_{III,y} + (z_{O'} - z_O) \cdot b_{III,z} &= h_{III} \end{aligned} \quad (3.20)$$

where (x_O, y_O, z_O) are the coordinates of point O while $\mathbf{b}_\eta = (b_{\eta,x}, b_{\eta,y}, b_{\eta,z})$ for $\eta = I, II$ and III is the surface normal vector for surface η at the point O.

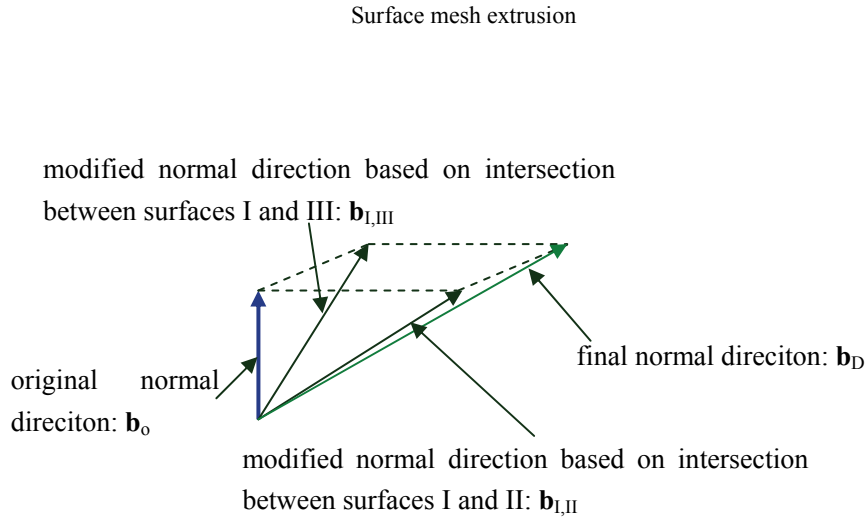


Figure 3.21 Determine the final normal side for node A near a corner point

Similar to the case for multi-cross intersection without corner point, when the angles between intersection surfaces near the corner point are small or when the thickness of the TWS is large when comparing with the mid-surface element size, normal sides near the corner point will be re-defined and adjusted to avoid the formations of invalid elements and normal sides. For a multi-cross intersection with corner node, normal sides of surface nodes near the corner point will be affected by two pairs of intersecting surfaces. Hence, the final direction of the normal side will be determined by combining the two modified normal vectors obtained by consideration of the two pairs of surface intersections. For example, in Figure 3.20, node D is a surface node close to the corner point O and its original normal surface vector (obtained by using Equation (3.14)) is denoted as $\mathbf{b}_o$. The final normal vector for node D, $\mathbf{b}_D$ will be obtained by combining $\mathbf{b}_{I,II}$, the modified normal obtained by considering the intersection between surfaces I and II and $\mathbf{b}_{I,III}$, the modified normal obtained by considering the intersection between surfaces I and III. Note that the vectors $\mathbf{b}_{I,II}$ and $\mathbf{b}_{I,III}$ are obtained by applying the algorithms described in Section 3.2 on $\mathbf{b}_o$ separately (Figure 3.21) and $\mathbf{b}_D$ can be expressed as

$$\mathbf{b}_D = (\mathbf{b}_{I,II} - \mathbf{b}_o) + (\mathbf{b}_{I,III} - \mathbf{b}_o) + \mathbf{b}_o \quad (3.21)$$

3.4 Mesh Extrusion Examples

In this section, several examples will be presented to show the effect of the surface mesh extrusion algorithm developed. Examples 3.1 to 3.4 are employed to demonstrate the main actions of the algorithms presented in Sections 3.2 to 3.3. The structures involved are constructed by single surfaces or simply intersected shells. Examples 3.5 and 3.6 are more realistic TWSs formed by complex intersections of surfaces. These two examples are used to show the validity of the presented algorithm for the extrusion of the surface mesh for TWSs.

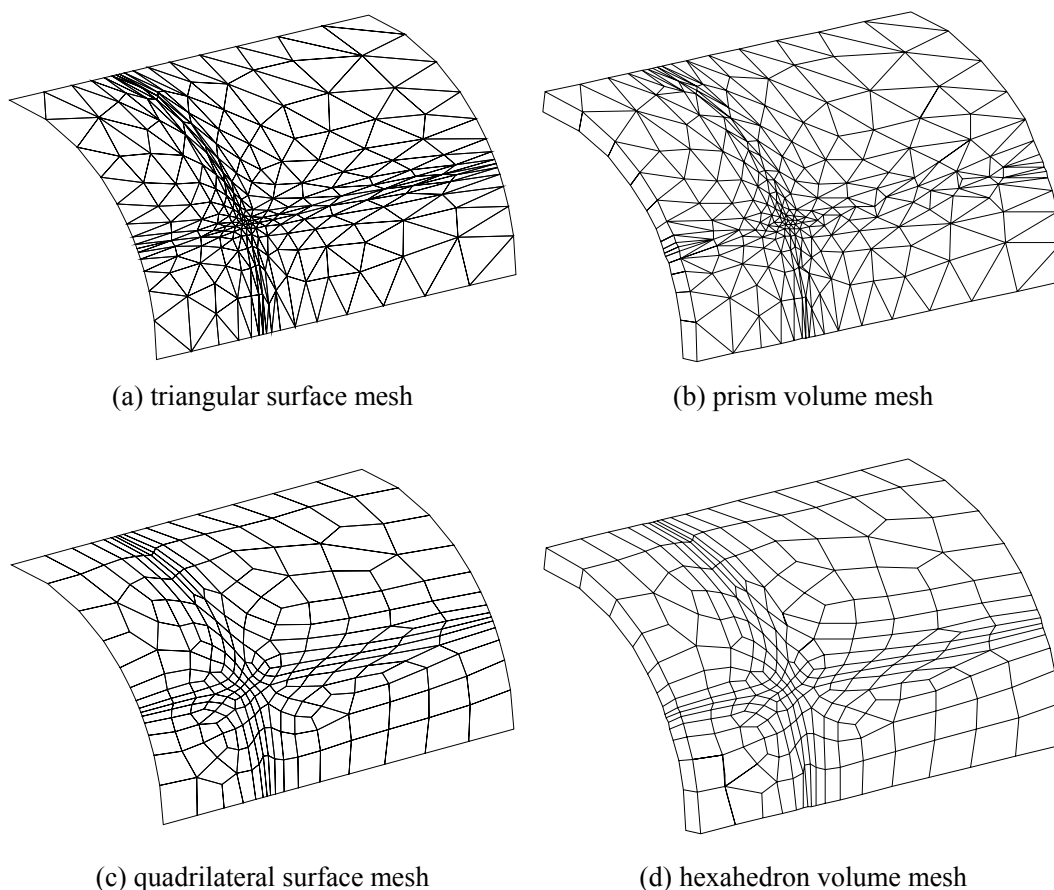


Figure 3.22 Meshes for Example 3.1

Example 3.1 is a single curved shell. The input surface model is highly anisotropic by the definition of metric tensors [3]. Figure 3.22a and Figure 3.22b

Surface mesh extrusion

show the input triangular surface mesh and the extruded prism volume mesh. If the surface mesh is given by an unstructured quadrilateral mesh (Figure 3.22c), the corresponding volume mesh extruded will be the hexahedron mesh shown in Figure 3.22d. Hence this example shows that the surface extrusion algorithm can be successfully applied to either triangular surface mesh or quadrilateral surface mesh. Furthermore, it can even be applied to a mixed surface mesh constructed by both triangular elements and quadrilateral elements.

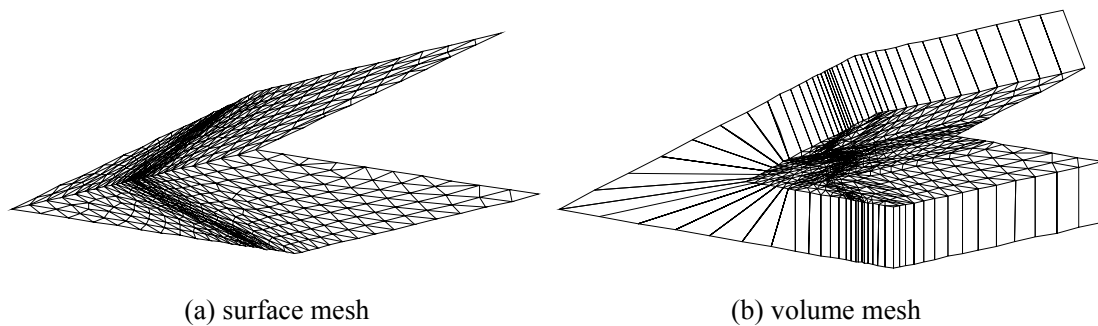


Figure 3.23 Meshes for Example 3.2

Example 3.2 is a model constructed by intersecting two thick planar surfaces. The input surface mesh and the extruded volume mesh are shown in Figure 3.23a and Figure 3.23b respectively. Note that, in this example, the input surface model is highly anisotropic with small surface intersection angle and large thickness of the surfaces (when compared with the element size). In such an extreme case, in order to obtain a valid solid mesh, it is necessary to adjust the normal side directions of the input surface mesh (as shown in Figure 3.23b) in order to avoid the formation of invalid elements. As a result, the direction of the input adapted trend line is modified locally near the intersection. Note that such situations do not normally occur in the usual applications of the purposed mesh generator and the main purpose of showing this example is to demonstrate the robustness of the algorithm.

Surface mesh extrusion

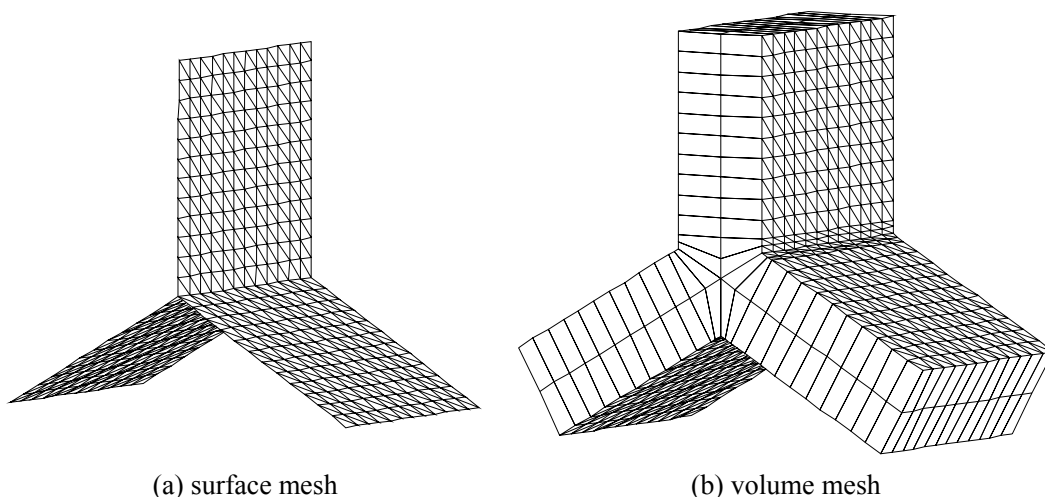


Figure 3.24 Meshes for Example 3.3

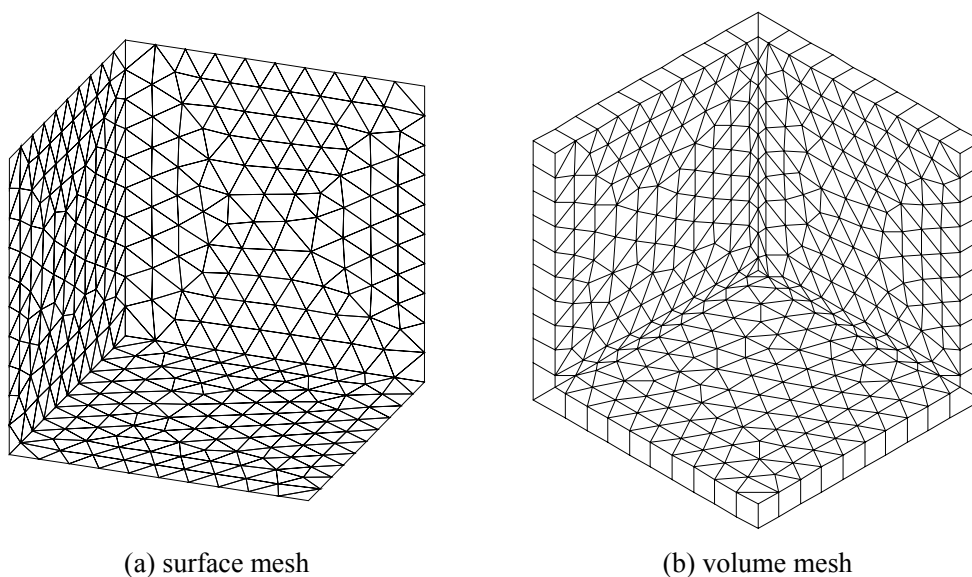


Figure 3.25 Meshes for Example 3.4

Example 3.3 and Example 3.4 are structures constructed by multi planar surfaces. Example 3.3 is a multi-cross structure and Example 3.4 is a structure with corner point. The input surface mesh and output volume mesh for Example 3.3 are shown in Figure 3.24a and Figure 3.24b respectively, while those meshes for Example 3.4 are given in Figure 3.25a and Figure 3.25b. Note that Example 3.3 consists of a multi-surface intersection (Figure 3.7b), so every surface has been divided into two parts and each part is formed separately.

Surface mesh extrusion

It should be remarked that for Examples 3.2 to 3.4, due to the normal side adjustment procedures (Sections 3.3.1.1, 3.3.1.2) for the treatment of surface intersections and corner points, the grading of the elements near the surface intersection and corner points are modified. In general, due to the use of such adjustment algorithms, the sizes of elements along the surface junctions and corner points are often reduced. Despite this would modify the mesh density locally, it is a desirable outcome as in most cases of structural analyses, denser elements are required along such structural junctions.

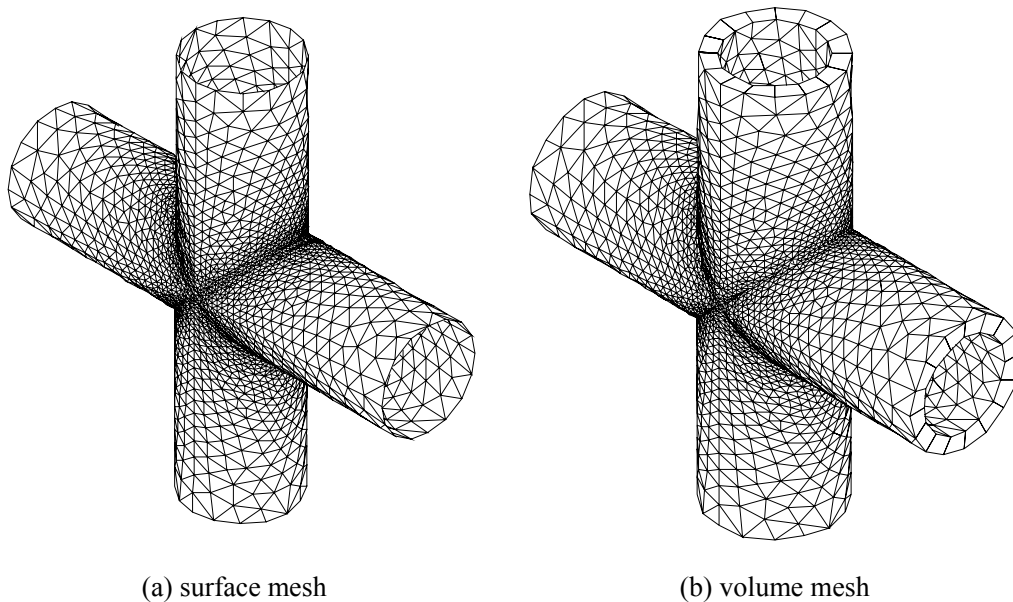
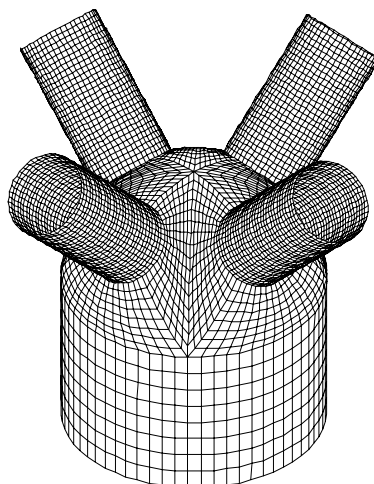


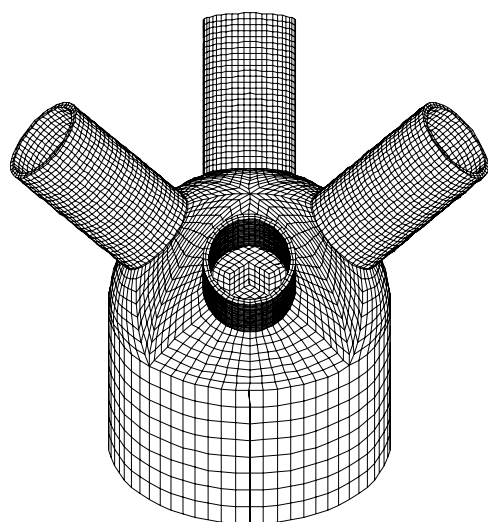
Figure 3.26 Meshes for Example 3.5

In Example 3.5, the model employed is formed by the intersection of two equal radius tubes and is a typical tubular structure (X-joint) used in the construction of offshore platforms. Smaller elements are used near the junction. The input graded surface mesh used and the volume mesh generated are shown in Figure 3.26a and Figure 3.26b. This example can be regarded as a complex case of the structures with corner points.

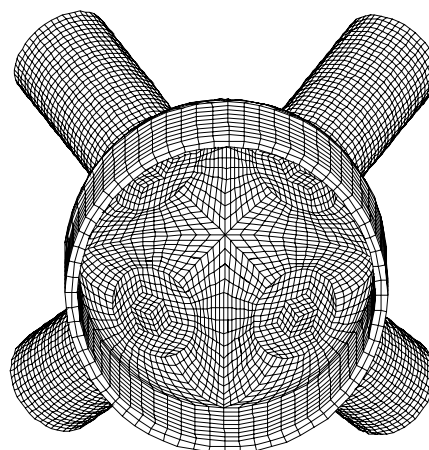
Surface mesh extrusion



(a) surface mesh



(b) volume mesh, top view



(c) volume mesh, bottom view

Figure 3.27 Meshes for Example 3.6

In the last example, the structure under consideration is formed by the intersection of a large structural steel spherical head with four smaller braces as shown in Figure 3.27a. It can be regarded as a complex case of multi-cross structures. In this example, the input mesh is a structured quadrilateral mesh obtained by the mapping technique. The volume mesh generated is shown in Figure 3.27b and Figure 3.27c in different direction of views. This example also shows that the surface mesh extrusion algorithm is not only suitable for the surface mesh generated by the program used in this study, but also can be applied to the mesh

generated by other surface mesh generators.

3.5 Closure

In this chapter, the algorithms for extruding the surface (both single surface and multi-surface) are given. According to the characteristics of the intersection involved, structures with multi-surface are divided into single surface intersection, multi-cross intersection and corner point intersection. Treatments for all these types of structures are designed. After the extrusion step, the surface mesh is converted to a volume mesh. If the input mesh is a triangular surface mesh, the output volume mesh will be the prism mesh. If the input mesh is quadrilateral mesh, then the output volume mesh will be the hexahedron mesh. For an input mesh with both triangular elements and quadrilateral elements, the algorithms presented in this chapter can also be applied and the extruded volume mesh will consist of prism elements and hexahedron elements. Examples show that the presented mesh extrusion scheme is valid and robust not only for simple TWSs but also for those complicated structures such as the tubular joints. It must be pointed out that surface extrusion is only the first step in the conversion from surface mesh to solid mesh. The extruded volume mesh is either a single-layered or double-layered mesh. In next chapter, other algorithms will be designed to convert the volume mesh into a 3D solid mesh which is suitable for FE analysis.

CHAPTER 4

SOLID MESH GENERATION

In Chapter 3, a new algorithm to extrude the surface mesh to volume mesh has been presented. After the extrusion procedure, the surface mesh is converted to a volume mesh which consists of one layer (for single surface and corner point structures) or two layers (for multi-cross structures). In practice, the volume mesh will be refined along the thickness direction to ensure the accuracy of the FE model. The final refined mesh is referred as the solid mesh. In this chapter, algorithms for the mesh conversion from volume to solid mesh work will be designed, which consist of two parts

- 1) *Volume mesh refinement*
- 2) *Solid mesh smoothing*

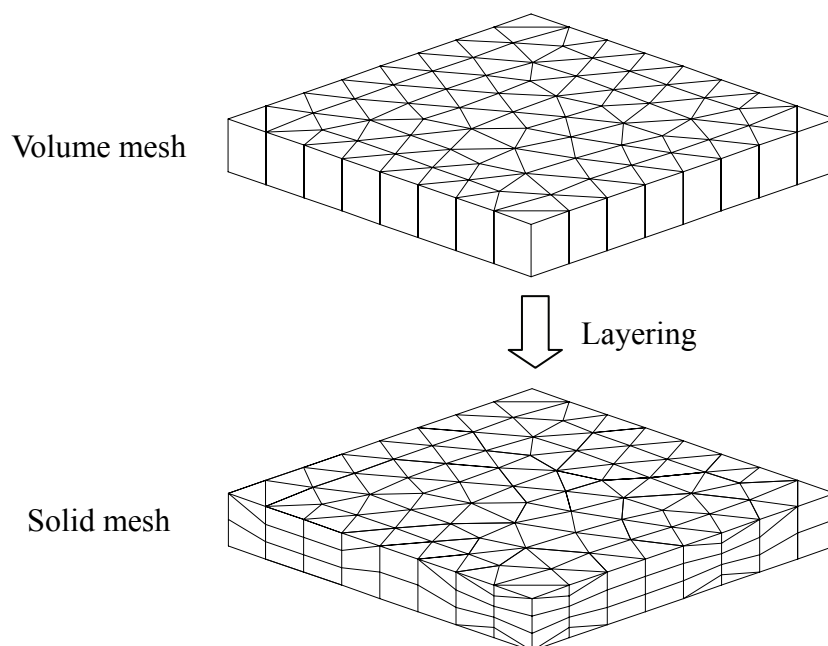


Figure 4.1 Layering the volume mesh

4.1 Volume Mesh Refinement

Volume mesh refinement is the step to refine the extruded mesh by laying the volume elements. This process is illustrated in Figure 4.1. Note that the middle facet of a volume element (Figure 3.3) does not divide the volume element into two layers: It is only referred as the mid-surface definition of the underlying TWS.

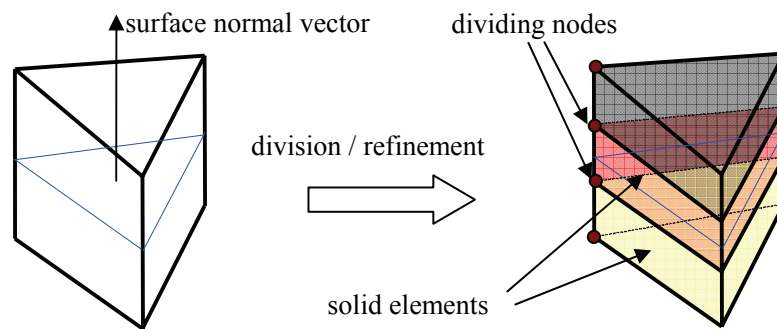


Figure 4.2 Division of a volume element, $NEL = 3$.

4.1.1 Refinement parameters used

The volume mesh refinement process is controlled by the input data NEL which define the number of layers of elements required in the thickness direction for all the nodal points of the surface mesh. In case that NEL is constant for all elements, the division process is trivial and involving uniform division of the volume elements into NEL solid elements as shown in Figure 4.2. However, in order to optimize the cost effectiveness of the FE analysis, values of NEL may vary along the input surface mesh and it is often required to divide the volume mesh into different numbers of layers of elements at different locations of the structure. Thus, a refinement scheme to maintain the compatibility of the edge and faces connectivity for the final solid mesh is needed. Towards this end, it is found that the complexity of refinement scheme

Solid mesh generation

increases as the differences in the NEL values among adjacent normal sides increase. Hence, for practical consideration, it is restricted that the maximum difference in the NEL values among adjacent elements cannot be greater than two. As mentioned in Section 3.2, NEL is treated as one of the input data, it is possible that the input NEL values may not satisfy the imposed restriction. Hence, an advancing front correction procedure described in Box 4.1 is employed to check and rectify the input NEL values. This procedure is illustrated in Figure 4.3.

- (i) Initialize the whole surface mesh to the *current region*.
- (ii) Within the *current region*, search for the normal sides that have the highest NEL value and name this NEL value as NELF. Quit if $NELF \leq 3$.
- (iii) Group those normal sides with NELF layers to *frontal region* and then exclude the *frontal region* from the *current region*.
- (iv) In the redefined *current region*, search for the *neighboring normal sides* of the *frontal region* and group them to the *frontal neighbor*.
- (v) Within the *frontal neighbor*, if the NEL value of any normal side is smaller than (NELF -2), change it to (NELF -2).
- (vi) Goto step (ii).

Box 4.1 Check and rectify the input NEL values

In Figure 4.3, the meaning of the colors of the normal sides is explained as following:

- Ⓝ : unchecked normal sides
- Ⓝ : normal sides in the *frontal region*
- Ⓝ : normal sides in the *frontal neighbor*
- Ⓝ : rectified normal sides

Solid mesh generation

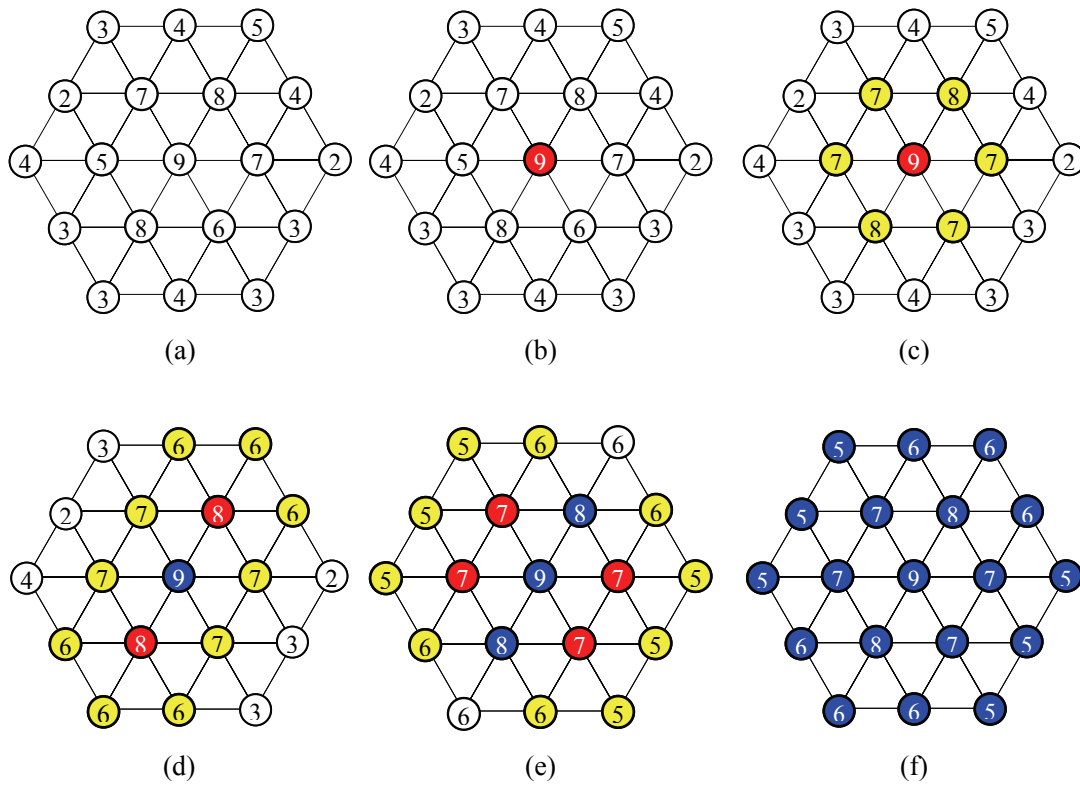


Figure 4.3 Rectification of input NEL values: (a) input NEL values; (b) modification starts at the normal sides with highest NEL values; (c) after first layer is modified; (d) after two layers are modified; (e) after three layers are modified; and (f) final NEL values

4.1.2 Refinement procedures along the thickness direction

After the values of NEL are checked and rectified, refinement will be carried out in the thickness direction to generate 3D solid elements. The refinement scheme used in this study consists of two distinct steps, namely, (i) the connection of dividing nodes for adjacent normal sides and (ii) the formation of 3D solid elements.

4.1.2.1 Connection of dividing nodes for adjacent normal sides

For a given normal side V_A , once NEL_A is determined, NEL_A+1 dividing nodes will be created (Figure 4.2). However, since NEL may vary from one normal side to the other, a connection scheme is needed to ensure the compatibility of the

Solid mesh generation

edges constructed when adjacent normal sides (normal sides that share a common vertical face) are connected. The connection edges generated will also divide the vertical faces of the volume elements in the thickness direction. Consider two adjacent normal sides V_A and V_B with NEL values equal to NEL_A and NEL_B respectively such that $NEL_A \geq NEL_B$. The scheme employed to connect V_A with V_B will depend on the parities (odd and even) of NEL_A and NEL_B and their difference, $NEL_A - NEL_B$. Due to the constraint imposed on the values of NEL in Section 4.1.1, it can be obtained that $NEL_A - NEL_B \leq 2$ and therefore, only five distinct cases are possible.

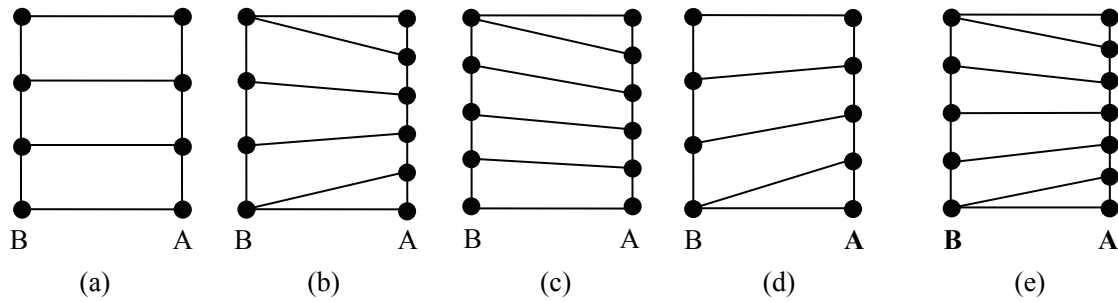


Figure 4.4 Connection of adjacent normal sides: (a) case of $NEL_A = NEL_B = 3$;
 (b) case of $NEL_A = 5 = NEL_B + 2$; (c) case of $NEL_A = 5 = NEL_B + 1$;
 (d) case of $NEL_A = 4 = NEL_B + 1$; and (e) case of $NEL_A = 6 = NEL_B + 2$

- (1) $NEL_A = NEL_B$: In this case, dividing nodes on V_A and V_B are connected in a simple one-to-one correspondence manner as shown in Figure 4.4a with $NEL_A = 3$.
- (2) $NEL_A > NEL_B$ and both NEL_A and NEL_B are odd numbers: In this case, it can be deduced that $NEL_A - NEL_B = 2$ and connection will start from the lowest dividing node of V_B to the second lowest dividing node of V_A . Connection then continues from the second lowest dividing node of V_B to the third lowest dividing node of V_A and so on. Construction will be ended when the uppermost dividing node of V_B is connected as shown in Figure 4.4b where $NEL_A = 5$.

Solid mesh generation

- (3) $NEL_A > NEL_B$ and NEL_A is an even number while NEL_B is an odd number: In this case, it can be deduced that $NEL_A - NEL_B = 1$ and connection will start from the second lowest dividing node of V_B to the second lowest dividing node of V_A . Connection then continues from the third lowest dividing node of V_B to the third lowest dividing node of V_A and so on. Construction will be ended when the uppermost dividing node of V_B is connected as shown in Figure 4.4c where $NEL_A = 5$.
- (4) $NEL_A > NEL_B$ and NEL_A is an odd number while NEL_B is an even number: In this case, once again, one has $NEL_A - NEL_B = 1$ and connection is done by reversing the connection sequence for the case (3) as shown in Figure 4.4d where $NEL_A = 4$.
- (5) $NEL_A > NEL_B$ and both NEL_A and NEL_B are even numbers: In this case, one has $NEL_A - NEL_B = 2$ and exactly the same connection sequence for the case (2) will be used as shown in Figure 4.4e where $NEL_A = 6$.

4.1.2.2 Formation of 3D solid elements

After all the dividing nodes for adjacent normal sides are constructed and vertical faces of the volume elements are divided, 3D elements will be formed by dividing the volume elements in the thickness direction. Depends on the shape (prism or hexahedron) of the volume elements, two different 3D element formation schemes are developed.

1) 3D element formation scheme for prism volume elements

For prism volume elements which are generated by extrusion of triangular mid-surface elements, the 3D solid element formation scheme used is relatively simple. 3D solid elements will be defined naturally by collecting edges and faces generated during the dividing nodes connection step and no further splitting of the

Solid mesh generation

volume elements is needed. An example to demonstrate the procedure used for an assembly of prism volume elements is given in Figure 4.5. As shown in Figure 4.5, three different types of 3D solid elements, namely, (1) 4-node tetrahedron, (2) 5-node pyramid [102] and (3) 6-node prism are generated. The numbers of different types of generated 3D solid elements in Figure 4.5b are listed in Table 4.1.

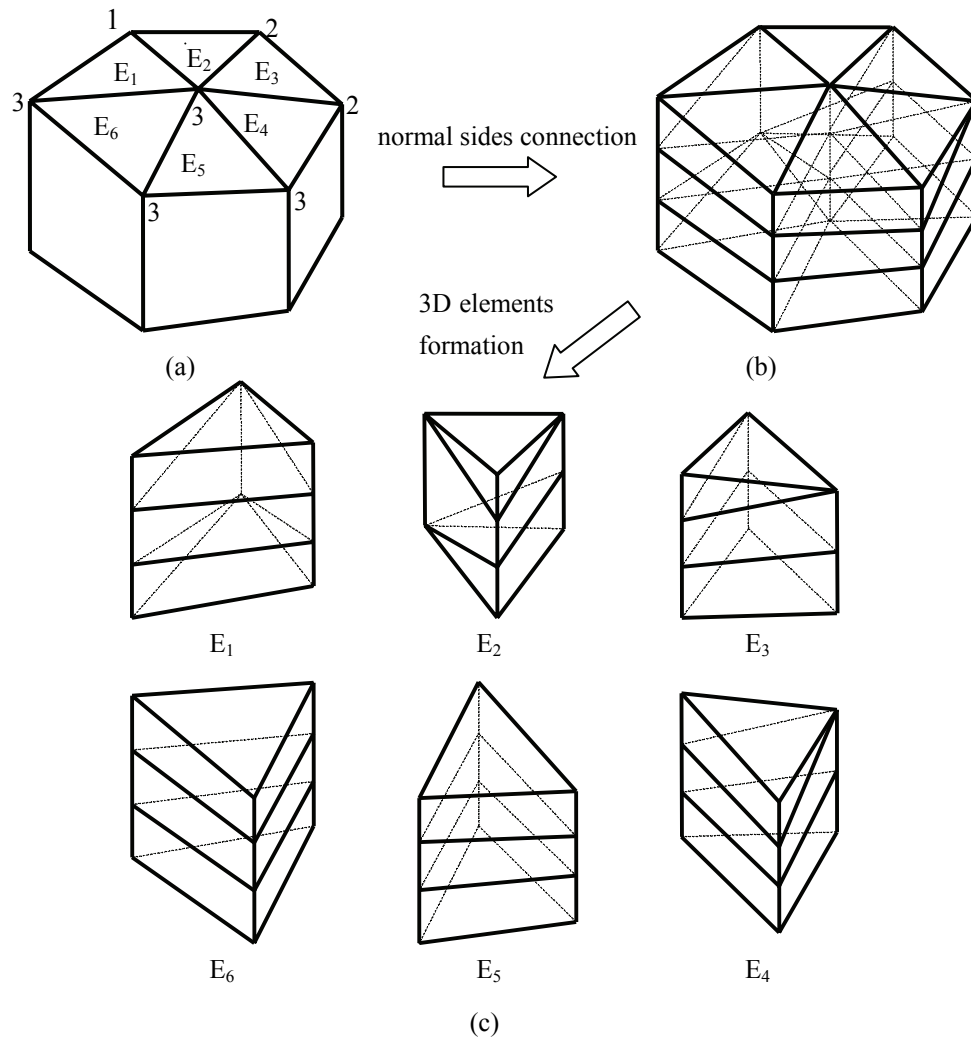


Figure 4.5 Formation of 3D solid elements from an assembly of prism volume elements: (a) an assembly of prism volume elements and NEL at normal sides; (b) connections of normal sides; and (c) formation of 3D solid elements

	Tetrahedron	Pyramid	Prism
Number of elements	2	4	12

Table 4.1 Number of different types of solid elements in Figure 4.5b

Solid mesh generation

2) 3D element formation scheme for hexahedral volume element

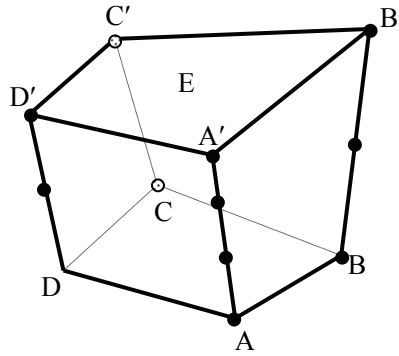


Figure 4.6 A hexahedral volume element E such that $NEL_{\max}=3$, $NELD=3$ and $NMS=1$

For a hexahedron volume element extruded from a quadrilateral mid-surface element, the element formation process will be more complicated and in some cases it is necessary to split a hexahedral volume element into two prism elements in order to maintain the compatibility of the resulted 3D mesh. Consider a hexahedral volume element E shown in Figure 4.6 such that the number of divisions at its four normal sides, V_η , $\eta = A, B, C$ and D are equal to NEL_η , $\eta = A, B, C$ and D respectively. From the constraint imposed on the values of NEL , it can be deduced that not all NEL_η are different. That is, if one defines $NELD$ as the number of normal sides that have different NEL values, then $NELD$ will satisfy the condition

$$1 \leq NELD_E \leq 3 \quad (4.1)$$

In addition, if one defines $NEL_{\max} = \max(NEL_A, NEL_B, NEL_C, NEL_D)$ and NMS as the number of normal sides with NEL values equal to $NEL_{\max}$, then the range of NMS will satisfy the following conditions:

$$\begin{aligned} NELD = 1 &\Rightarrow NMS = 4 \\ NELD = 2 &\Rightarrow 1 \leq NMS \leq 3 \\ NELD = 3 &\Rightarrow 1 \leq NMS \leq 2 \end{aligned} \quad (4.2)$$

Furthermore, without loss in generality, one can always assume that $NEL_A = NEL_{\max}$. In case that when $NMS > 1$, the normal side V_A can always be selected in such a way that its node number is the smallest among all the nodes with $NEL = NEL_{\max}$. By

using the definitions of NELD and NMS and Equations (4.1) and (4.2), the element formation scheme for the hexahedral volume element E can be summarized as follow.

- (1) $NELD = 1$ and $NMS = 4$: In this case, dividing nodes on all normal sides will be connected in a simple one-to-one correspondence manner.
- (2) $NELD = 2$ and $NMS = 2$: In this case, no splitting is needed and 3D solid elements can be formed directly by collecting edges and faces generated during the dividing nodes connection step.
- (3) $NELD = 2$, $NMS = 1$ or 3 and $NELD = 3$, $NMS = 1$ or 2 : In these cases, it is necessary to split the hexahedral volume element into two prism volume elements along the diagonal faces formed by V_A and its opposite normal side V_C (Figure 4.7). After the two prism volume elements are formed, dividing nodes on V_A and V_C are then connected along the diagonal faces. Finally, 3D solid elements will be formed by collecting edges and faces generated during the dividing nodes connection step.

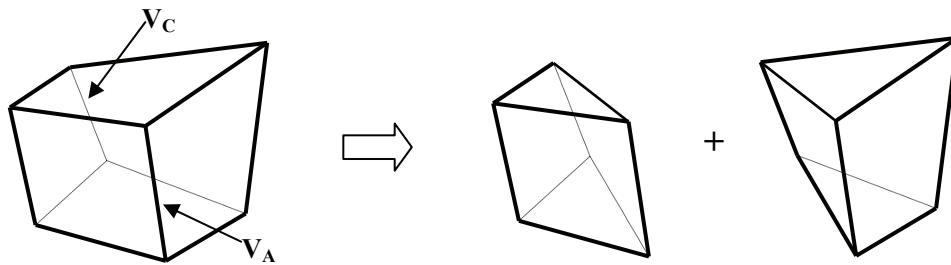


Figure 4.7 Splitting of a hexahedral volume element into two prism volume elements

In order to demonstrate the refinement procedure used, four set of examples showing how the elements are formed are given in Figure 4.8 to Figure 4.11 for cases when $NELD > 1$ and splitting is needed. From Figure 4.8 to Figure 4.11, it can be seen that whenever hexahedral volume elements are involved, in addition to the tetrahedron, prism and pyramid elements, 8-node hexahedral elements will also

Solid mesh generation

be generated. The numbers of different types of generated solid elements for the examples shown in Figure 4.8 to Figure 4.11 are listed in Table 4.2 to Table 4.5, respectively.

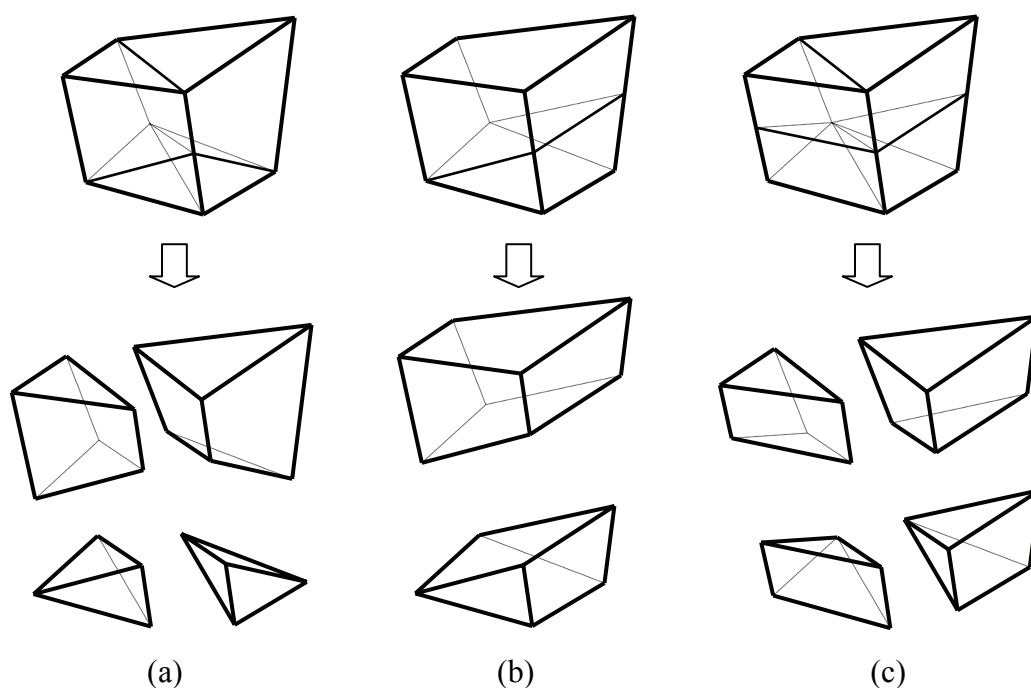


Figure 4.8 Examples (set 1) for refinement of hexahedral volume element when $NELD=2$: (a) $NMS=1(2-1-1-1)$; (b) $NMS=2(2-2-1-1)$; and (c) $NMS=3(2-2-1-2)$ (Numbers inside brackets are values of $NEL_A-NEL_B-NEL_C-NEL_D$ respectively)

	Tetrahedron	Pyramid	Prism	Hexahedron
Figure 4.8a	2	0	2	0
Figure 4.8b	0	0	1	1
Figure 4.8c	0	2	2	0

Table 4.2 Number of different types of solid elements in Figure 4.8

Solid mesh generation

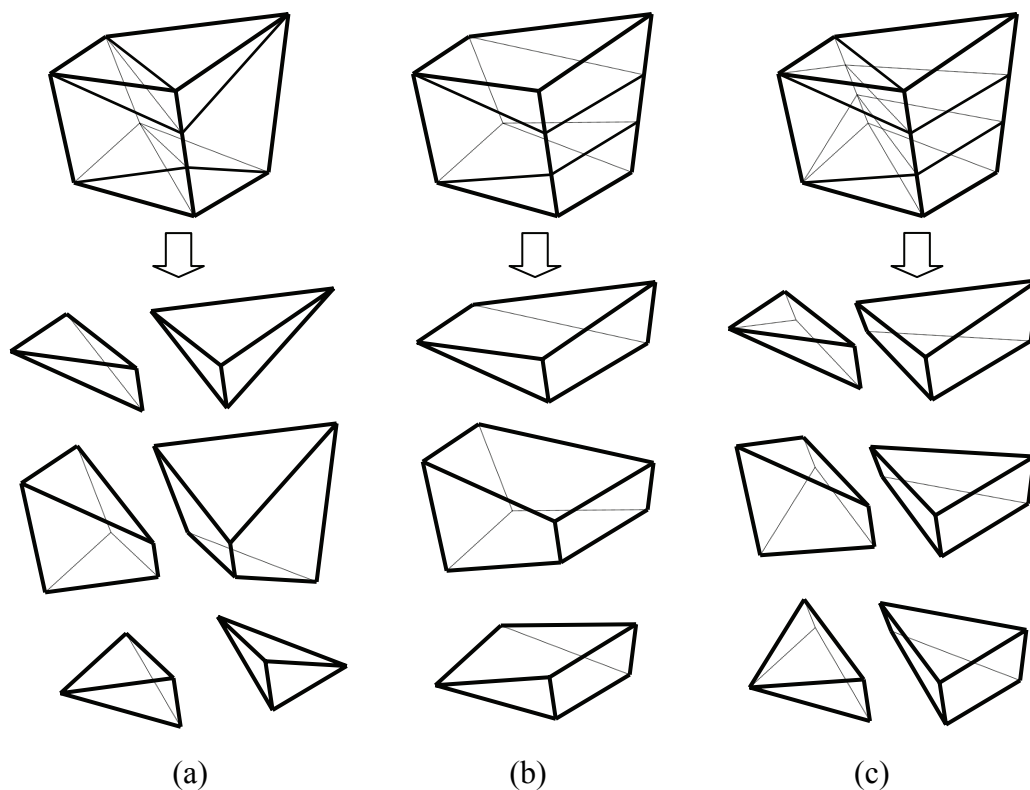


Figure 4.9 Examples (set 2) for refinement of hexahedral volume element when $NELD=2$: (a) $NMS=1$ (3-1-1-1); (b) $NMS=2$ (3-3-1-1); and (c) $NMS=3$ (3-3-3-1) (Numbers inside brackets are values of NEL_A - NEL_B - NEL_C - NEL_D respectively)

	Tetrahedron	Pyramid	Prism	Hexahedron
Figure 4.9a	4	0	2	0
Figure 4.9b	0	0	2	1
Figure 4.9c	0	2	4	0

Table 4.3 Number of different types of solid elements in Figure 4.9

Solid mesh generation

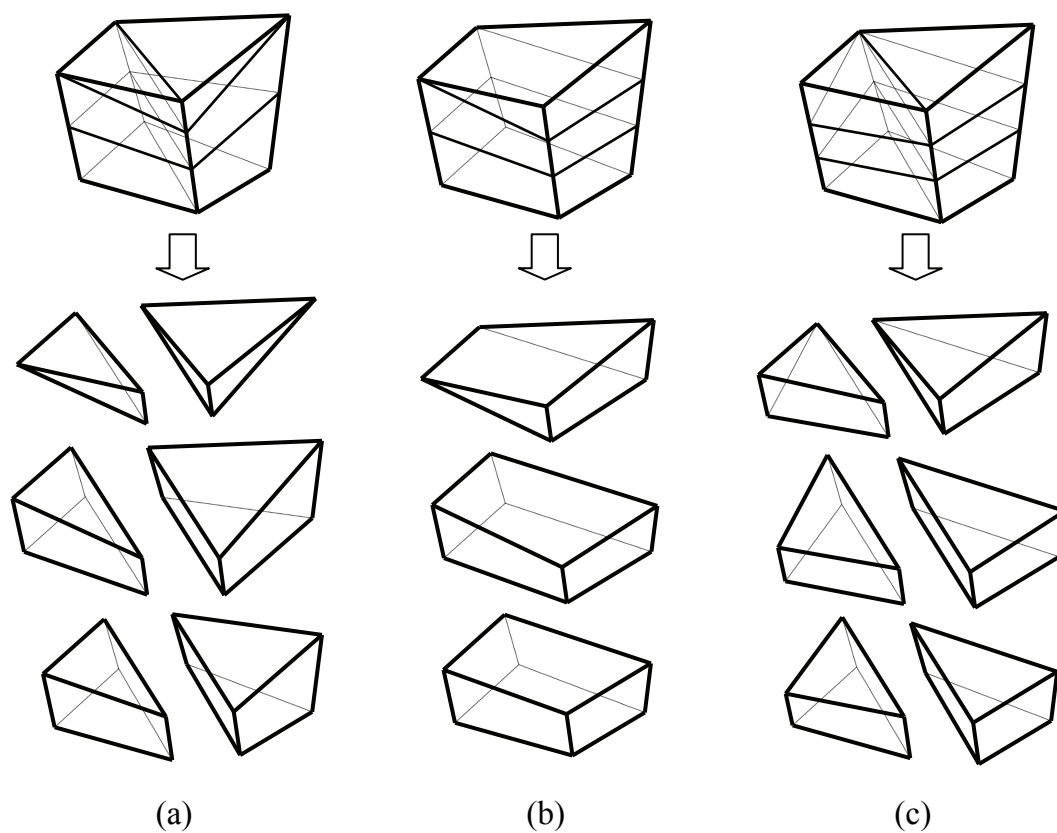


Figure 4.10 Examples (set 3) for refinement of hexahedral volume element when NELD=2: (a) NMS=1 (3-3-3-2); (b) NMS=2 (3-3-2-2); and (c) NMS=3 (3-3-2-3) (Numbers inside brackets are values of NEL_A - NEL_B - NEL_C - NEL_D respectively)

	Tetrahedron	Pyramid	Prism	Hexahedron
Figure 4.10a	2	0	4	0
Figure 4.10b	0	0	1	2
Figure 4.10c	0	2	4	0

Table 4.4 Number of different types of solid elements in Figure 4.10

Solid mesh generation

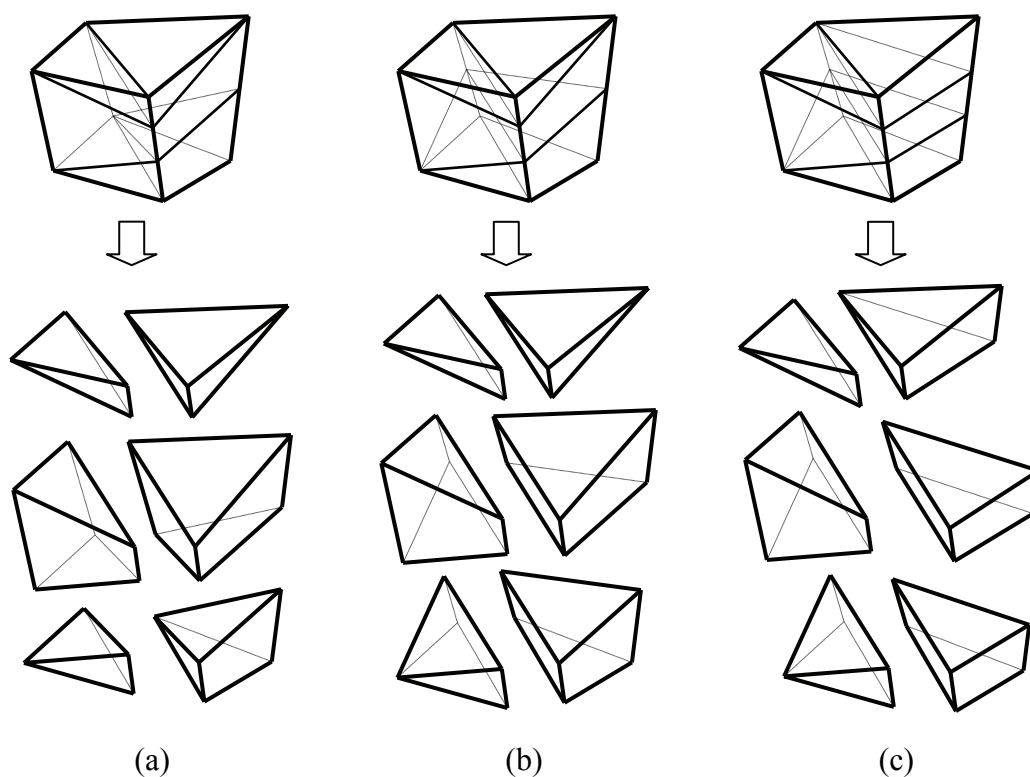


Figure 4.11 Examples (set 3) for refinement of hexahedral volume element when $NEL_D=3$: (a) $NMS=1$ (3-2-1-1); (b) $NMS=1$ (3-2-2-1); and (c) $NMS=2$ (3-3-2-1) (Numbers inside brackets are values of NEL_A - NEL_B - NEL_C - NEL_D respectively)

	Tetrahedron	Pyramid	Prism	Hexahedron
Figure 4.11a	3	1	2	0
Figure 4.11b	2	1	3	0
Figure 4.11c	1	2	3	0

Table 4.5 Number of different types of solid elements in Figure 4.11

4.2 Mesh Quality Measurement and Enhancement

Generally, the solid mesh generated by above refining scheme is not optimized in element shape qualities. In order to get a better performance in the FE analysis procedure, the solid mesh quality needs to be measured and enhanced. Considering that in the current generation procedures, volume elements will be formed before they are further converted to solid elements, the mesh quality enhancement process can be applied in two separate steps. The first step, *normal sides smoothing* will be applied to the extruded volume mesh while in the second step, *dividing nodes smoothing* will be applied to the refined 3D solid mesh. Before the introduction of schemes for these two steps, some reviews will be presented for the mesh measurement technique first.

4.2.1 Some reviews on mesh measurement

Triangle and quadrilateral are the basic elements in 2D mesh and surface mesh, while tetrahedron and hexahedron are the main elements used in 3D mesh. The emphasis of mesh quality measurement is to design measures to describe the shape qualities of these elements.

A fair measure of elemental shape should have four attributes, which are listed as following, in the order of importance.

- (1) An ability to detect all degenerate elements
- (2) Non-dimensionality
- (3) Boundedness
- (4) Normalization

The ability to detect all degenerate elements means that the measure must yield a

Solid mesh generation

value of zero for two (three)-dimensional elements that have no volume. The absence of dimensions means that similar triangles (tetrahedrons) or quadrilaterals (hexahedrons) should yield the same dimensionless measurement. Boundedness means that the measure cannot yield an arbitrarily large value. Normalization allows better comparisons among measures by forcing measures to take on positive values between zero and one.

There are many fair triangle measures. Among them, the measure proposed by Bhatia and Lawrence [103] is the most popular one. This measure is defined as

$$\alpha = \frac{4\sqrt{3}A}{l_1^2 + l_2^2 + l_3^2} \quad (4.3)$$

where A is the area of triangle and l_i ($i = 1, 2, 3$) is the length of side. Obviously, an equilateral triangle has the best shape quality value of α , which is equal to 1. With respect to the *metric tensor* [11], for the triangle shown in Figure 4.12a, α is given by

$$\alpha = \min(\alpha(\mathbf{P}_1), \alpha(\mathbf{P}_2), \alpha(\mathbf{P}_3)) \quad (4.4)$$

where

$$\alpha(\mathbf{P}_i) = \frac{2\sqrt{3} \cdot \text{Det}(\mathbf{M}_i) \cdot \text{Det}([\mathbf{P}_2 - \mathbf{P}_1, \mathbf{P}_3 - \mathbf{P}_1])}{\tilde{l}(\mathbf{M}_i, \mathbf{P}_1\mathbf{P}_2)^2 + \tilde{l}(\mathbf{M}_i, \mathbf{P}_2\mathbf{P}_3)^2 + \tilde{l}(\mathbf{M}_i, \mathbf{P}_1\mathbf{P}_3)^2} \quad (4.5)$$

$\text{Det}(\mathbf{M})$ means the determinant of the metric tensor $\mathbf{M}$ (Equation (3.7)) and $\tilde{l}(\mathbf{M}, \mathbf{P}_i\mathbf{P}_j)$ is the distance between two points $\mathbf{P}_i$ and $\mathbf{P}_j$ using metric tensor specification (Equation (3.8)).

For quadrilateral elements, Lo [104] used β as the measure.

$$\beta = \frac{\alpha_3 \cdot \alpha_4}{\alpha_1 \cdot \alpha_2} \quad (4.6)$$

where α_i ($i=1, 2, 3, 4$) is the measure obtained from Equation (4.4) which

Solid mesh generation

corresponds to the four triangles formed by using the two diagonals P_1P_3 and P_2P_4 (Figure 4.12b). The four values of α are arranged in the order:

$$\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \alpha_4 \quad (4.7)$$

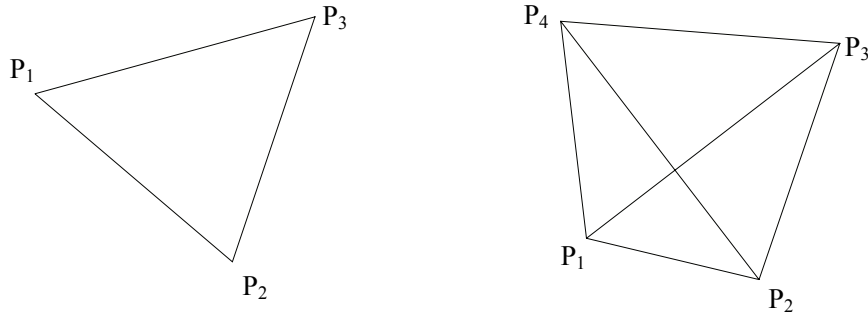


Figure 4.12 Measures of triangle and quadrilateral

It must be pointed out that the α and β values defined in Equation (4.4) and Equation (4.6) only account for the shape distortion effect but not the edge length of the element. An equilateral triangle of any size will always attain a maximum α value of 1.0. In view of this, a combined shape and size parameter, μ , will be used to assess the overall quality of the element.

$$\mu = \begin{cases} \alpha(\delta_1\delta_2\delta_3)^{\frac{1}{3}} & \text{for triangles} \\ \beta(\delta_1\delta_2\delta_3\delta_4)^{\frac{1}{4}} & \text{for quadrilaterals} \end{cases} \quad (4.8)$$

where δ_i is the length deviation of the i th edge from unity and it is defined as

$$\delta_i = \min(\tilde{l}_i, \frac{1}{\tilde{l}_i}) \quad (4.9)$$

where $\tilde{l}_i$ is the length of the i th edge of the element. More details on metric surface mesh quality measurement are given in reference [105].

Measures for 3D elements are more complicated than those for 2D elements because not only the qualities of the faces, but also the qualities of the volumes, must be considered. For tetrahedron element, a simply and extremely popular measure is

defined as

$$q = \frac{3r}{R} \quad (4.10)$$

where r is the *inradius* and R is the *circumradius*. For hexahedron element, Robinson [106] used 8 parameters to define the measures. Related details can be found in reference [106, 107].

4.2.2 Volume mesh quality measurement and normal sides smoothing

Since volume elements are formed by extruding mid-surface elements along the optimal normal direction (Figure 3.2), when measuring the shape quality of a volume element one should therefore consider both the *facet shape qualities* of its three (top, middle, bottom) facets and the *normal direction qualities* of the normal sides of the element.

4.2.2.1 Facet shape quality measurement

For a given volume element, the parameter $\phi \in [0, 1]$ to measure its facet shape quality is defined as

$$\phi = (q_t \cdot q_m \cdot q_b)^{1/3} \quad (4.11)$$

where q_t , q_m and q_b are the shape quality measures for the top, middle and bottom facets of the volume element respectively. In here, the measures q_t , q_m and q_b are taken as the α and β qualities measures (Equation (4.4) and Equation (4.6)) for triangular and quadrilateral facets respectively. For a given normal side $\mathbf{V}$, the parameter $\Phi \in [0, 1]$ defined as

$$\Phi = \left(\prod_{i=1}^{NB} \phi_i \right)^{1/NB} \quad (4.12)$$

Solid mesh generation

will be used to measure the average facet shape quality of all the volume elements connected to it. In Equation (4.12), NB is the number of elements adjacent to the normal side and ϕ_i is the facet shape quality of the i th volume element.

4.2.2.2 Normal direction quality measurement

For a given normal side $\mathbf{V}$ with normalized normal direction $\mathbf{b}$ that connects to NB mid-surface elements, the parameter $\Psi \in [0, 1]$ defined as

$$\Psi = \frac{1}{\text{NB}} \sum_{i=1}^{\text{NB}} (\mathbf{b} \cdot \mathbf{n}_i) \quad (4.13)$$

will be used to measure the quality of the normal direction. In Equation (4.13), $\mathbf{n}_i$ is the normalized surface normal vector of the i th adjacent mid-surface element (Figure 3.3). Ψ will attain its maximum value of unity when the normal direction $\mathbf{b}$ coincides with all the normal surface vectors of the adjacent mid-surface elements.

4.2.2.3 Overall shape quality measure and normal side smoothing

With the definitions of Φ and Ψ , the overall shape quality measure, Θ , for a normal side is defined as

$$\Theta = \Phi^\lambda \cdot \Psi^{(1-\lambda)} \quad (4.14)$$

and is used during the normal side smoothing. In Equation (4.14), λ is a real number such that $0 \leq \lambda \leq 1$. If λ is close to unity, during the smoothing process, normal side direction will be mainly adjusted based on the facet shape quality of the volume elements. However, if $\lambda = 0$, smoothing will be entirely controlled by the normal direction quality. For the optimal value of λ , it is found that the shape qualities of the top and bottom facets are quite sensitive to the change in the directions of the normal sides and a higher value of λ will result in a more effective improvement of the shape quality of the volume mesh. In fact, it is found that a

Solid mesh generation

value of $\lambda > 0.5$ will often lead to better volume meshes. Further numerical tests show that a value of $\lambda = 0.8$ seems to be appropriate for most practical cases and thus is adopted in the current study.

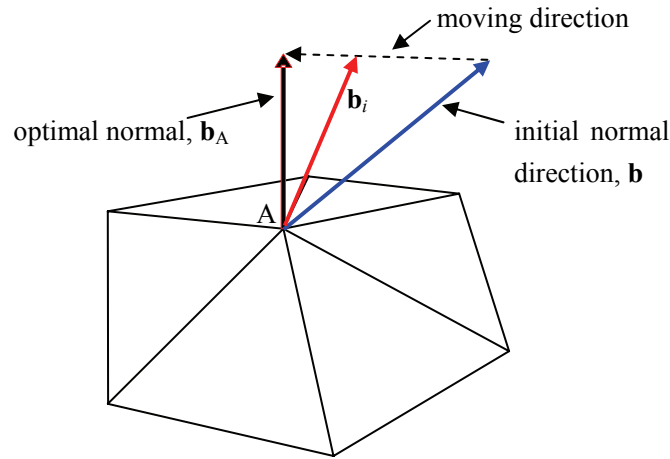


Figure 4.13 Normal side smoothing

For a given normal side corresponding to a node A with normal side direction $\mathbf{b}$ (Figure 4.13), the unit vector, $\mathbf{b}_A$ obtained by solving Equation (3.14) (or provided by the user) will be considered as the optimal normal direction. During the normal side smoothing procedure, a series of vectors defined as (c.f. Equation 3.18)

$$\mathbf{b}_i = \mathbf{b} \left(1 - \frac{i}{\text{ND}}\right) - \frac{i}{\text{ND}} \mathbf{b}_A, \text{ for } i = 1, \dots, \text{ND} \quad (4.15)$$

will be generated in the order from $i = 1$ to ND (Figure 4.12). The vector $\mathbf{b}_i$ will be accepted if it leads to an increase in the overall shape quality measure Θ . Similar to the case of normal side adjustment (Section 3.3.1.2), it is found that a value of ND=20 will lead to improvement in most practical cases.

For structures formed by only a single surface, the initial nodal normal, $\mathbf{b}$, is just the same as the optimal normal $\mathbf{b}_A$, hence there is no need to apply normal side smoothing for this kind of structures. Only for those structures constructed by intersected surfaces, normal side smoothing is required because some nodal normals

near the intersection have been shifted away from their optimal directions in the extrusion process (Section 3.3).

4.2.3 Dividing node smoothing

After 3D solid elements are formed, dividing node smoothing will be applied to adjust the positions of interior dividing nodes. Note that in order to maintain the consistency between the final mesh and the original geometry of the TWS, coordinates of dividing nodes which are lying on the top and bottom boundaries will not be altered. Definitions of the shape quality measures used for 3D solid elements will be introduced first before the description of the dividing node smoothing procedure.

4.2.3.1 Face shape quality measurement for 3D solid elements

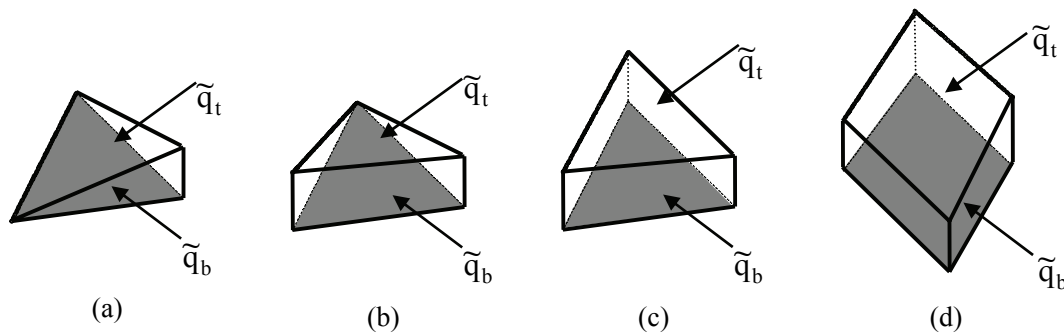


Figure 4.14 Face shape quality measure for 3D solid elements:
(a) tetrahedron; (b) pyramid; (c) prism; and (d) hexahedral

Similar to the facet shape quality measure for volume elements, the face shape quality measure for a solid element, $\tilde{\phi}$, can be defined as (c.f. Equation (4.11))

$$\tilde{\phi} = \sqrt{\tilde{q}_t \cdot \tilde{q}_b} \quad (4.16)$$

where $\tilde{q}_t$ and $\tilde{q}_b$ are the shape (α and β) qualities of the top and bottom faces of

Solid mesh generation

the 3D element respectively (Figure 4.14). For a node connects to NB elements, the parameter $\tilde{\Phi} \in [0, 1]$ defined as (c.f. Equation (4.12))

$$\tilde{\Phi} = \left(\prod_{i=1}^{NB} \tilde{\phi}_i \right)^{1/NB} \quad (4.17)$$

is employed to measure the average face shape quality for all the elements surrounding it.

4.2.3.2 Face normal vector quality measurement

Consider an interior dividing node Q for an assembly of 3D solid elements shown in Figure 4.15, the *face normal vector quality measure*, $\tilde{\Psi}$, is computed by using the following steps:

- (i) Identify the dividing nodes that are lying on the same normal side and one level below and above Q, denote them as L and U respectively. (Figure 4.15a)
- (ii) Identify all the 3D elements that contain both the nodes Q and L (shaded elements in Figure 4.15b) and collect them in a set denoted as $\mathcal{E}$.
- (iii) Retrieve the upper faces of elements in $\mathcal{E}$ and collect them in a set denoted as $\mathcal{F}$. Denote the number of faces in $\mathcal{F}$ as NF. The set $\mathcal{F}$ for Q is shown in Figure 4.15c, with NF = 6.
- (iv) Compute the unit normal vector of the faces retrieved and denote them as $\mathbf{f}_i$ for $i = 1, \dots, \text{NF}$.
- (v) Compute the face normal vector quality measure, $\tilde{\Psi} \in [0, 1]$ as (c.f. Equation (4.13))

$$\tilde{\Psi} = \frac{1}{2 \cdot \text{NF}} \sum_{i=1}^{\text{NF}} (\mathbf{u} \cdot \mathbf{f}_i + \mathbf{l} \cdot \mathbf{f}_i) \quad \text{with} \quad \mathbf{u} = \frac{\overrightarrow{\text{QU}}}{|\overrightarrow{\text{QU}}|} \text{ and } \mathbf{l} = \frac{\overrightarrow{\text{LQ}}}{|\overrightarrow{\text{LQ}}|} \quad (4.18)$$

Solid mesh generation

Similar to Ψ , $\tilde{\Psi}$ will attain unity when the normal vectors of all the faces adjacent to Q coincides with both $\mathbf{u}$ and $\mathbf{l}$.

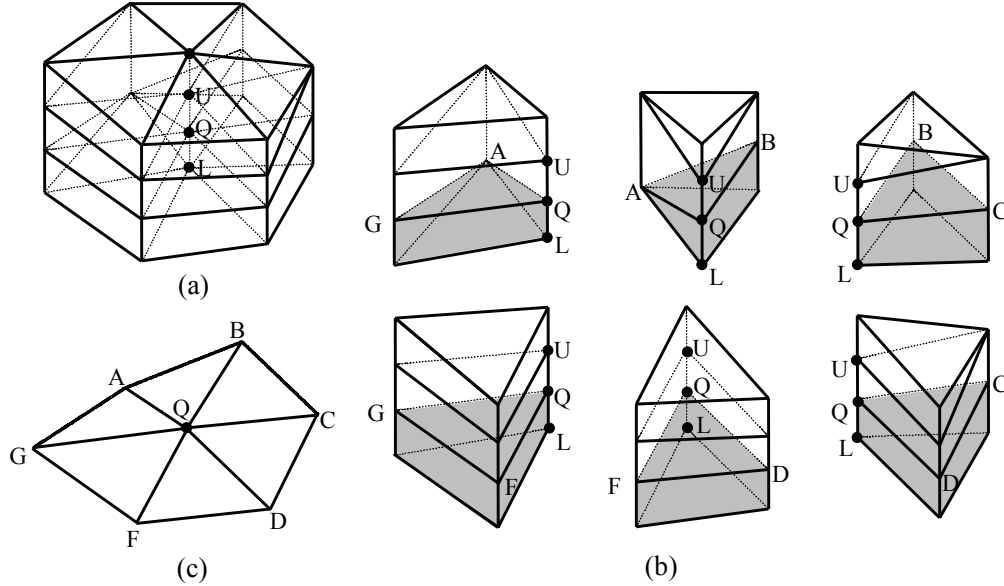


Figure 4.15 Face normal vector quality measure for interior node Q:
 (a) an assembly of 3D solid elements; (b) elements connectivity and
 (c) faces associated with interior node Q

4.2.3.3 Overall shape quality measure and dividing node smoothing

With the definitions of $\tilde{\Phi}$ and $\tilde{\Psi}$, the overall shape quality measure, $\tilde{\Theta}$, for dividing node smoothing is defined as (c.f. Equation (4.14))

$$\tilde{\Theta} = \tilde{\Phi}^\lambda \cdot \tilde{\Psi}^{(1-\lambda)} \quad (4.19)$$

Similar to the case of normal side smoothing, a value of $\lambda = 0.8$ is used in Equation (4.19). For a given dividing node Q, dividing node smoothing is started by identifying the face set $\mathcal{F}$ (Section 4.2.3.2, step (iii)) associated with it. New position of Q will be determined by applying the standard Laplacian smoothing method to the neighbourhood nodes in the set $\mathcal{F}$. The new position of a dividing node will be accepted only when the overall quality $\tilde{\Theta}$ after smoothing is greater than that before the movement. Numerical tests show that three cycles of smoothing can achieve a good smoothing effect.

4.3 Solid Mesh Generation Examples

In this section, nine examples will be presented to test the performance of the solid mesh generation algorithms developed in this chapter. The structures involved in Examples 4.1 to 4.3 are structures intersected by planar surfaces or simply curved surfaces. These three examples are used to demonstrate the effects of the processes of volume mesh refinement and solid mesh smoothing. From Example 4.4 to Example 4.8, the structures employed are more complicated structures which are constructed by two or more than two intersected surfaces. These examples are provided to show the wide applicability of the whole solid mesh generation scheme developed in Chapter 3 and Chapter 4 for realistic TWSs.

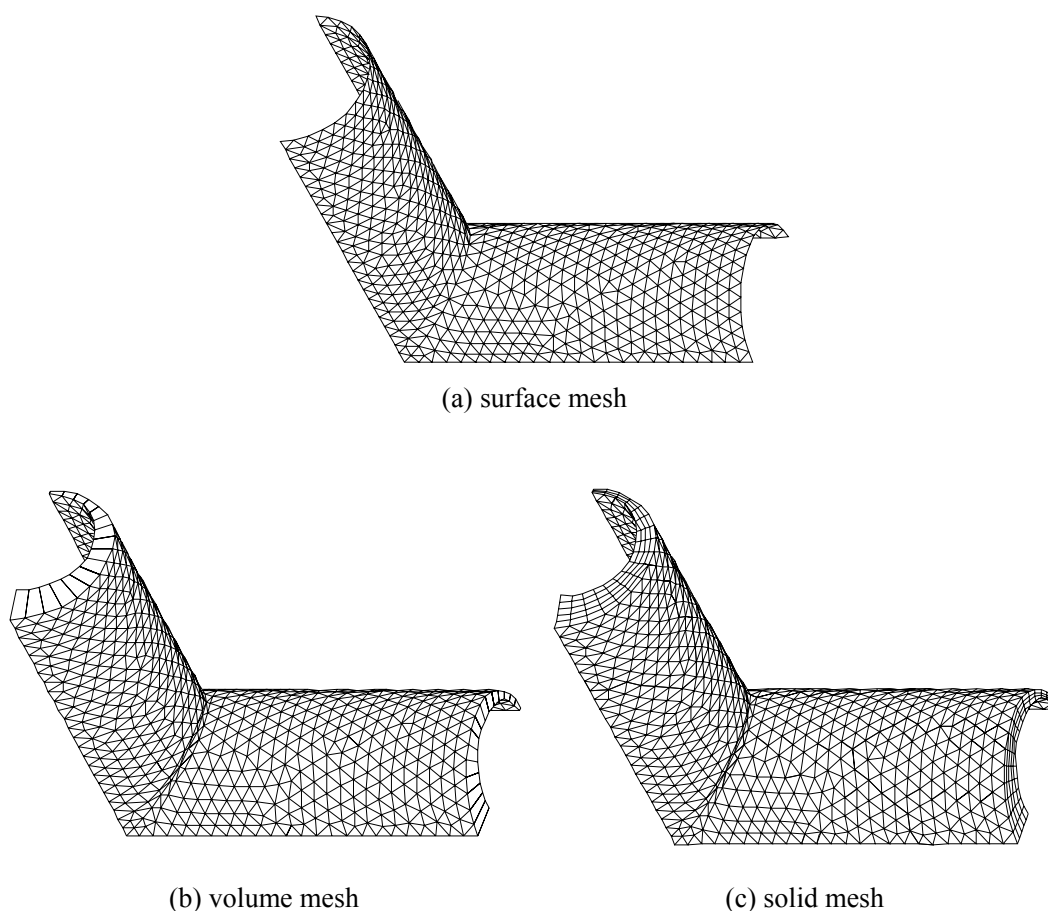


Figure 4.16 Meshes for Example 4.1

Solid mesh generation

Example 4.1 is a structure consisting of two cylindrical shells with non-uniform thickness. Figure 4.16a shows the input surface mesh and Figure 4.16b shows the extruded volume mesh. Since the thickness of structure is changed, different layers are used for different volume elements and more layers are used where the thickness is bigger. Smoothing process has not been applied to this example. The average Θ and $\tilde{\Theta}$ values of the generated solid mesh are 0.977414 and 0.879047, respectively.

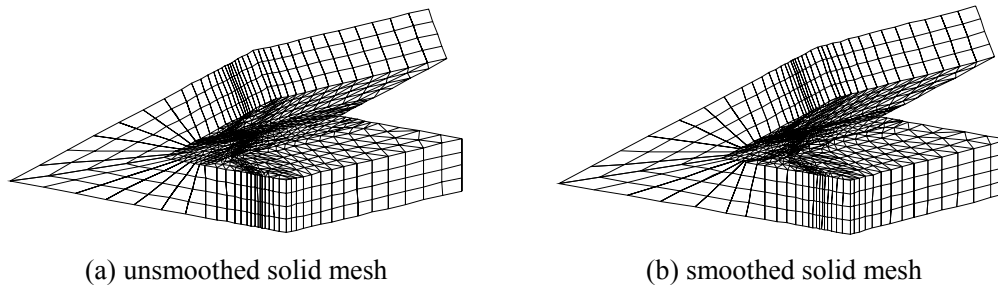


Figure 4.17 Meshes for Example 4.2

Example 4.2 is the one that has been used in Chapter 3, where the surface mesh and volume mesh can be found in Figure 3.23a and Figure 3.23b. In this chapter, the volume mesh is uniformly divided into 4 layers and the process of solid mesh enhancement is applied. The unsmoothed solid mesh and smoothed solid mesh are shown in Figure 4.17a and Figure 4.17b for comparison. The average Θ and $\tilde{\Theta}$ values before smoothing are 0.597249 and 0.560290, respectively. After smoothing, the average Θ and $\tilde{\Theta}$ values are improved to 0.598389 and 0.562124. It can be found that the mesh quality has been slightly improved.

Example 4.3 is a structure constructed by a planar surface and a curved shell. Figure 4.18a shows the input surface mesh and Figure 4.18b shows the extruded volume mesh. In order to show the conformability of the connectivity of nodes and faces, the volume elements that are nearer to the intersection part have been divided

Solid mesh generation

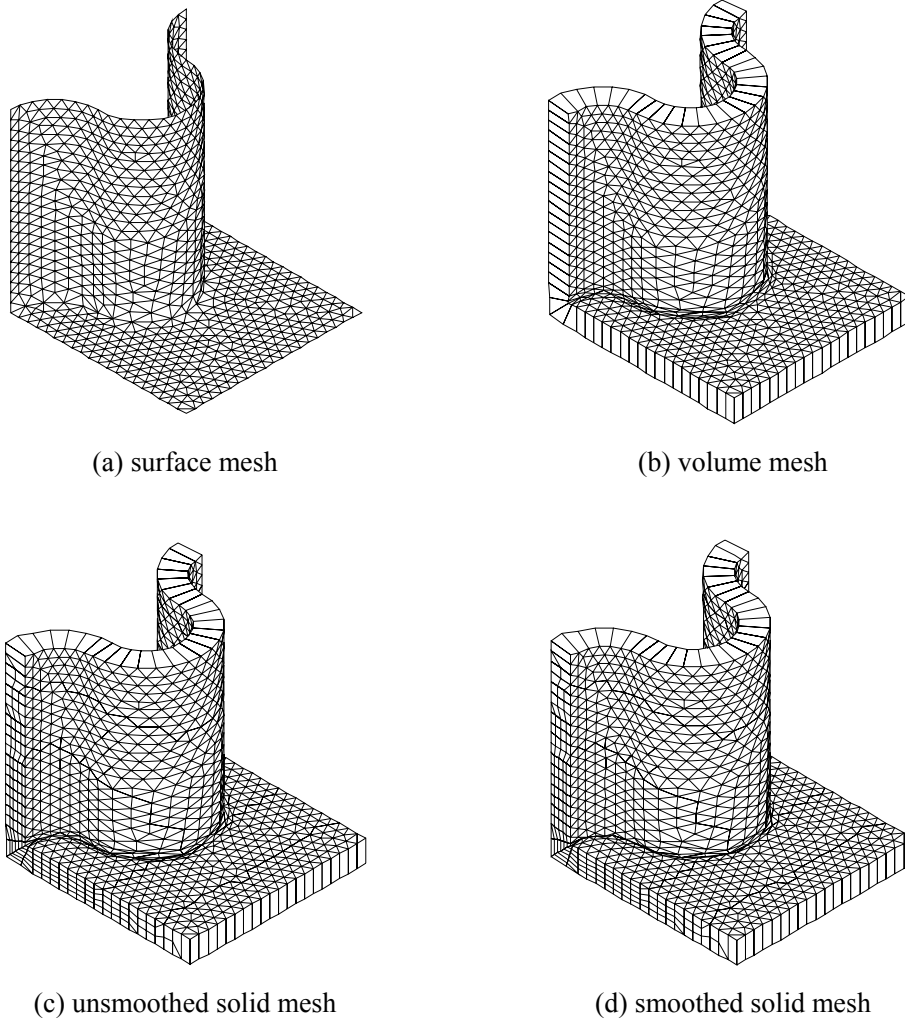
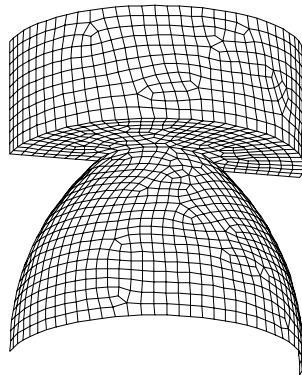


Figure 4.18 Meshes for Example 4.3

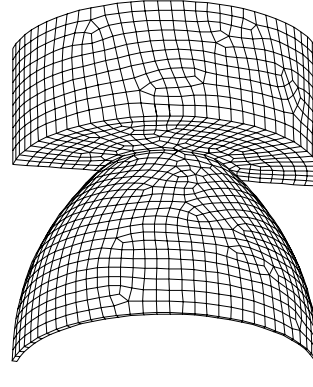
into more layers. The unsmoothed solid mesh is shown in Figure 4.18c and the corresponding Θ and $\tilde{\Theta}$ values are 0.965132 and 0.883057. After smoothing, the Θ and $\tilde{\Theta}$ values of the final solid mesh (Figure 4.18d) are 0.9654864 and 0.891015, respectively. Again, the mesh quality is improved by smoothing.

In Examples 4.4, the structure under consideration is a nozzle [4] constructed by intersection of three shell walls. The uniform input quadrilateral mesh is shown in Figure 4.19a while the extruded volume mesh is shown in Figure 4.19b. Both the wall thickness and the specified element layers in the thickness direction vary along the structures, which can be detected in the solid mesh (Figure 4.19c). Zoom-in

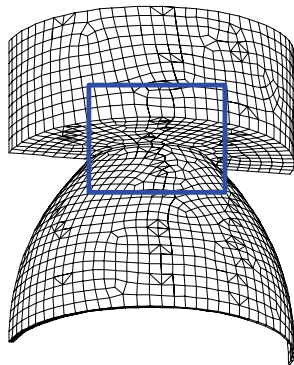
Solid mesh generation



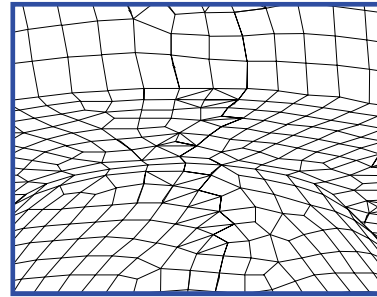
(a) surface mesh



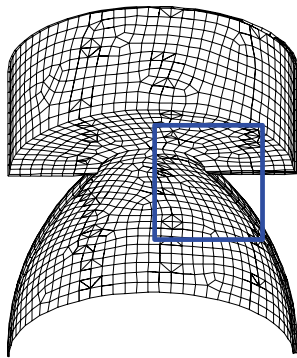
(b) volume mesh



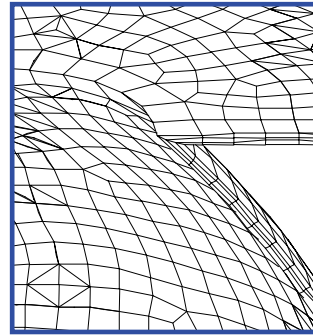
(c) solid mesh, front face



(d) zoom in view for front face



(e) solid mesh, back face



(f) zoom in view for back face

Figure 4.19 Meshes for Example 4.4

views for Figure 4.19c and Figure 4.19e are shown in Figure 4.19d and Figure 4.19f, respectively. In this case, since the number of layers of element varies along the mesh, during the volume mesh refinement step, some hexahedral volume elements are divided into prism volume elements (Figure 4.19c and Figure 4.19e).

Solid mesh generation

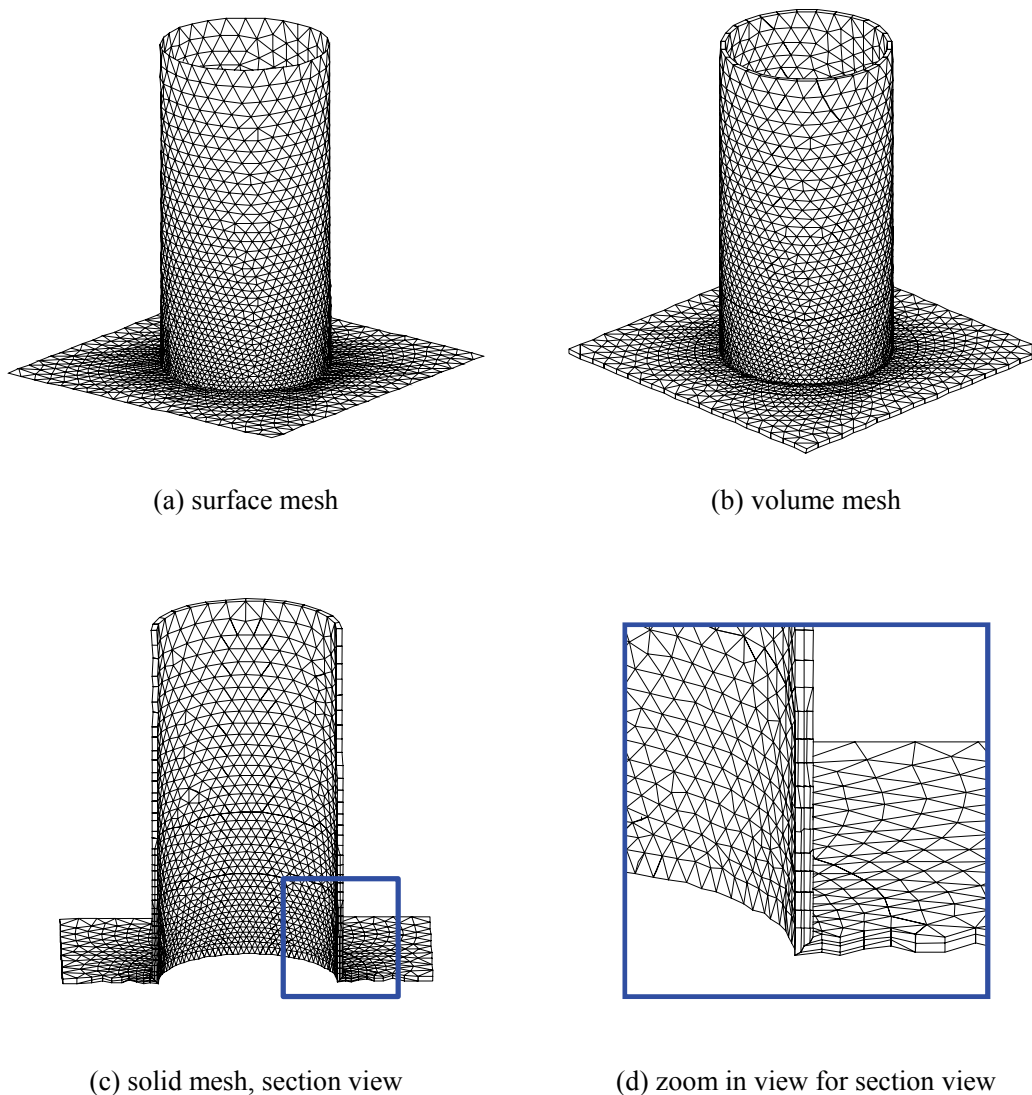
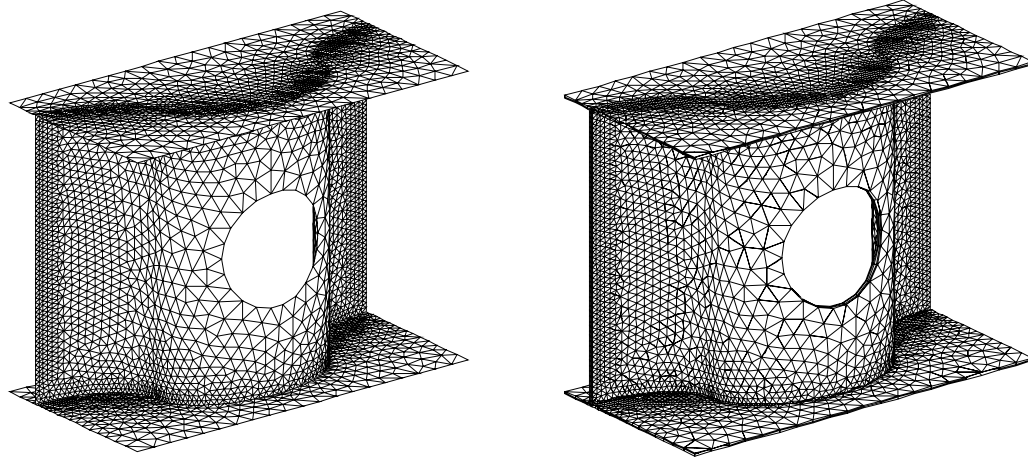


Figure 4.20 Meshes for Example 4.5

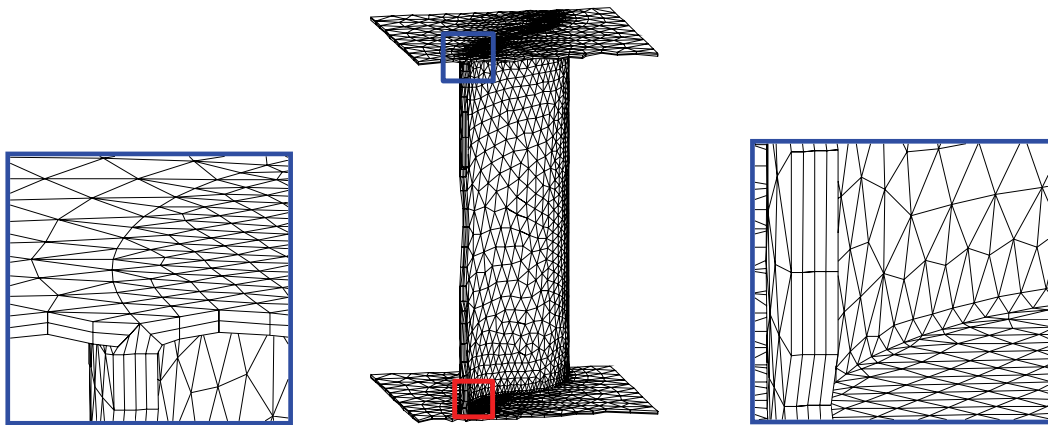
In Examples 4.5 and 4.6, the structures modelled are formed by intersecting curved surfaces with planar surfaces. In Example 5, a cylinder is intersected with a plane and the surface mesh is shown in Figure 4.20a. In this example, smaller size elements and more element layers are used near the intersection; while at the mesh boundary, only one layer of element is used. So the volume mesh and solid mesh look the same from the external view (Figure 4.20b). In order to observe the details inside, a section view is provided in Figure 4.20c and a zoom-in view near the intersection is shown in Figure 4.20d. In Example 6, a multi-cross structure is

Solid mesh generation



(a) surface mesh

(b) volume mesh



(d) zoom in view A

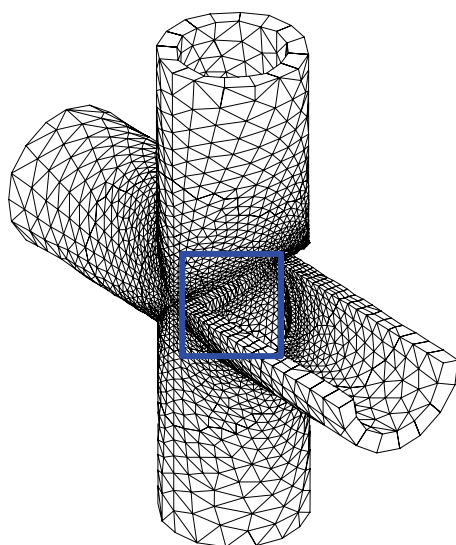
(c) solid mesh, section view

(e) zoom in view B

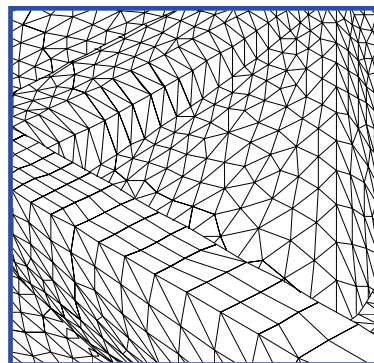
Figure 4.21 Meshes for Example 4.6

formed by the intersections of two planar surfaces with a curved surface (Figure 4.21a). Volume mesh is not given in this example but the 3D solid mesh generated is shown in Figure 4.21b. Its section view and two zoom-in views are shown in Figure 4.21c, Figure 4.21d and Figure 4.21e respectively. Again, in this example, smaller elements and more elements layers are specified near the intersections.

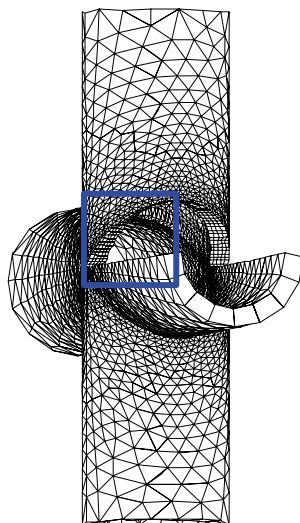
Solid mesh generation



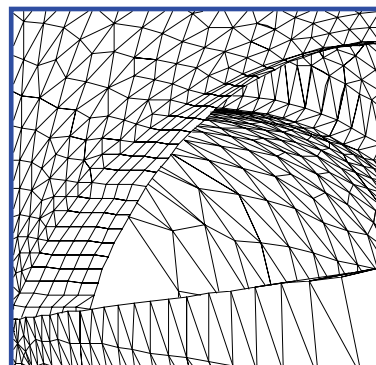
(a) solid mesh, section view I



(b) zoom in view for section view I



(c) solid mesh, section view II



(d) zoom in view for section view II

Figure 4.22 Meshes for Example 4.7

Example 4.7 and 4.8 are the examples that have been used in Chapter 3. The surface meshes and volume meshes for these two examples can be found in Figures 3.26 and 3.27. In this Chapter, only section views of them are provided. For Example 7, section and zoom-in views near the joint are given in Figures 4.20a to 4.20d. It can be seen that in order to obtain an accurate solution near the joint, the mesh is refined towards the intersection in both the surface and thickness directions.

Solid mesh generation

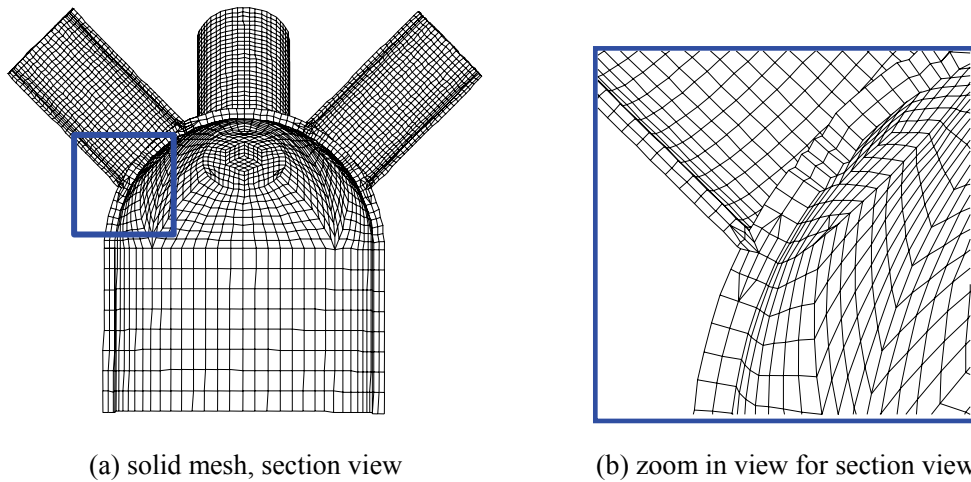


Figure 4.23 Meshes for Example 4.8

For Example 8, the solid mesh generated has four layers of elements along the brace-head intersections while only two layers of elements are used in the rest of the structure. Section and zoom-in views of this example are shown in Figure 4.23a and Figure 4.23b.

4.4 Closure

In this chapter, algorithms to convert the volume meshes to solid meshes are presented. Such a conversion is mainly implemented by a volume mesh refinement procedure. Furthermore, two smoothing techniques or solid mesh quality enhancement are employed to improve the quality of the refined solid mesh.

The volume mesh refinement step is implemented by dividing the elements into layers along thickness direction. A constraint imposed on the layer difference is defined first. Then an algorithm for checking and rectifying the input layers is designed for the case that the input data do not comply with this constraint. Finally, the algorithm for connecting the dividing nodes is introduced. Note that the connecting algorithm is different for triangular input surface mesh and quadrilateral input surface mesh.

Solid mesh generation

In the solid mesh quality enhancement procedure, a normal side smoothing process is first applied to adjust the directions for those normal sides that have been altered in the extrusion step. A solid mesh smoothing process is then carried out to amend the positions of the dividing nodes. Different mesh quality measures have been designed for these two smoothing techniques.

Up to the end of this Chapter, the mesh generation module is finished. The solid mesh obtained can be applied to a FE analysis module for further numerical calculation process.

CHAPTER 5

STRESS RECOVERY AND ERROR ESTIMATION

In Chapter 3 and Chapter 4, the detailed algorithms to convert surface meshes to solid meshes have been presented. In this chapter, efforts will be devoted to the following three parts:

- (1) *Selection of 3D elements for FE analysis*
- (2) *Stress recovery for TWSs*
- (3) *A priori error estimation for TWSs*
- (4) *A posterior error estimation for TWSs*

5.1 Selection of Elements

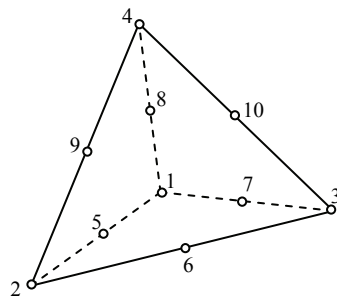
There are different types of elements in the solid mesh, which are generated by the algorithms presented in Chapters 3 and 4. If the input surface mesh consists of only triangular elements, depending on the number of layers of elements (NEL), the final solid mesh may consists of prism elements only (for uniform NEL values) or tetrahedron, pyramid and prism elements (for non-uniform NEL values) . If there are quadrilateral elements existing in the surface mesh, then hexahedron elements will also exist in the solid mesh. The solid element types in the final 3D solid mesh are summarized in Table 5.1.

Note that all the above mentioned elements are first order solid elements. Compared to the first order elements, the second order (quadratic) elements can

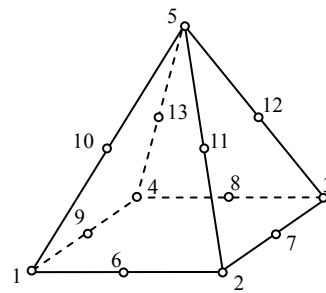
Elements in surface mesh	Uniform NEL	Non-uniform NEL
triangle	prism	tetrahedron, pyramid and prism
quadrilateral	hexahedron	tetrahedron, pyramid, prism and hexahedron
triangle & quadrilateral	prism and hexahedron	tetrahedron, pyramid, prism and hexahedron

Table 5.1 Element types in the final solid mesh

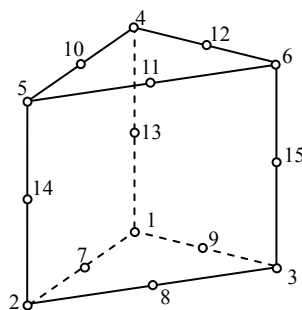
provide more accurate FE results when the number of elements does not change. In order to get a more reliable FE solution, the second order elements are adopted in this study for the FE analysis for TWSs. An algorithm developed by Lee [5] to increase the order of solid elements is used to convert the first order solid meshes into the second order solid meshes. Thus, in the final solid mesh, the following types of elements will be formed: 10-node tetrahedron (T10), 13-node pyramid (P13), 15-node prism (P15) and 20-node hexahedron (H20) (Figure 5.1).



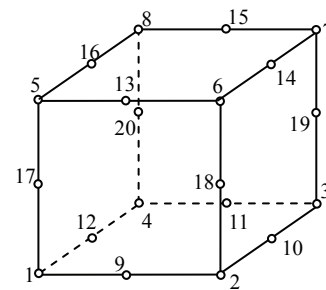
(a) 10-node tetrahedron



(b) 13-node pyramid



(c) 15-node prism



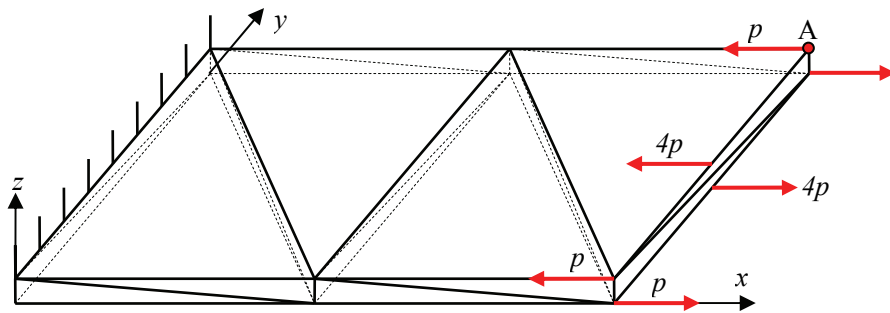
(d) 20-node hexahedron

Figure 5.1 Quadric elements used in the FE analysis

As mentioned in Chapter 1, ABAQUS, a leading FEM software package, is employed to carry out the FE analysis in the current research. Among those quadric elements shown in Figure 5.1, the 13-node element (Figure 5.1b) is not included in the element library of ABAQUS. However, using the user defined subroutine, UEL, which is provided by ABAQUS, a 13-node pyramid element can be developed based on the shape functions presented by Bedrosian [102]. The nodal parent coordinates for this element are listed in Table 5.2.

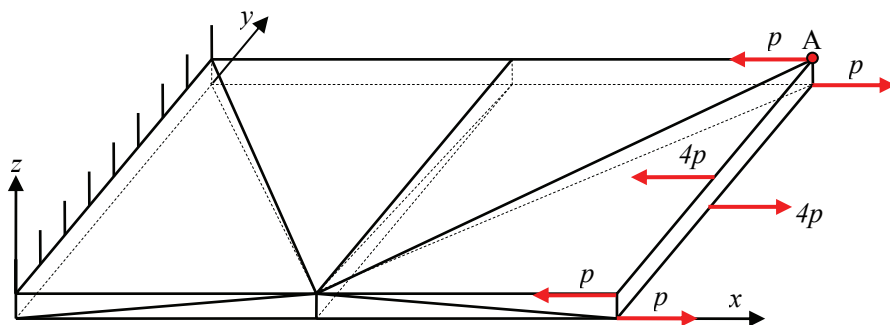
No.	x	y	z
1	-1	-1	0
2	1	-1	0
3	1	1	0
4	-1	1	0
5	0	0	1

Table 5.2 Coordinates of the corner nodes of P13 element



Size: $10 \times 10 \times t$, $E = 2 \times 10^8$, $\nu = 0$

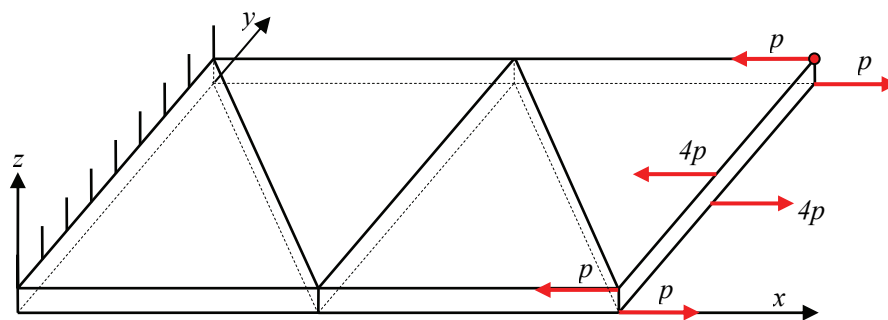
Figure 5.2 Cantilever plate of 10-node tetrahedron elements



Size: $10 \times 10 \times t$, $E = 2 \times 10^8$, $\nu = 0$

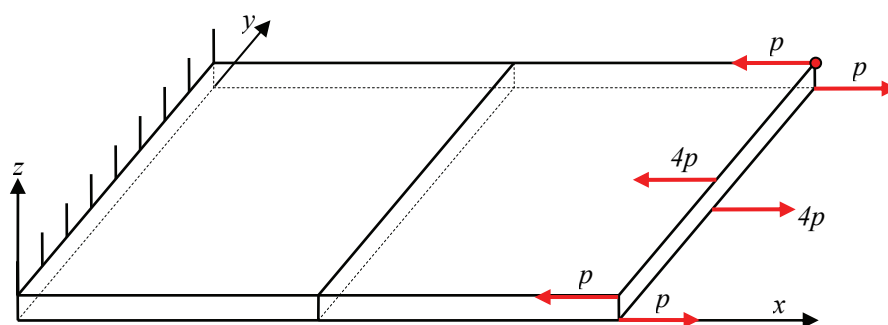
Figure 5.3 Cantilever plate of 13-node pyramid elements

Stress recovery and error estimation



Size: $10 \times 10 \times t$, $E = 2 \times 10^8$, $\nu = 0$

Figure 5.4 Cantilever plate of 15-node prism elements



Size: $10 \times 10 \times t$, $E = 2 \times 10^8$, $\nu = 0$

Figure 5.5 Cantilever plate of 20-node hexahedron elements

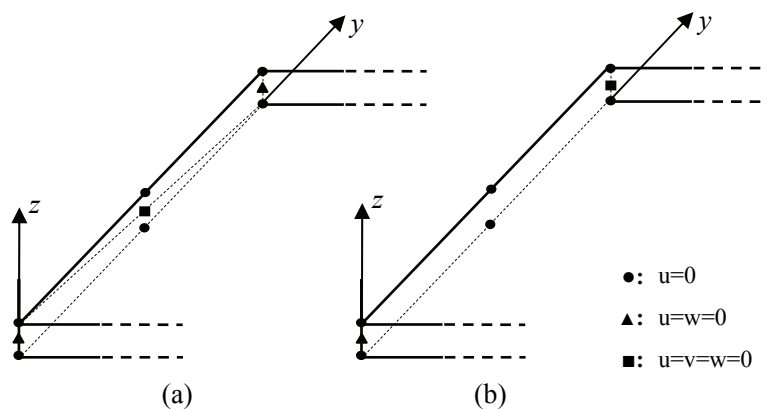


Figure 5.6 Boundary condition for the cantilever plate example:

(a) boundary condition for Figure 5.2; (b) boundary condition for Figure 5.3, Figure 5.4 and Figure 5.5

It should be pointed out that some high-performance elements [8, 108, 109] have been developed recently based on the nonconforming displacement function [110]. In order to compare these elements to the standard quadratic solid elements

Stress recovery and error estimation

in the analysis of TWSs, a simple test is made in this section. In this test, to distinguish from standard solid elements, the high-performance elements are denoted as RNT10 [8], RNP13 [108], RNP15 (see Appendix A) and RNH20 [108], which are corresponding to the tetrahedron, pyramid, prism and hexahedron elements, respectively. A thin, flat plate with a fixed boundary on one end is analyzed (Figures 5.2 to 5.5). On the free end, a bending moment represented by a pair of distributed load is applied. The nodal forces corresponding to the distributed load are shown in Figures 5.2 to 5.5. In order to observe the effect of the plate thickness t , different values of t are considered in this test. The corresponding nodal load values of p for different values of t are given in Table 5.3.

t	0.1	0.05	0.02	0.01	0.005	0.002
p	4.44E+03	1.11E+03	1.78E+02	4.44E+01	1.11E+01	1.78E+00

Table 5.3 Load values for different plate thickness

Element	$t = 0.1$	$t = 0.05$	$t = 0.02$	$t = 0.01$	$t = 0.005$	$t = 0.002$
T10	0.7551	0.7550	0.7550	0.7547	0.7548	0.6286
RNT10	0.7757	0.7752	0.7750	0.7754	0.7764	0.4847
P13	0.7502	0.7500	0.7500	0.7494	0.7528	1.6450
RNP13	0.7511	0.7509	0.7509	0.7513	0.7595	2.386
P15	0.7937	0.7936	0.7936	0.7926	0.7979	0.4646
RNP15	0.7727	0.7728	0.7728	0.7551	0.7623	0.7366
H20	0.7499	0.7500	0.7500	0.7502	0.7547	0.2953
RNH20	0.7997	0.7999	0.8000	0.8000	0.8077	0.9503
Exact	0.8000					

Table 5.4 Deflections for a cantilever plate

Under different loads, the deflections at point A (Figures 5.2 to 5.5) are calculated and listed in Table 5.4. From the results given in Table 5.4, it can be seen

that the high-performance elements do not show an obvious advantage compared to the standard elements. In addition, it is also found that when the plate is extremely thin ($t = 0.002$), both the standard elements and the high-performance elements perform badly. This means when the length to thickness ratio is about $10/0.002 (= 5000)$, all the above solid elements are unable to provide good FE results. In Chapter 1, it has been mentioned that the length to thickness ratio of the structure is set to be within the range of $[10, 100]$ in this thesis, hence this extreme case can be discarded. Consider that the high-performance elements will take more computational time to form the element stiffness in the FE analysis, and not much improvement can be obtained by employing the high-performance elements, the standard quadratic solid elements are used in the current study for the analysis of TWSs.

5.2 Basic Notation

In this section, some basic notations will first be defined for the derivation of the stress recovery procedure and error estimation scheme for TWSs.

5.2.1 Nodal normal vector

The nodal normal vector is defined as a unit vector along *the nodal normal direction*. For a TWS, the nodal normal direction can be defined as the direction approximately perpendicular to the shell surface. If the TWS is formed by shell elements, then the nodal normal direction will be defined at every surface node and this direction is normal to the surface at that node. While if solid elements are used to mesh the TWS, the nodal normal direction will be defined at every corner node (not the mid-side node for a quadratic element) of the elements in the solid mesh. If the corner node is located on the middle surface, its nodal normal direction will be perpendicular to the middle surface. Similarly, if a corner node is located on the

Stress recovery and error estimation

boundary surface (Figure 5.7) of the TWS, its nodal normal direction will be perpendicular to the boundary surface. However, for a dividing node inside the structure, there is no corresponding surface can be found so that the nodal normal direction for this case cannot be clearly defined by geometry description. In a solid mesh generated by the scheme presented in Chapters 3 and 4, the nodal normal vector of an interior dividing node P , which is denoted as $\mathbf{V}_P$, can be determined by the algorithm given in Box 5.1 (Figure 5.7).

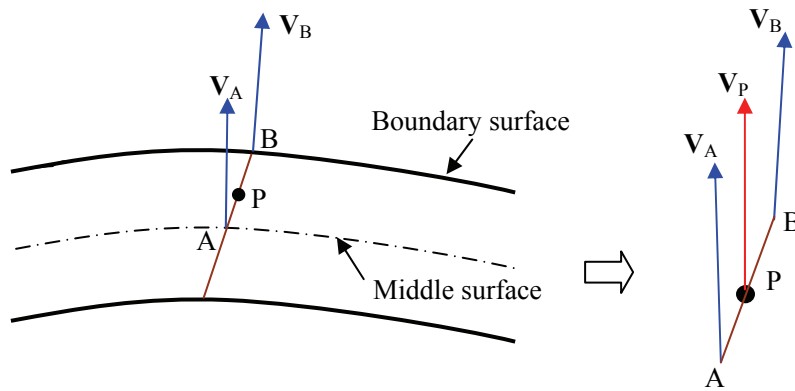


Figure 5.7 Normal of dividing node

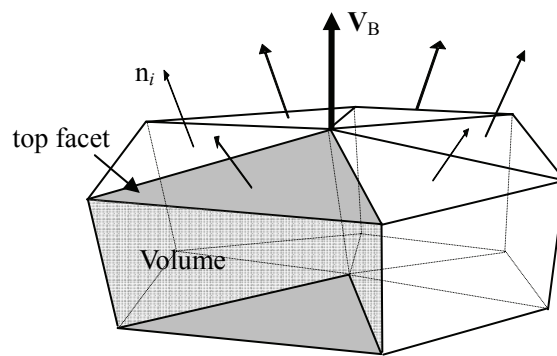


Figure 5.8 Normal vector of a point on boundary surface

- i) Locate the normal side which consists of the dividing node, here is AB in Figure 5.7. A is the corresponding surface mesh node and B is the dividing node on the structure boundary surface formed by top facets (Figure 3.3).
- (ii) Determine $\mathbf{V}_A$, the normal vector at point A, which can be retrieved from the surface extrusion procedure.
- (iii) Determine $\mathbf{V}_B$, the normal vector at point B. Using LSF technique to the normal vectors of all the top facets (Figure 3.3) adjacent to B (Figure 5.8), $\mathbf{V}_B$ can be obtained by solving the equation system

$$\begin{cases} \sum (\mathbf{V}_B \cdot \mathbf{n}_i) \cdot n_i^x = \sum n_i^x \\ \sum (\mathbf{V}_B \cdot \mathbf{n}_i) \cdot n_i^y = \sum n_i^y \\ \sum (\mathbf{V}_B \cdot \mathbf{n}_i) \cdot n_i^z = \sum n_i^z \end{cases} \quad (5.1)$$

- (iv) Calculate $\mathbf{V}_P$, the normal vector at node P by

$$\mathbf{V}'_P = \frac{\mathbf{V}_A \cdot |\mathbf{BP}| + \mathbf{V}_B \cdot |\mathbf{AP}|}{|\mathbf{AB}|} \quad \text{and} \quad \mathbf{V}_P = \frac{\mathbf{V}'_P}{\|\mathbf{V}'_P\|} \quad (5.2)$$

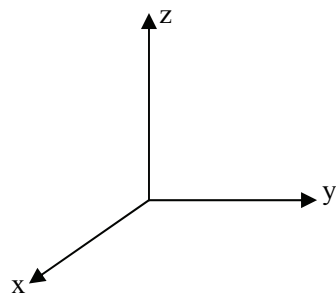
Box 5.1 Definition of the nodal normal vector for an interior corner node

From the above algorithm it can be seen that the nodal normal direction does not have a real geometry definition, it is only a name for a convenient explanation of the algorithms that will be presented. This direction will approach to the normal direction of middle surface when the structure becomes thinner. For an extreme case when the thickness of the structure is zero, the structure can be represented by a surface and the nodal normal direction will coincide with the geometry normal direction of the surface.

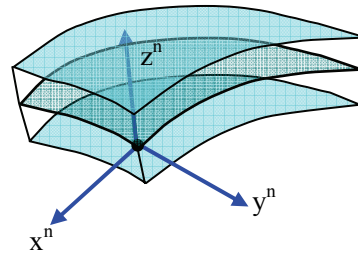
5.2.2 Coordinate Systems

In this thesis, four coordinate systems (Figure 5.9) are used for the definition of the geometry and the deformation of a solid mesh of TWS.

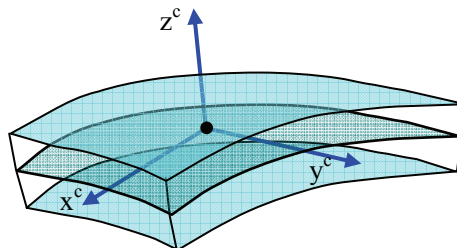
- (1) *Global coordinate system* (x, y, z or x_i). The global Cartesian coordinate system defines the nodal coordinates and the global nodal displacements of the mesh (Figure 5.9a). The three axis vectors of the global coordinate system (GCS) are denoted as $\mathbf{e}_1(0, 0, 1)$, $\mathbf{e}_2(0, 1, 0)$ and $\mathbf{e}_3(0, 0, 1)$.
- (2) *Local coordinate system* (x', y', z' or x'_i). This coordinate system is used to define the local stress and local strain components at any point within the finite element mesh. For any point within the mesh, the z' direction is taken to be normal to the shell surface.
- (3) *Nodal coordinate system* (x^n, y^n, z^n or x_i^n). The nodal coordinate system is a set of local Cartesian coordinate systems established at all corner nodes of elements in the solid mesh (Figure 5.9b). x^n and y^n are along the surface direction and z^n is along the nodal normal direction (Section 5.2.1).
- (4) *Convective coordinate system* (x^c, y^c, z^c or x_i^c). This convective coordinate system is established at any point inside a solid element (Figure 5.9c).



(a) global coordinate system



(b) nodal coordinate system



(c) convective coordinate system

Figure 5.9 Coordinate systems for solid elements of the TWS

The *nodal coordinate system* (NCS) and the *convective coordinate system* (CCS) can be regarded as two particular cases of the *local coordinate system* (LCS). For the case that the position under analysis is at the element corner node, the NCS of this node will represent the LCS. For other cases, the CCS will be used as the LCS (Equation (5.3)).

$$\text{LCS} = \begin{cases} \text{NCS} & \text{for element corner nodes} \\ \text{CCS} & \text{for other positions} \end{cases} \quad (5.3)$$

5.2.2.1 Nodal coordinate system (NCS)

The coordinate axes of the NCS are denoted as $\hat{\mathbf{V}}_1$, $\hat{\mathbf{V}}_2$ and $\hat{\mathbf{V}}_3$. $\hat{\mathbf{V}}_3$ is just the nodal normal vector of the element corner node. The unit vector $\hat{\mathbf{V}}_2$ is simply determined such that it is perpendicular to $\hat{\mathbf{V}}_3$ and parallel to the global y - z plane,

$$\hat{\mathbf{V}}_2 = \frac{\hat{\mathbf{V}}_3 \times \mathbf{e}_1}{|\hat{\mathbf{V}}_3 \times \mathbf{e}_1|} \quad (5.4)$$

and $\hat{\mathbf{V}}_1$ is normal to both $\hat{\mathbf{V}}_2$ and $\hat{\mathbf{V}}_3$

$$\hat{\mathbf{V}}_1 = \frac{\hat{\mathbf{V}}_2 \times \hat{\mathbf{V}}_3}{|\hat{\mathbf{V}}_2 \times \hat{\mathbf{V}}_3|} \quad (5.5)$$

It must be mentioned that Equation (5.4) may fail when $\hat{\mathbf{V}}_3$ is parallel to the global x axis (i.e. $\mathbf{e}_1$). In this case, $\hat{\mathbf{V}}_1$ needs to be established first by

$$\hat{\mathbf{V}}_1 = \frac{\hat{\mathbf{V}}_3 \times \mathbf{e}_2}{|\hat{\mathbf{V}}_3 \times \mathbf{e}_2|} \quad (5.6)$$

and then $\hat{\mathbf{V}}_2$ is obtained by

$$\hat{\mathbf{V}}_2 = \frac{\hat{\mathbf{V}}_3 \times \hat{\mathbf{V}}_1}{|\hat{\mathbf{V}}_3 \times \hat{\mathbf{V}}_1|} \quad (5.7)$$

A transformation matrix from GCS to NCS is defined as

$$\Phi = [\hat{\mathbf{V}}_1 \quad \hat{\mathbf{V}}_2 \quad \hat{\mathbf{V}}_3] \quad (5.8)$$

Then the coordinates in NCS can be calculated by

$$[x'' \quad y'' \quad z'']^T = \Phi^T \cdot [x \quad y \quad z]^T \quad (5.9)$$

And the stress transformation from GCS to NCS is expressed by

$$\boldsymbol{\sigma}^n = \Phi^T \cdot \boldsymbol{\sigma} \cdot \Phi \quad (5.10)$$

where $\boldsymbol{\sigma}^n$ and $\boldsymbol{\sigma}$ are the stress tensors in NCS and GCS, respectively.

5.2.2.2 Convective coordinate system (CCS)

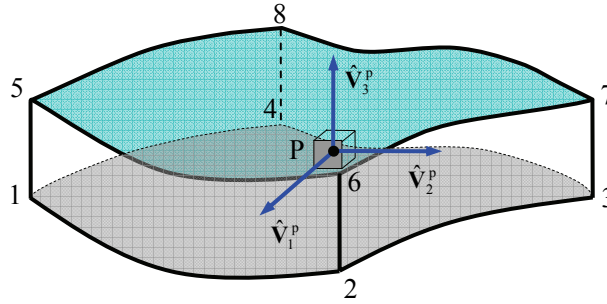


Figure 5.10 Convective coordinate system

The CCS for stress recovery was first proposed by Lee [111] for shell analysis and the essence of this approach is to establish a local convective Cartesian system within an element by interpolating the NCSs at the element corner nodes. In the current study, this coordinate system is also applied to the analysis of TWSs. For shell analysis, the elements involved are surface elements while for 3D analysis, the elements used are solid elements. For a point P within an element (Figure 5.10), the three coordinate axes of the CCS are denoted as $\hat{\mathbf{V}}_1^P$, $\hat{\mathbf{V}}_2^P$ and $\hat{\mathbf{V}}_3^P$. $\hat{\mathbf{V}}_3^P$ can be determined by the interpolation on the nodal normal vectors of all the corner nodes of this element (Equation (5.11)).

Stress recovery and error estimation

$$\hat{\mathbf{V}}_3^P = \frac{\sum_{i=1}^{NEN} N^i \hat{\mathbf{V}}_3^i}{\left| \sum_{i=1}^{NEN} N^i \hat{\mathbf{V}}_3^i \right|} \quad (5.11)$$

In Equation (5.11), N^i is the linear element shape function which belongs to the i th element node at position P. NEN is the number of element nodes, which is 8 for the element shown in Figure 5.10. $\hat{\mathbf{V}}_3^i$ is the nodal normal vector of the i th element corner node. An assistant vector $\hat{\mathbf{V}}_e^P$, is created using the same way (Equation (5.12)).

$$\hat{\mathbf{V}}_e^P = \frac{\sum_{i=1}^{NEN} N^i \hat{\mathbf{V}}_1^i}{\left| \sum_{i=1}^{NEN} N^i \hat{\mathbf{V}}_1^i \right|} \quad (5.12)$$

Then $\hat{\mathbf{V}}_2^P$ and $\hat{\mathbf{V}}_1^P$ are obtained by

$$\hat{\mathbf{V}}_2^P = \frac{\hat{\mathbf{V}}_3^P \times \hat{\mathbf{V}}_e^P}{\left| \hat{\mathbf{V}}_3^P \times \hat{\mathbf{V}}_e^P \right|} \quad (5.13)$$

$$\hat{\mathbf{V}}_1^P = \frac{\hat{\mathbf{V}}_2^P \times \hat{\mathbf{V}}_3^P}{\left| \hat{\mathbf{V}}_2^P \times \hat{\mathbf{V}}_3^P \right|} \quad (5.14)$$

Note that in a solid element, the CCS is not a fixed coordinate system but will vary smoothly within the domain enclosed by this element. Similar to Equations (5.9) and (5.10), the coordinates and stress transformation from GCS to CCS can be expressed by

$$\begin{bmatrix} x^c & y^c & z^c \end{bmatrix}^T = \begin{bmatrix} \hat{\mathbf{V}}_1^P & \hat{\mathbf{V}}_2^P & \hat{\mathbf{V}}_3^P \end{bmatrix}^T \cdot \begin{bmatrix} x & y & z \end{bmatrix}^T \quad (5.15)$$

and

$$\boldsymbol{\sigma}^c = \begin{bmatrix} \hat{\mathbf{V}}_1^P & \hat{\mathbf{V}}_2^P & \hat{\mathbf{V}}_3^P \end{bmatrix}^T \cdot \boldsymbol{\sigma} \cdot \begin{bmatrix} \hat{\mathbf{V}}_1^P & \hat{\mathbf{V}}_2^P & \hat{\mathbf{V}}_3^P \end{bmatrix} \quad (5.16)$$

where (x^c, y^c, z^c) are the coordinates defined in the CCS at point P. $\boldsymbol{\sigma}^c$ and $\boldsymbol{\sigma}$ are the stress tensors defined in the CCS and GCS, respectively.

According to the properties of element shape functions, it can be found that when a point coincides with one of the element corner nodes, the CCS of this point will be the same as the NCS defined at that corner node.

5.2.3 Energy norm and error norm

In general, FEM is used to solve a problem given by a governing differential equation of the form

$$\mathbf{H}\mathbf{u} - \mathbf{q} = 0 \quad (5.17)$$

in the domain Ω with boundary $\partial\Omega$, under the boundary conditions

$$\mathbf{M}\mathbf{u} - \mathbf{t} = 0 \quad (5.18)$$

In Equation (5.17), $\mathbf{H}$ is a differential operator, $\mathbf{q}$ is the corresponding loading and $\mathbf{u}$ is the primary unknown of the problem. In Equation (5.18), $\mathbf{M}$ is another linear operator and $\mathbf{t}$ are the corresponding boundary conditions imposed to the system. For the FEM in mechanical analysis, $\mathbf{u}$ often represents the displacements. Based on $\mathbf{u}$, two generalized derivatives, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$, which are known as the strain and stress, can be obtained by

$$\boldsymbol{\varepsilon} = \mathbf{L}\mathbf{u} \text{ and } \boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \quad (5.19)$$

where $\mathbf{L}$ is a first order differential operator and $\mathbf{D}$ is the material operator (or elasticity matrix). For 3D problems involving isotropic material, the expression for $\mathbf{D}$ is

$$\mathbf{D} = \begin{bmatrix} D_A & D_B & D_B & 0 & 0 & 0 \\ D_B & D_A & D_B & 0 & 0 & 0 \\ D_B & D_B & D_A & 0 & 0 & 0 \\ 0 & 0 & 0 & D_C & 0 & 0 \\ 0 & 0 & 0 & 0 & D_C & 0 \\ 0 & 0 & 0 & 0 & 0 & D_C \end{bmatrix} \quad (5.20)$$

where

$$D_A = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad D_B = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad D_C = \frac{E}{2(1+\nu)} \quad (5.21)$$

In Equation (5.21), E is the elastic modulus and ν is the Poisson's ratio.

For every point, $\mathbf{p} \in \Omega$, the strain energy function $W(\mathbf{p})$ is defined as

$$W(\mathbf{p}) = \boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} = \boldsymbol{\varepsilon}^T \mathbf{D} \boldsymbol{\varepsilon} \quad (5.22)$$

$W(\mathbf{p})$ is a real valued function and always greater than or equal to zero as the material matrix, $\mathbf{D}$, is positive definite. The strain energy of the exact solution, U , can then be written as

$$U = \frac{1}{2} \int_{\Omega} W d\Omega \quad (5.23)$$

The energy norm of the exact solution $\mathbf{u}$, $\|\mathbf{u}\|_{\Omega}$, over the whole domain can be defined as

$$\|\mathbf{u}\|_{\Omega} = \sqrt{\int_{\Omega} (\mathbf{L}\mathbf{u})^T \mathbf{D}(\mathbf{L}\mathbf{u}) d\Omega} = \sqrt{\int_{\Omega} \boldsymbol{\varepsilon}^T \mathbf{D} \boldsymbol{\varepsilon} d\Omega} = \sqrt{\int_{\Omega} \boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} d\Omega} \quad (5.24)$$

From Equations (5.23) and (5.24), the relationship between of the energy norm and strain energy is obtained by

$$\|\mathbf{u}\| = \sqrt{2U} \quad (5.25)$$

The errors of the FE solution are measured by the energy norm of the error, i.e.

$$\|\mathbf{e}\|_{\Omega} = \sqrt{\int_{\Omega} (\mathbf{L}\mathbf{e}_u)^T \mathbf{D}(\mathbf{L}\mathbf{e}_u) d\Omega} = \sqrt{\int_{\Omega} \mathbf{e}_\sigma^T \mathbf{D}^{-1} \mathbf{e}_\sigma d\Omega} = \sqrt{\int_{\Omega} (\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.26)$$

When the problem domain is discretized into a set of finite elements, the square of energy norms can be obtained by summing up the individual element contributions, i.e.

$$\|\mathbf{u}\|_{\Omega}^2 = \sum_{i=1}^{NE} \|\mathbf{u}\|_i^2 \quad \text{and} \quad \|\mathbf{e}\|_{\Omega}^2 = \sum_{i=1}^{NE} \|\mathbf{e}\|_i^2 \quad (5.27)$$

where NE is the total number of elements in the problem domain.

5.2.4 Separation of stress and energy norm

For TWSs, if the LCS (Section 5.2.2) is used to define the stress, then the local stress components $\sigma_{x'}$, $\sigma_{y'}$ and $\tau_{xy'}$ (Figure 5.11) can be grouped to the surface (or membrane) direction components while $\sigma_{z'}$, $\tau_{yz'}$ and $\tau_{zx'}$ (Figure 5.11) can be grouped to the components involved in the normal direction. It is found that the surface direction components often become predominant for TWSs, which means the values of these stress components are much bigger (often 10 times larger) than the stress components involved in the normal direction. According to this characteristic, in shell formulation, $\sigma_{z'}$ is always assumed to be zero and the total energy norm and error norm are divided into the surface part and normal part for separate analysis. In the current study, for the 3D analysis of TWSs, this characteristic of predomination of surface stress components also needs to be considered.

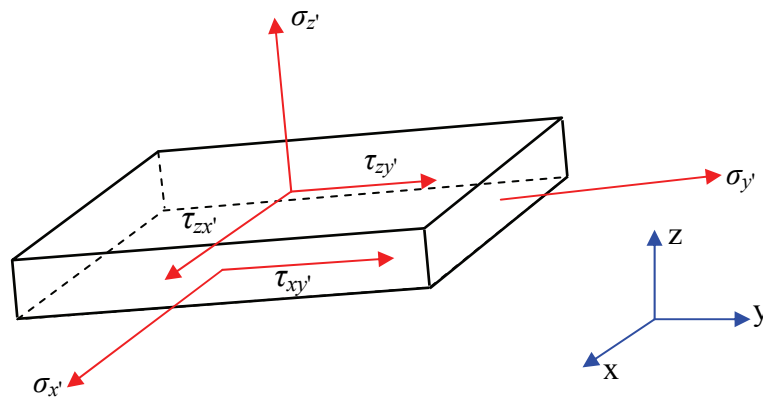


Figure 5.11 Local stress components

In the current study, in order to focus on the stress components in surface direction, similar to the shell analysis, the energy norm should be separated into the surface part and normal part. To achieve this target, it is hoped that the local stress can be partitioned to two parts as

Stress recovery and error estimation

$$\boldsymbol{\sigma} = \begin{bmatrix} \text{surf} \boldsymbol{\sigma}^T & \text{norm} \boldsymbol{\sigma}^T \end{bmatrix}^T \quad (5.28)$$

where $\text{surf} \boldsymbol{\sigma}$ is the vector involved in the surface direction and $\text{norm} \boldsymbol{\sigma}$ is the vector involved in normal direction (how to determine $\text{surf} \boldsymbol{\sigma}$ and $\text{norm} \boldsymbol{\sigma}$ will be discussed later on). The local strain can be accordingly partitioned to

$$\boldsymbol{\epsilon} = \begin{bmatrix} \text{surf} \boldsymbol{\epsilon}^T & \text{norm} \boldsymbol{\epsilon}^T \end{bmatrix}^T \quad (5.29)$$

It is easy to take it for granted that $\sigma_{x'}$, $\sigma_{y'}$ and $\tau_{x'y'}$ are grouped to $\text{surf} \boldsymbol{\sigma}$ while $\sigma_{z'}$, $\tau_{y'z'}$ and $\tau_{z'x'}$ are set to $\text{norm} \boldsymbol{\sigma}$, which is expressed by Equation (5.30),

$$\text{surf} \boldsymbol{\sigma} = [\sigma_{x'}, \sigma_{y'}, \tau_{x'y'}]^T, \quad \text{norm} \boldsymbol{\sigma} = [\sigma_{z'}, \tau_{x'z'}, \tau_{y'z'}]^T \quad (5.30)$$

Then $\text{surf} \boldsymbol{\epsilon}$ and $\text{norm} \boldsymbol{\epsilon}$ will be expressed as

$$\text{surf} \boldsymbol{\epsilon} = [\epsilon_{x'}, \epsilon_{y'}, \gamma_{x'y'}]^T, \quad \text{norm} \boldsymbol{\epsilon} = [\epsilon_{z'}, \gamma_{x'z'}, \gamma_{y'z'}]^T \quad (5.31)$$

In order to maintain the relationship

$$\begin{bmatrix} \text{surf} \boldsymbol{\sigma} \\ \text{norm} \boldsymbol{\sigma} \end{bmatrix} = \mathbf{D} \cdot \begin{bmatrix} \text{surf} \boldsymbol{\epsilon} \\ \text{norm} \boldsymbol{\epsilon} \end{bmatrix} \quad (5.32)$$

The elasticity matrix $\mathbf{D}$ will be partitioned accordingly to

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_{12} \\ \mathbf{D}_{12}^T & \mathbf{D}_2 \end{bmatrix} \quad (5.33)$$

where

$$\mathbf{D}_1 = \begin{bmatrix} D_A & D_B & 0 \\ D_B & D_A & 0 \\ 0 & 0 & D_C \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} D_A & 0 & 0 \\ 0 & D_C & 0 \\ 0 & 0 & D_C \end{bmatrix} \quad \text{and} \quad \mathbf{D}_{12} = \begin{bmatrix} D_B & 0 & 0 \\ D_B & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.34)$$

D_A , D_B and D_C are the values calculated from Equation (5.21). The strain energy function (Equation (5.22)) is then obtained by

$$\begin{aligned} W(p) &= \begin{bmatrix} (\text{surf} \boldsymbol{\epsilon})^T & (\text{norm} \boldsymbol{\epsilon})^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_{12} \\ \mathbf{D}_{12}^T & \mathbf{D}_2 \end{bmatrix} \cdot \begin{bmatrix} \text{surf} \boldsymbol{\epsilon} \\ \text{norm} \boldsymbol{\epsilon} \end{bmatrix} \\ &= (\text{surf} \boldsymbol{\epsilon})^T \cdot \mathbf{D}_1 \cdot \text{surf} \boldsymbol{\epsilon} + (\text{norm} \boldsymbol{\epsilon})^T \cdot \mathbf{D}_{12}^T \cdot \text{surf} \boldsymbol{\epsilon} + (\text{surf} \boldsymbol{\epsilon})^T \cdot \mathbf{D}_{12} \cdot \text{norm} \boldsymbol{\epsilon} + (\text{norm} \boldsymbol{\epsilon})^T \cdot \mathbf{D}_2 \cdot \text{norm} \boldsymbol{\epsilon} \end{aligned} \quad (5.35)$$

Stress recovery and error estimation

In Equation (5.35), it is found that $_{\text{surf}} \boldsymbol{\varepsilon}$ and $_{\text{norm}} \boldsymbol{\varepsilon}$ are coupled together. Hence the energy cannot be separated into two parts which are only involved in $_{\text{surf}} \boldsymbol{\varepsilon}$ and $_{\text{norm}} \boldsymbol{\varepsilon}$, which means such a separation of stress and strain is unsuccessful.

Another idea is to separate the strain energy expression directly by

$$\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} = _{\text{surf}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma}) + _{\text{norm}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma}) \quad (5.36)$$

where

$$_{\text{surf}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma}) = \sigma_{x'} \varepsilon_{x'} + \sigma_{x'} \varepsilon_{x'} + \tau_{x'y'} \gamma_{x'y'} \quad (5.37)$$

and

$$_{\text{norm}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma}) = \sigma_{z'} \varepsilon_{z'} + \tau_{x'z'} \gamma_{x'z'} + \tau_{y'z'} \gamma_{y'z'} \quad (5.38)$$

However, it cannot be proved analytically that $_{\text{surf}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma})$ and $_{\text{norm}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma})$ are non-negative. In fact, it is easy to construct a numerical example such that $_{\text{surf}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma})$ or $_{\text{norm}} (\boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma})$ is less than zero, which violates the rule that the strain energy cannot be negative. Thus, this energy separation method is not acceptable, either.

In the current study, $\sigma_{x'}$, $\sigma_{y'}$, $\sigma_{z'}$ and $\tau_{x'y'}$ are grouped to the surface part while $\tau_{x'z'}$ and $\tau_{y'z'}$ are grouped to the normal part (Equation (5.39)).

$$_{\text{surf}} \boldsymbol{\sigma} = [\sigma_{x'}, \sigma_{y'}, \sigma_{z'}, \tau_{x'y'}]^T, \quad _{\text{norm}} \boldsymbol{\sigma} = [\tau_{x'z'}, \tau_{y'z'}]^T \quad (5.39)$$

Hence the matrix $\mathbf{D}$ can be partitioned as

$$\mathbf{D} = \begin{bmatrix} _{\text{surf}} \mathbf{D} & 0 \\ 0 & _{\text{norm}} \mathbf{D} \end{bmatrix}, \quad _{\text{surf}} \mathbf{D} = \begin{bmatrix} D_A & D_B & D_B & 0 \\ D_B & D_A & D_B & 0 \\ D_B & D_B & D_A & 0 \\ 0 & 0 & 0 & D_C \end{bmatrix}, \quad _{\text{norm}} \mathbf{D} = \begin{bmatrix} D_C & 0 \\ 0 & D_C \end{bmatrix} \quad (5.40)$$

where D_A , D_B and D_C are the values given in Equation (5.21). Then the strain energy function (Equation (5.22)) can be calculated by

Stress recovery and error estimation

$$\begin{aligned} W(p) &= \begin{bmatrix} (\mathbf{\sigma}_{\text{surf}})^T & (\mathbf{\sigma}_{\text{norm}})^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{D}_{\text{surf}}^{-1} & 0 \\ 0 & \mathbf{D}_{\text{norm}}^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{\sigma}_{\text{surf}} \\ \mathbf{\sigma}_{\text{norm}} \end{bmatrix} \\ &= (\mathbf{\sigma}_{\text{surf}})^T \cdot \mathbf{D}_{\text{surf}}^{-1} \cdot \mathbf{\sigma}_{\text{surf}} + (\mathbf{\sigma}_{\text{norm}})^T \cdot \mathbf{D}_{\text{norm}}^{-1} \cdot \mathbf{\sigma}_{\text{norm}} \end{aligned} \quad (5.41)$$

It can be proved that $\mathbf{D}_{\text{surf}}$ and $\mathbf{D}_{\text{norm}}$ obtained by Equation (5.40) are both positive, hence

$$(\mathbf{\sigma}_{\text{surf}})^T \cdot \mathbf{D}_{\text{surf}}^{-1} \cdot \mathbf{\sigma}_{\text{surf}} \geq 0 \quad \text{and} \quad (\mathbf{\sigma}_{\text{norm}})^T \cdot \mathbf{D}_{\text{norm}}^{-1} \cdot \mathbf{\sigma}_{\text{norm}} \geq 0 \quad (5.42)$$

Then the energy norm can be successfully separated into the surface part and normal part by

$$\|\mathbf{u}\|_{\Omega} = \sqrt{\|\mathbf{u}\|_{\Omega}^2_{\text{surf}} + \|\mathbf{u}\|_{\Omega}^2_{\text{norm}}} \quad (5.43)$$

where

$$\begin{cases} \|\mathbf{u}\|_{\Omega}^2_{\text{surf}} = \int_{\Omega} [(\mathbf{\sigma}_{\text{surf}})^T \cdot \mathbf{D}_{\text{surf}}^{-1} \cdot \mathbf{\sigma}_{\text{surf}}] d\Omega \\ \|\mathbf{u}\|_{\Omega}^2_{\text{norm}} = \int_{\Omega} [(\mathbf{\sigma}_{\text{norm}})^T \cdot \mathbf{D}_{\text{norm}}^{-1} \cdot \mathbf{\sigma}_{\text{norm}}] d\Omega \end{cases} \quad (5.44)$$

In Equation (5.44), $\|\mathbf{u}\|_{\Omega}^2_{\text{surf}}$ and $\|\mathbf{u}\|_{\Omega}^2_{\text{norm}}$ are the surface energy norm and the normal energy norm of the solution, respectively. It needs to be pointed out that in this separation method, the surface part of energy norm is not only contributed by the surface direction stress components since σ_z is also added to this part. Consider that σ_z , the local normal stress, is trivial compared to the surface stress components and even omitted in shell analysis, such kind of a separation is still acceptable.

5.3 Stress Recovery for TWSs

In the FEM, the FE stresses are derived from the computed displacements. Since FEM can only provide C_0 continuity at the element boundaries, the functional stresses are thus discontinuous at the element boundaries so that they are generally less accurate than the primary quantities (the displacements). In order to enhance the accuracy of the computed stresses, some *stress recovery* schemes have been developed, among which the *superconvergent patch recovery* (SPR) technique is the

most famous one because of its easy implementation and good performance. In the current study, this technique is applied to the analysis of TWSs.

5.3.1 A brief review on the SPR technique

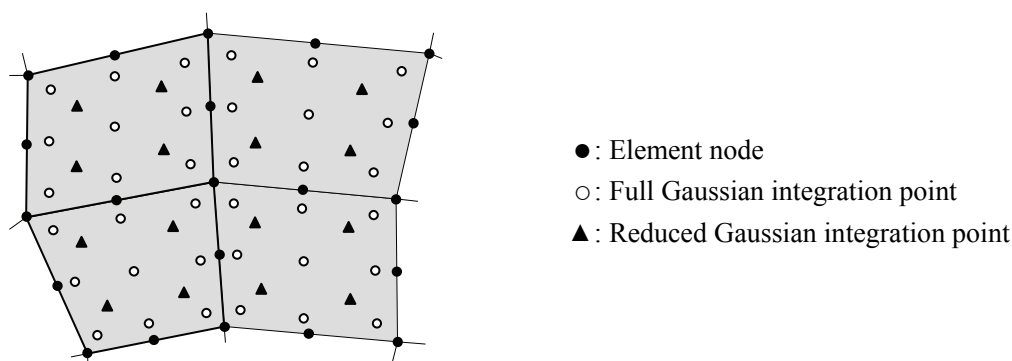


Figure 5.12 Superconvergent points within the 8-node quadrilateral elements

It is found that within an element, the FE stresses at some special points are more accurate than the stresses elsewhere. These special points are defined as the superconvergent points because they can provide higher convergence rate of stresses compared to the stresses at other locations of the element. It is also found that for most of the cases, the positions of the superconvergent points are coincident with the positions of the reduced Gaussian integration points for complete exact integration of an undistorted element (Figure 5.12).

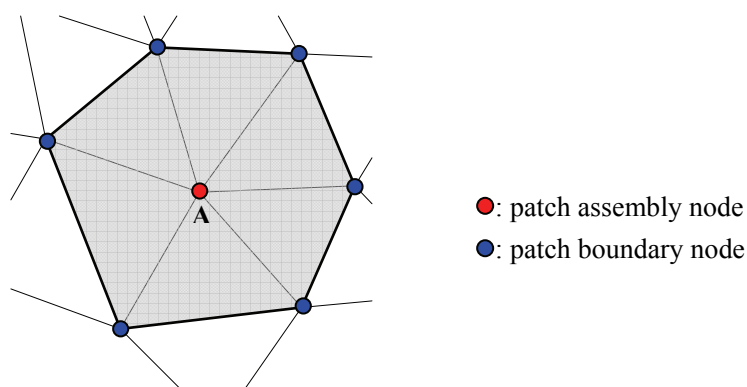


Figure 5.13 Element patch of node A

Based on this phenomenon, the superconvergent patch recovery technique [10, 112, 113] was created. In this technique, an element patch will be established for every nodal point and the patch will consist of all the neighboring elements of that nodal point. In Figure 5.13, an element patch is created for node A. A is called the *patch assembly node*. The element corner nodes which are located on the boundary of the patch are named as the *patch boundary nodes*. Within the element patch, a continuous stress field is constructed and the recovered nodal stress $\tilde{\sigma}$ is expressed by a polynomial expansion:

$$\tilde{\sigma} = \mathbf{P}\mathbf{a} \quad (5.45)$$

where $\mathbf{P}$ is a polynomial about coordinates x, y and z . $\mathbf{a}$ are the unknown parameters of the polynomial. With the data of FE stresses at some *sampling points* within the mesh, $\mathbf{a}$ can be determined via least square fitting. By minimizing the error function

$$\begin{aligned} F(\mathbf{a}) &= \sum_{i=1}^n (\hat{\sigma}(x_i, y_i) - \tilde{\sigma}(x_i, y_i))^2 \\ &= \sum_{i=1}^n (\hat{\sigma}(x_i, y_i) - \mathbf{P}(x_i, y_i)\mathbf{a})^2 \end{aligned} \quad (5.46)$$

it can be obtained that

$$\sum_{i=1}^n \mathbf{P}^T(x_i, y_i) \mathbf{P}(x_i, y_i) \mathbf{a} = \sum_{i=1}^n \mathbf{P}^T(x_i, y_i) \hat{\sigma}(x_i, y_i) \quad (5.47)$$

where n is the number of sampling points in the patch; $\hat{\sigma}$ and (x_i, y_i, z_i) are the FE stress and coordinates at the i th sampling point respectively. Therefore, $\mathbf{a}$ is solved by

$$\mathbf{a} = \mathbf{A}^{-1}\mathbf{b} \quad (5.48)$$

where

$$\begin{cases} \mathbf{A} = \sum_{i=1}^n \mathbf{P}^T(x_i, y_i, z_i) \cdot \mathbf{P}(x_i, y_i, z_i) \\ \mathbf{b} = \sum_{i=1}^n \mathbf{P}^T(x_i, y_i, z_i) \cdot \hat{\sigma}(x_i, y_i, z_i) \end{cases} \quad (5.49)$$

If superconvergent points are selected as the sampling points, then it can be proved that the recovered stresses are also superconvergent [10] and hence they are more accurate than the nodal stresses which are obtained from FE solution directly.

5.3.2 Modified SPR technique for single surface structures

It has been mentioned in Section 5.2.4 that for TWSs, the stresses are separated into the surface direction part and normal direction part. Hence in the stress recovery procedure, the stresses should be projected to the local coordinate system and some modification should be made for the conventional SPR method.

5.3.2.1 Selection of appropriate polynomial terms

Generally, the ideal polynomial terms used in the SPR technique should be identical to those terms occurring in element shape functions. In consideration of the characteristic of surface direction predomination, on the premise that the number of terms for surface direction coordinates are kept unchanged, a relatively less number of terms can be used for the normal direction coordinates. Such a treatment can save the computational time for the stress recovery procedure while the accuracy of the recovered stresses will not be affected too much. In the current study, since the quadratic solid elements are used for TWSs, the highest orders of x , y and z appearing in the shape functions are all equal to 2. Suppose x' and y' are the local surface direction coordinates and z' is the local normal direction coordinate, then in the polynomial $\mathbf{P}$ used for stress recovery, with the intention of focusing on the terms involving the surface direction coordinates, the highest order of z' coordinate can be reduced to 1, which means a linear variation is assumed for the stress components along the normal direction. Such an assumption is also made in shell element analysis because the energy contribution by normal direction stress components is trivial. For this reason, the polynomial $\mathbf{P}$ can be formed by

multiplying a quadratic surface element shape function with a linear polynomial of z' . For example, the standard polynomial terms used for a P15 element, which are express by

$$\{1, x', y', z', x'^2, y'^2, z'^2, x' y', y' z', z' x', x' y' z', x'^2 z', y'^2 z', z'^2 x', z'^2 y'\} \quad (5.50)$$

are now simplified to

$$\begin{aligned} & \{1, x', y', x'^2, x' y', y'^2\} \cdot \{1, z'\} \\ &= \{1, x', y', x'^2, x' y', y'^2, z', z' x', y' z', x'^2 z', x' y' z', y'^2 z'\} \end{aligned} \quad (5.51)$$

Note that the number of polynomial terms is reduced from 15 to 12, as a calculation for $\mathbf{P}^T \cdot \mathbf{P}$ (Equation (5.49)) is needed for every element patch, such a decrease of polynomial terms will reduce the computational cost.

5.3.2.2 Local patch coordinate system

Note that within a single element patch, there must be a unique definition for the local coordinates so that Equation (5.49) can be applied. Thus, it is required to create a *patch coordinate system* for each element patch. The patch coordinate system can be regarded as a local coordinate system within the patch, hence the patch coordinates are still denoted as x' , y' and z' . In the patch coordinate system, Equation (5.49) will be rewritten as

$$\begin{cases} \mathbf{A} = \sum_{i=1}^n \mathbf{P}^T(x'_i, y'_i, z'_i) \cdot \mathbf{P}(x'_i, y'_i, z'_i) \\ \mathbf{b} = \sum_{i=1}^n \mathbf{P}^T(x'_i, y'_i, z'_i) \cdot \hat{\sigma}(x'_i, y'_i, z'_i) \end{cases} \quad (5.52)$$

Note that the patch assembly node is set as the origin of the patch coordinate system so that the recovered stresses at this point can be simply obtained by replacing (x', y', z') with $(0, 0, 0)$ for $\mathbf{P}$ in Equation (5.52).

There are two alternative approaches to establish the local path coordinate system in this study. One is referred as the *SPRI* method and the other is referred as

the *SPR2* method.

(1) *SPR1 method*

In the first approach, the NCS (Section 5.2.2.1) of patch assembly node will be used as the local patch coordinate system and the patch assembly node is the origin of this coordinate system. Since the patch coordinate system is created at the specified node, it is a fixed system within the patch. For every sampling point within the patch, it will be projected to this coordinate system and thus its patch coordinates can be obtained.

(2) *SPR2 method*

In the second approach, the local patch coordinate system is established by the concept of CCS (Section 5.2.2.2). For every sampling point, its CCS will be created first. Then the CCS will be moved by changing the origin position to the position of patch assemble node. The local patch coordinate system will be a group of the above moved CCSs for all the sampling points. Hence, in this approach, the local patch coordinate system is not a fixed system but will vary at different locations within the patch.

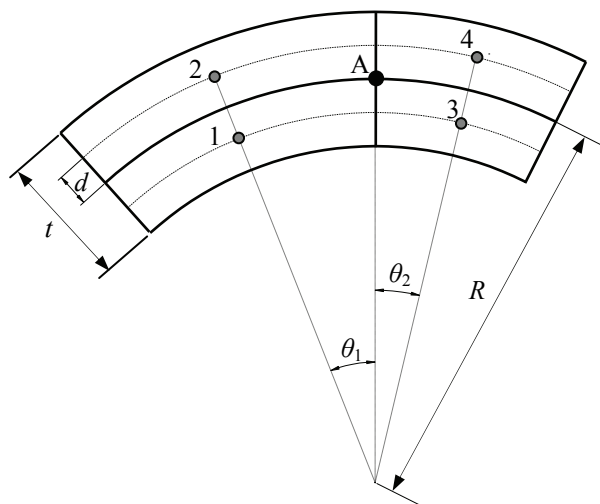


Figure 5.14 A simple 2D patch for the comparison of SPR1 and SPR2

In order to determine which method should be adopted, a simple test is used to check which method is more accurate. To simplify the analysis, a 2D mesh (should be regarded as the 2D section view of a curved shell) is used, which is shown in Figure 5.14. A is the patch assembly node and there are a total of four linear elements in this patch. For each element, there is one sampling point so that there are a total of 4 sampling points, which are denoted as point 1, 2, 3 and 4. Each element has two long edges and two short edges. All the long edges are arcs and they are sharing the same center point. Node A is on the arc whose radius is R . Points 1 and 3 are on the arc whose radius is $R-d$. Points 2 and 4 are on the arc whose radius is $R+d$.

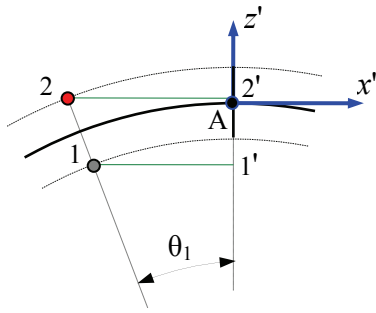


Figure 5.15 SPR1 method

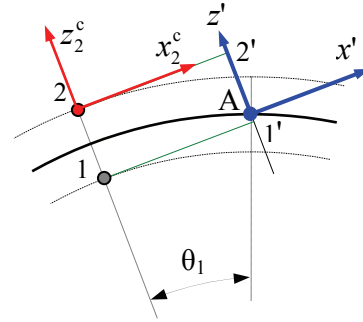


Figure 5.16 SPR2 method

According to the characteristic of the mesh, when using SPR1 method, the patch coordinates for the sampling points are (refer to Figure 5.15, where x' -A- z' is the NCS for node A and it is also the patch coordinate system)

$$\begin{cases} x'_1 = -(R-d) \cdot \sin \theta_1 \\ x'_2 = -(R+d) \cdot \sin \theta_1 \\ x'_3 = (R-d) \cdot \sin \theta_2 \\ x'_4 = (R+d) \cdot \sin \theta_2 \end{cases} \quad \text{and} \quad \begin{cases} z'_1 = (R-d) \cdot \cos \theta_1 - R \\ z'_2 = (R+d) \cdot \cos \theta_1 - R \\ z'_3 = (R-d) \cdot \cos \theta_2 - R \\ z'_4 = (R+d) \cdot \cos \theta_2 - R \end{cases} \quad (5.53)$$

when using SPR2 method, the patch coordinates for the super convergent points are (refer to Figure 5.16, where x_2^c -2- z_2^c and x' -A- z' are the CCS and patch coordinate system for sampling point 2, respectively)

Stress recovery and error estimation

$$\begin{cases} x'_1 = -R \cdot \sin \theta_1 \\ x'_2 = -R \cdot \sin \theta_1 \\ x'_3 = R \cdot \sin \theta_2 \\ x'_4 = R \cdot \sin \theta_2 \end{cases} \quad \text{and} \quad \begin{cases} z'_1 = (R-d) - R \cdot \cos \theta_1 \\ z'_2 = (R+d) - R \cdot \cos \theta_1 \\ z'_3 = (R-d) - R \cdot \cos \theta_2 \\ z'_4 = (R+d) - R \cdot \cos \theta_2 \end{cases} \quad (5.54)$$

In order to simplify the analysis, only the normal stress will be compared. As the normal stress distribution of TWSs can be approximately regarded as linear, in this test, the distribution of the exact normal stress, $\sigma_{z'}$, is assumed to be

$$\sigma_{z'} = 5z' \quad (5.55)$$

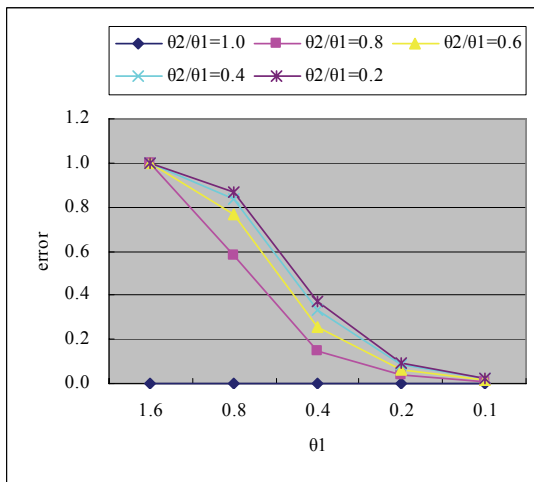
Hence, the recovered normal stress, denoted as $\tilde{\sigma}_{z'}$, can be written as the following polynomial expansion

$$\tilde{\sigma}_{z'} = a_0 + a_1 z' \quad (5.56)$$

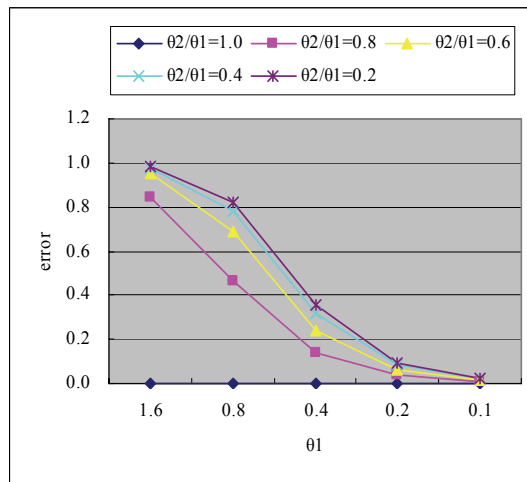
The relative stress error for this test is defined as

$$error = \sum_{i=1}^4 (\tilde{\sigma}_{z'} - \sigma_{z'})^2 / \sum_{i=1}^4 (\sigma_{z'})^2 \quad (5.57)$$

The test will check the effect on influence of the ratio of sizes of neighboring elements and the effect on influence of the element thickness.

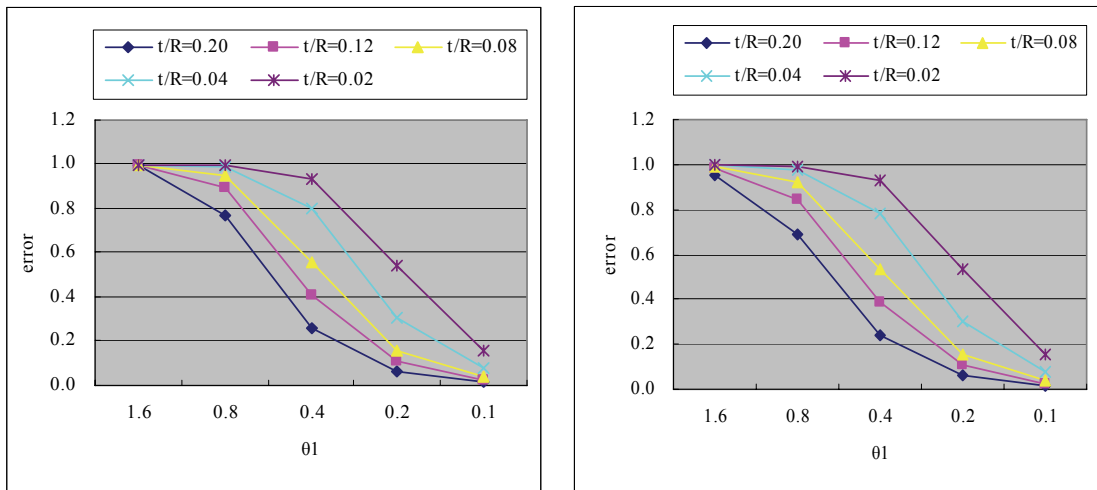


(a) SPR1 error ($R = 20, t = 8$)



(b) SPR2 error ($R = 20, t = 8$)

Stress recovery and error estimation



(c) SPR1 error ($R = 20$, $\theta_2/\theta_1 = 0.6$)

(d) SPR2 error ($R = 20$, $\theta_2/\theta_1 = 0.6$)

Figure 5.17 Comparison of SPR1 and SPR2

Results from Figure 5.17a and Figure 5.17b show the effect on influence of the ratio of the sizes of neighboring elements. It can be found that when the ratio (θ_2/θ_1) is increasing, the error is becoming bigger. Results from Figure 5.17c and Figure 5.17d show the effect on influence of the shell thickness. It can be seen that when the shell thickness (t) is decreasing, the error is becoming bigger. By comparing the errors of SPR1 and SPR2 from Figure 5.17, it can be found that SP2 can generate smaller error than SPR1. However, such an advantage is not that obvious. Therefore, the recovery effect of SPR2 is only slightly better than that of SPR1. Consider that a lot of operations of transformation between coordinate systems are needed in SPR2, in order to save the computational time, only SPR1 is used for all the numerical tests in this thesis.

5.3.2.3 Scaling of patch coordinates

It must be pointed out that when applying the SPR1 technique, it is required to carry out many calculations on the inversion of matrix A (Equation (5.52)). If some matrices are bad conditioned so that they are nearly to be singular, then the

computational error is big and may lead to an unreasonable result. In order to improve the computational stability, it is necessary to improve the condition of the matrix A . To resolve this problem, a frequently used method is to use a locally normalized coordinate system within the patch [112, 114]. Such a technique is known as the *scaling* and the normalized patch coordinates are denoted as x^p, y^p and z^p in this thesis.

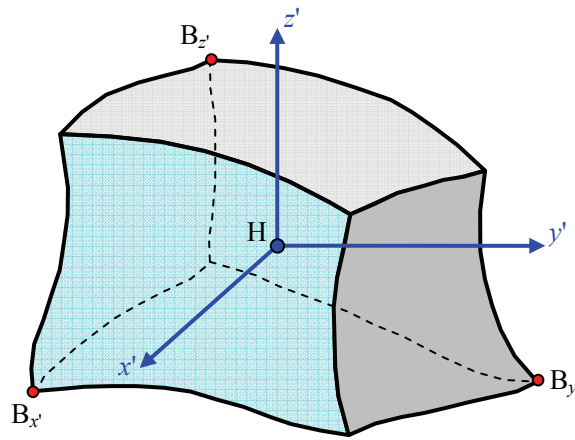


Figure 5.18 Maximum boundary distances of the patch

In order to restrict all the patch coordinate components to the range $[-1, 1]$, the *patch maximum boundary distances* must be determined first. Patch maximum boundary distances consist of three components: $MD_{x'}$, $MD_{y'}$ and $MD_{z'}$, which are along x' , y' and z' directions respectively. Suppose the patch assembly node is denoted as H and the patch boundary node is denoted as B_i ($i = 1, \dots, N$ where N is the number of patch boundary nodes), $MD_{x'}$, $MD_{y'}$ and $MD_{z'}$ are determined by

$$\begin{cases} MD_{x'} = \max(|\overrightarrow{HB_i} \cdot \hat{\mathbf{V}}_1|) \\ MD_{y'} = \max(|\overrightarrow{HB_i} \cdot \hat{\mathbf{V}}_2|) \\ MD_{z'} = \max(|\overrightarrow{HB_i} \cdot \hat{\mathbf{V}}_3|) \end{cases} \quad (5.58)$$

$\hat{\mathbf{V}}_1$, $\hat{\mathbf{V}}_2$ and $\hat{\mathbf{V}}_3$ are the coordinate axis vectors of the NCS at the patch assembly point. Note that $MD_{x'}$, $MD_{y'}$ and $MD_{z'}$ are not the projected values of the maximum

distances from one boundary node to the patch assembly node, but belong to three different nodes. In an element patch shown in Figure 5.18, $MD_{x'}$, $MD_{y'}$ and $MD_{z'}$ are calculated from three different boundary nodes, which are $B_{x'}$, $B_{y'}$ and $B_{z'}$.

After the patch maximum boundary distances are determined, the normalized patch coordinates of the i th sampling point (x_i^p, y_i^p, z_i^p) can be calculated by

$$x_i^p = \frac{x'_i}{MD_{x'}}; y_i^p = \frac{y'_i}{MD_{y'}}; z_i^p = \frac{z'_i}{MD_{z'}} \quad (5.59)$$

where (x'_i, y'_i, z'_i) are the patch coordinates of the sampling point. Therefore, Equation (5.52) will be further modified to

$$\begin{cases} \mathbf{A} = \sum_{i=1}^n \mathbf{P}^T(x_i^p, y_i^p, z_i^p) \cdot \mathbf{P}(x_i^p, y_i^p, z_i^p) \\ \mathbf{b} = \sum_{i=1}^n \mathbf{P}^T(x_i^p, y_i^p, z_i^p) \hat{\sigma}(x_i^p, y_i^p, z_i^p) \end{cases} \quad (5.60)$$

5.3.2.4 Boundary patch enhancement

When the SPR technique is used for the stress recovery, it is required that the *number of sampling points* (NSP) cannot be less than m , the number of the parameters in $\mathbf{a}$ (Equation (5.45)). This requirement is expressed by

$$NSP \geq m \quad (5.61)$$

It is found that for a patch established on an interior node (*Interior Patch*), the above condition can be satisfied for most of the cases; while for the patch created at a boundary node (*Boundary Patch*), there may not be enough sampling points in the patch so that the requirement given in Equation (5.61) cannot be satisfied. To resolve this problem, Zienkiewicz and Zhu [10] suggest using the interior patches that contain the boundary nodes to evaluate the stresses of the boundary nodes.

However, this method cannot always be followed for some boundary elements that have no interior node. This case usually happens when the boundary element is a triangle for 2D domain or a tetrahedron for 3D domain (Figure 5.19).

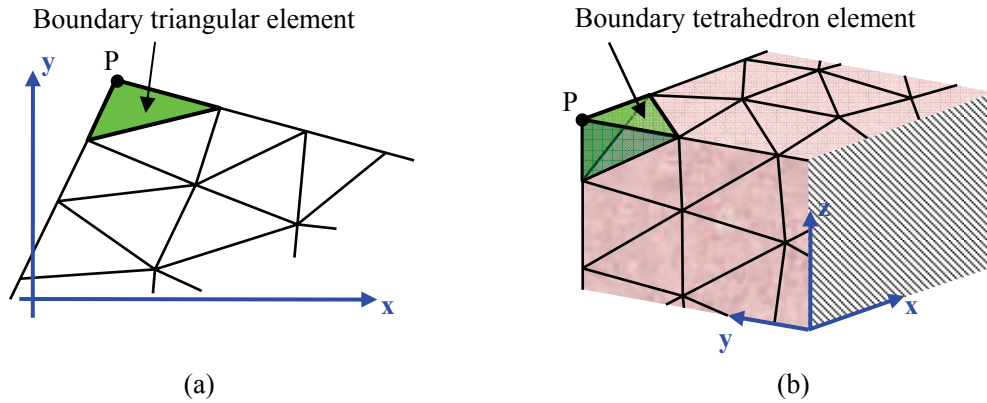


Figure 5.19 Patch of a boundary element which does not consist of interior node:

- (a) boundary triangular element in a 2D mesh
- (b) boundary tetrahedron element in a 3D mesh

In Figure 5.19, the patch of node P has only one element and there is no interior node in this patch so that the procedure proposed by Zienkiewicz and Zhu cannot be used. In order to provide enough sampling points so that the condition (5.61) can be satisfied, Labbe and Garon [115] suggested using the full Gaussian integration points which are used for the element stiffness formation as the sample stress points for boundary patches. This treatment ensures that the systems of equations for the patches are always solvable, but it will lower the convergence rate of the recovered stresses at the boundary nodes. Hence, the recovered stresses at the boundary are not as accurate as those of the interior nodes. However, in order to assure that there are sufficient sampling points for the boundary patch, this method is adopted in the current study. Note that the method of using full Gaussian integration points as the sampling points will not be applied to all the boundary patches but only be used for those patches that cannot satisfy the condition given in (5.61).

It is found that even there are enough sampling points for a patch, the matrix $\mathbf{A}$ in Equation (5.49) may still be ill-conditioned so that the recovered stresses will be unreasonable. Such cases often happen in boundary patches due to the element orientation. In order to improve the accuracy as well as the stability of the local projection scheme, Wiberg and Abdulwahab [116] proposed the method to impose equilibrium equation to the recovery scheme. Belytschko and Blacker inherited [117] this method and they further presented the algorithm of imposing natural boundary conditions to enhance the SPR technique. Implementation on equilibrium equation and natural boundary conditions was also discussed by Lo and Lee [118]. Although these two methods can significantly improve the accuracy of the recovered stresses for most cases, they also bring about a great amount of additional computational tasks to the recovery process. In order to achieve a better recovery effect at relatively lower computational cost, Ródenas, etc. [119] proposed a modified SPR technique (SPR-R), which imposes the given boundary tractions on the boundary patches before solving the systems of equations. For example, if a component of the recovered stress in a patch, denoted as $\tilde{\sigma}_i$, is expressed by

$$\tilde{\sigma}_i = a_{i1} + a_{i2}x^p + a_{i3}y^p + a_{i4}z^p + a_{i5}(x^p)^2 + \dots \quad (5.62)$$

then at the patch assembly node where the patch coordinates are $(0, 0, 0)$, $\tilde{\sigma}_i$ can be simply obtained by

$$\tilde{\sigma}_i = a_{i1} \quad (5.63)$$

When σ_i , the exact stress component at this node is already known from the boundary traction value b_i , then the recovered stress can be replaced by b_i directly so that

$$a_{i1} = b_i \quad (5.64)$$

Hence, the unknown a_{i1} is determined before solving Equation (5.48).

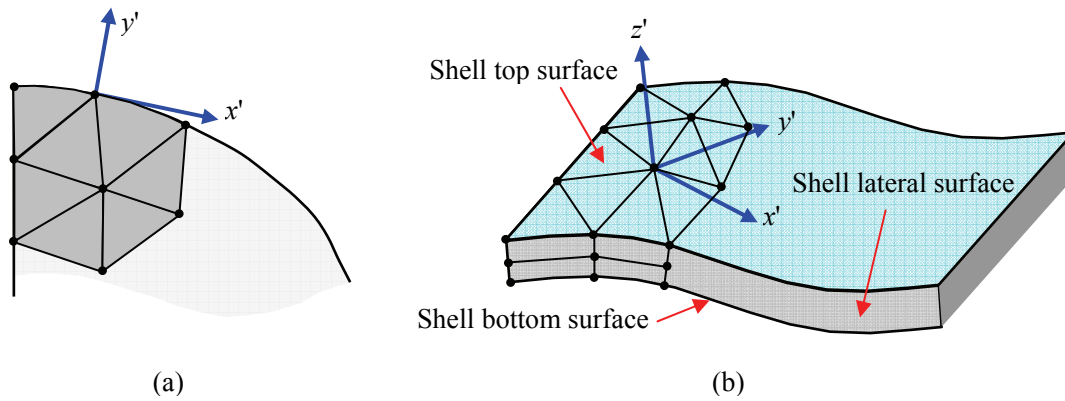


Figure 5.20 Local coordinate system used for SPR-R technique:
(a) 2D mesh; (b) 3D mesh for TWS

The SPR-R technique proposed by Ródenas, etc. is designed for 2D elements. In the current study, it is applied to the 3D elements of solid meshes of TWSs. It must be mentioned that the boundary tractions are referred to the values defined in local coordinate systems at the boundary nodes. For a 2D mesh, the local coordinate system is formed along the boundary tangent direction and normal direction (Figure 5.20a). While for a 3D mesh used for TWSs, the local coordinate system is formed along the surface direction and normal direction for the shell top surface and bottom surface (Figure 5.20b). At the shell lateral surfaces, this technique is not applied. Note that for 3D meshes of TWSs, the local coordinate system is just set to the NCS which has been defined in Section 5.2.2. The SPR-R technique is simple and intuitive and its application requires a reduced extra computational cost with respect to the SPR technique.

5.3.3 Stress recovery for structures with junction

For the structures formed by intersected surfaces, there is no distinct surface direction and normal direction near the junctions. Therefore, the way to separate the stresses and energy into surface part and normal part is not suitable for the domains near the junctions. In addition, due to the stress concentration near the junctions, the stress distribution there is more complicated than that in the area far away from the

junctions. At this part, the structure behaves more like a general 3D structure rather than a shell structure. Hence, in this study, the 3D elements near the junctions will be treated as general solid elements and the standard SPR technique will be applied. According to this treatment, a TWS with surface intersections will be divided into the *junction zone* and the *shell zone*.

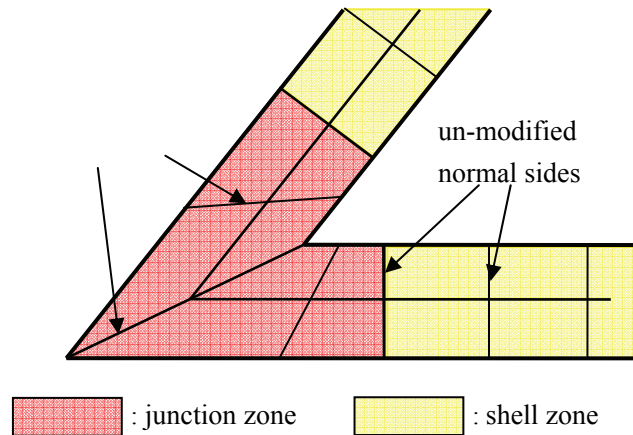


Figure 5.21 Junction zone and shell zone of a TWS with surface intersections

During the mesh generation procedure, some normal sides along the surface intersections or near the junctions have been shifted from their original positions (Section 3.3). For the volume elements that consist of the modified normal sides, they are named as the *junction volume elements*, all the other volume elements are named as the *shell volume elements*. The solid elements that are divided from the junction volume elements will be set to the junction zone while those elements divided from the shell volume elements are set to the shell zone (Figure 5.21).

In the process of element patch formation, if within the patch there is an element from the junction zone, then this patch will be defined as a *junction patch* (Figure 5.22a). While for an element patch that all of its included elements are within the shell zone, it will be defined as a *shell patch* (Figure 5.22b). For the junction patch, the standard SPR scheme is applied and the polynomial terms used

for this patch are the standard polynomial terms rather than the reduced ones shown in Section 5.3.2.1. In a different way, for the shell patch, the modified SPR technique (SPR1) given in Section 5.3.2.2 will be used.

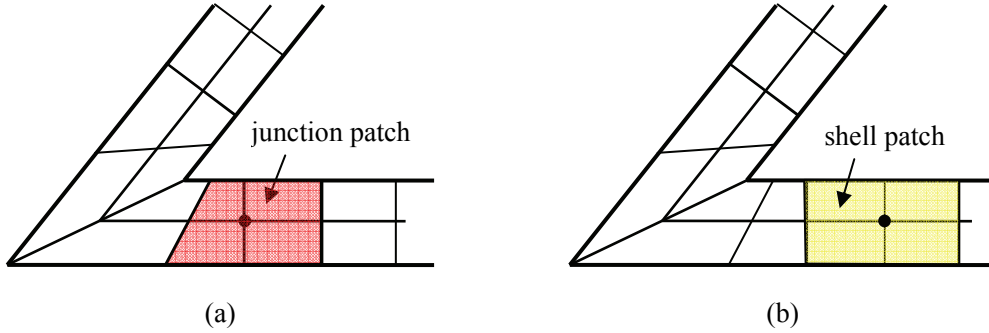


Figure 5.22 Junction patch and shell patch in a structure shown in Figure 5.21

5.4 *A Priori* Error Estimation for TWSs

In Chapter 2, it has been introduced that the FEM error is bounded by a function related to the element size and the highest order of polynomials for interpolation. For TWSs, since the energy norm is divided into the surface part and normal part for separate analysis in this current study, the error norm should also be separated accordingly by

$$\|\mathbf{e}\|_{\Omega} = \sqrt{\|_{\text{surf}}\|\mathbf{e}\|_{\Omega}^2 + \|_{\text{norm}}\|\mathbf{e}\|_{\Omega}^2} \quad (5.65)$$

where

$$\|_{\text{surf}}\|\mathbf{e}\|_{\Omega} = \sqrt{\int_{\Omega} (\|_{\text{surf}}\boldsymbol{\sigma} - \|_{\text{surf}}\hat{\boldsymbol{\sigma}})^T \|_{\text{surf}}\mathbf{D}^{-1} (\|_{\text{surf}}\boldsymbol{\sigma} - \|_{\text{surf}}\hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.66)$$

and

$$\|_{\text{norm}}\|\mathbf{e}\|_{\Omega} = \sqrt{\int_{\Omega} (\|_{\text{norm}}\boldsymbol{\sigma} - \|_{\text{norm}}\hat{\boldsymbol{\sigma}})^T \|_{\text{norm}}\mathbf{D}^{-1} (\|_{\text{norm}}\boldsymbol{\sigma} - \|_{\text{norm}}\hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.67)$$

where $\|_{\text{surf}}\|\mathbf{e}\|_{\Omega}$ is the surface part of error norm and $\|_{\text{norm}}\|\mathbf{e}\|_{\Omega}$ is the normal part of error norm. $\|_{\text{surf}}\hat{\boldsymbol{\sigma}}$ and $\|_{\text{norm}}\hat{\boldsymbol{\sigma}}$ are the corresponding FE solutions of $\|_{\text{surf}}\boldsymbol{\sigma}$ and $\|_{\text{norm}}\boldsymbol{\sigma}$. $\|_{\text{surf}}\mathbf{D}$ and $\|_{\text{norm}}\mathbf{D}$ are the partitioned parts of matrix $\mathbf{D}$ (Equation (5.40)).

With the above definitions, the *a priori* estimate for TWSs can be expressed by

$$\|\mathbf{e}\|_{\Omega} \leq Ch^{\rho} \quad (5.68)$$

where C is a constant and h is the element size. ρ is the convergence rate, which is determined by

$$\rho = \min(\lambda, p) \quad (5.69)$$

where λ (<1.0) is the strength of singularity [120], which characterizes the smoothness of the exact solution of the problem. p is the order of the element shape functions. If ND is the dimension of the problem (ND = 2 for 2D problems and so on), then the relationship between h and NTD, the total DOFs of the mesh, can be written as

$$\text{NTD} \propto h^{-\text{ND}} \quad (5.70)$$

Thus, Equation (5.68) can be written as

$$\|\mathbf{e}\|_{\Omega} \leq C'(\text{NTD})^{-\rho/\text{ND}} \quad (5.71)$$

where C' is another constant different from C . In the current study, because 3D solid elements are used for the analysis of TWSs, ND = 3 in Equation (5.71). In order to increase the convergence rate of the solution for the h -refinement, the finite element mesh should be constructed in such a way that the error is equally distributed among all the elements in the mesh. With such an optimal mesh, the effect of singularities will be largely eliminated and hence the convergence rate is just set to be p .

5.5 *A Posteriori* Estimation for TWSs

Different from the *a priori* error estimation, which only shows a theoretic bound of the overall FEM error and the convergence rate before the implementation of the FE analysis, the *a posteriori* error estimation is made after the FE solutions are generated, which can approximately reflect the error in different local positions by detailed values. In this study, *a posteriori* error estimation scheme for TWSs is designed based on the error estimator proposed by Zienkiewicz and Zhu [10].

5.5.1 A brief review on Z-Z error estimator

Among all the *a posteriori* error estimators, the one proposed by Zienkiewicz and Zhu [73] is the most widely used one. The essence of Zienkiewicz-Zhu estimator (also referred as Z-Z or Z^2 error estimator) is to use the recovered solution in place of the exact solution in the computation of the error estimates. Hence, the local error $\bar{\mathbf{e}}_\sigma$ of the FE stress $\hat{\boldsymbol{\sigma}}$ can be calculated by

$$\bar{\mathbf{e}}_\sigma = \tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}} \quad (5.72)$$

where $\tilde{\boldsymbol{\sigma}}$ is the recovered stress. The main advantages of Z-Z error estimator are the simplicity of its implementation and its cost effectiveness.

The overall error norm defined by Z-Z error estimator is written as

$$\|\bar{\mathbf{e}}\| = \sqrt{\int_{\Omega} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.73)$$

where Ω is the domain on which the problem is defined and $\mathbf{D}$ is the elasticity matrix. The error norm can also be calculated within each element by

$$\|\bar{\mathbf{e}}\|_e = \sqrt{\int_{\Omega_e} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.74)$$

The relationship between the overall error norm and the element error norm is

$$\|\bar{\mathbf{e}}\| = \sqrt{\sum_{e=1}^n (\|\bar{\mathbf{e}}\|_e)^2} \quad (5.75)$$

where n is the total number of elements in the mesh.

If the exact solutions are available, then the exact error norm is

$$\|\mathbf{e}\| = \sqrt{\int_{\Omega} (\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) d\Omega} \quad (5.76)$$

where $\boldsymbol{\sigma}$ is the exact stress vector. The error norm of the recovered stresses can be calculated by

$$\|\tilde{\mathbf{e}}\| = \sqrt{\int_{\Omega} (\boldsymbol{\sigma} - \tilde{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} (\boldsymbol{\sigma} - \tilde{\boldsymbol{\sigma}}) d\Omega} \quad (5.77)$$

Zienkiewicz and Zhu designed a variable, named as the *effectivity index*, to show

the quality of the error estimator. The effectivity index is calculated by

$$\theta = \frac{\|\tilde{e}\|}{\|e\|} \quad (5.78)$$

It can be deduced that the effectivity index θ is restricted by a range shown as

$$1 - \frac{\|\tilde{e}\|}{\|e\|} \leq \theta \leq 1 + \frac{\|\tilde{e}\|}{\|e\|} \quad (5.79)$$

As the mesh is refined, $\|\tilde{e}\|$ will tend to zero so that θ will converge to unity as well. The effectivity index value can be used to judge the performance of the error estimator i.e. the estimated error is more reliable when the θ value is closer to 1.

5.5.2 Error estimation for TWSs

After all the nodal stresses have been recovered, the error norm defined by Z-Z error estimator (Equation (5.74)) will be calculated by integrating the error over the whole volume of each element. In order to obtain more accurate results, higher order Gaussian integration points will be used for the numerical integration within every solid element. In the current study, 4th order Gaussian integration points are employed for all the quadratic solid elements. Table 5.5 shows the *number of integration points* (NGP) used for the calculation of energy norm for all the solid elements used in this study. For comparison, NGP used for the formation of element stiffness and the number of sampling points used for stress recovery are also shown in Table 5.5.

Element type	T10	P13	P15	H20
Stiffness Matrix formation	6	27	9	27
Stress Recovery	4	8	6	8
Energy Norm Calculation	12	64	30	125

Table 5.5 Number of integration points used for the quadratic solid elements

For the elements from the junction zone, the standard 3D Z-Z error estimator

will be used and no partition of the energy or error norm is required. While for the elements from the shell zone, since the recovery scheme is based on SPR1 recovery technique (Section 5.3.2.2), the Z-Z error estimator will be amended accordingly due to the energy separation.

5.5.2.1 Conjoint polynomial interpolation

Note that in the SPR technique, Equation (5.42) is only used for the calculation of the nodal stresses. For the stresses at any position within the element, Zienkiewicz and Zhu [10] suggested using the nodal interpolation method to calculate them. Therefore, the smoothed stress of any point A within an element can be obtained by

$$\tilde{\sigma}^A = \sum_{i=1}^n N_i \cdot \tilde{\sigma}_i \quad (5.80)$$

where N_i is the shape function belong to the i th corner node at this point; $\tilde{\sigma}_i$ is the nodal recovered stress at the i th corner node of this element; n is the number of corner nodes in this element. Note that for quadratic solid elements, since the nodal stresses at the mid-side nodes are not recovered from the SPR technique, the shape functions used in Equation (5.80) will be the functions used for the corresponding linear solid elements.

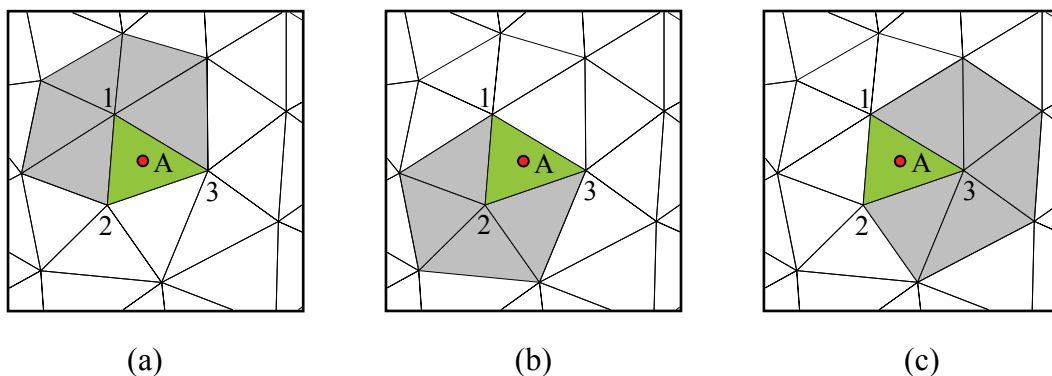


Figure 5.23 Conjoint polynomial interpolation at point A:
(a) patch for node 1; (b) patch for node 2 and (c) patch for node 3

It has been found that the smoothed stresses obtained by Equation (5.80) are not that accurate because the accuracy of the recovered nodal stresses is usually less than the accuracy of the stresses elsewhere in the patch [121]. To resolve this problem, an enhanced algorithm called *conjoint polynomial interpolation* [121] has been proposed. In this algorithm, the nodal recovered stresses, $\tilde{\sigma}_i$ in Equation (5.80) will be replaced by $\tilde{\sigma}_i^A$, the recovered stresses of point A in the patch that is assembled at the i th element node (Figure 5.23). $\tilde{\sigma}_i^A$ is calculated by

$$\tilde{\sigma}_i^A = \mathbf{P}_i^A \mathbf{a}_i \quad (5.81)$$

Hence, Equation (5.80) is now rewritten as

$$\tilde{\sigma}^A = \sum_{i=1}^n N_i \cdot \tilde{\sigma}_i^A \quad (5.82)$$

The conjoint polynomial interpolation significantly improves the results compared to the standard nodal interpolation, hence the resulted error estimator is more reliable. In the current study, conjoint polynomial interpolation will be applied to all the solid elements, either the ones from the junction zone or those from the shell zone.

5.5.2.2 Estimated error for TWSs

According to the stress recovery scheme proposed in Section 5.3, the estimated error of TWSs will be divided into two parts: error of the junction zone and error of the shell zone. In addition, according to the stresses separation scheme, the error of the shell zone will be further divided into the shell surface error and shell normal error. For the estimated error norms of the junction zone, they can be calculated by Equations (5.74) and (5.75). For the estimated error norms of the shell zone, since the recovered stresses can be partitioned into the surface part and normal part

Stress recovery and error estimation

$$\tilde{\boldsymbol{\sigma}} = \left[\left(\text{surf} \tilde{\boldsymbol{\sigma}} \right)^T, \left(\text{norm} \tilde{\boldsymbol{\sigma}} \right)^T \right]^T \quad (5.83)$$

then in each element, the norm of the estimate error is

$$\|\bar{\mathbf{e}}\|_e = \sqrt{\text{surf} \|\bar{\mathbf{e}}\|_e^2 + \text{norm} \|\bar{\mathbf{e}}\|_e^2} \quad (5.84)$$

where

$$\begin{cases} \text{surf} \|\bar{\mathbf{e}}\|_e = \sqrt{\int_{\Omega_e} (\text{surf} \tilde{\boldsymbol{\sigma}} - \text{surf} \hat{\boldsymbol{\sigma}})^T \text{surf} \mathbf{D}^{-1} (\text{surf} \tilde{\boldsymbol{\sigma}} - \text{surf} \hat{\boldsymbol{\sigma}}) d\Omega} \\ \text{norm} \|\bar{\mathbf{e}}\|_e = \sqrt{\int_{\Omega_e} (\text{norm} \tilde{\boldsymbol{\sigma}} - \text{norm} \hat{\boldsymbol{\sigma}})^T \text{norm} \mathbf{D}^{-1} (\text{norm} \tilde{\boldsymbol{\sigma}} - \text{norm} \hat{\boldsymbol{\sigma}}) d\Omega} \end{cases} \quad (5.85)$$

which are the norms of the estimated element error in surface direction and normal direction, respectively. The norm of the estimated error for the whole shell zone is

$$\|\bar{\mathbf{e}}\|_{\Omega} = \sqrt{\text{surf} \|\bar{\mathbf{e}}\|_{\Omega}^2 + \text{norm} \|\bar{\mathbf{e}}\|_{\Omega}^2} \quad (5.86)$$

where

$$\begin{cases} \text{surf} \|\bar{\mathbf{e}}\|_{\Omega} = \sqrt{\sum \text{surf} \|\bar{\mathbf{e}}\|_e^2} \\ \text{norm} \|\bar{\mathbf{e}}\|_{\Omega} = \sqrt{\sum \text{norm} \|\bar{\mathbf{e}}\|_e^2} \end{cases} \quad (5.87)$$

which are the norms of the estimated error for the whole shell zone in surface direction and normal direction, respectively.

For the junction zone, the effectivity indexes can be calculated by Equation (5.78) directly. For the shell zone, the effectivity index for an element will be divided into the surface direction effectivity index and the norm direction effectivity index, which are written as

$$\text{surf} \theta_e = \frac{\text{surf} \|\bar{\mathbf{e}}\|_e}{\text{surf} \|\mathbf{e}\|_e} \quad \text{and} \quad \text{norm} \theta_e = \frac{\text{norm} \|\bar{\mathbf{e}}\|_e}{\text{norm} \|\mathbf{e}\|_e} \quad (5.88)$$

and the overall effectivity index for this element is

$$\theta_e = \frac{\|\bar{\mathbf{e}}\|_e}{\|\mathbf{e}\|_e} = \frac{\sqrt{\text{surf} \|\bar{\mathbf{e}}\|_e^2 + \text{norm} \|\bar{\mathbf{e}}\|_e^2}}{\sqrt{\text{surf} \|\mathbf{e}\|_e^2 + \text{norm} \|\mathbf{e}\|_e^2}} \quad (5.89)$$

Similarly, the effectivity indexes for the whole shell zone are expressed by

Stress recovery and error estimation

$$\theta_{\Omega}^{\text{surf}} = \frac{\|\bar{\mathbf{e}}\|_{\Omega}}{\|\mathbf{e}\|_{\Omega}}, \quad \theta_{\Omega}^{\text{norm}} = \frac{\|\bar{\mathbf{e}}\|_{\Omega}}{\|\mathbf{e}\|_{\Omega}} \quad \text{and} \quad \theta_{\Omega} = \frac{\|\bar{\mathbf{e}}\|_{\Omega}}{\|\mathbf{e}\|_{\Omega}} \quad (5.90)$$

A summary of the equations used for the junction zone and shell zone in *a posteriori* error estimation is given in Table 5.6.

Equation	Junction zone	Shell zone
Error	(5.74), (5.75)	(5.84) to (5.87)
Effectivity index	(5.78)	(5.90)

Table 5.6 Equations in *a posteriori* error estimation for junction zone and shell zone

5.6 Error Estimation Examples

In this section, two simple examples will be provided to test the effect of the designed error estimation scheme. Exact solutions are available for both these two examples.

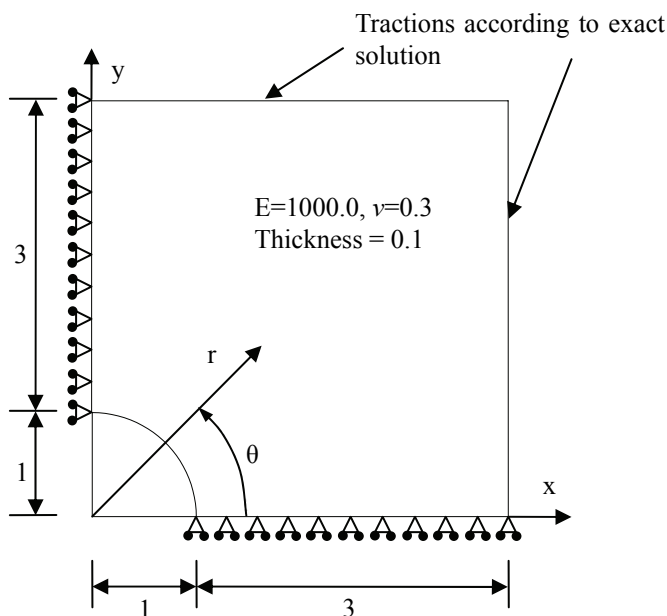


Figure 5.24 Example 5.1: plate with central circular hole

Example 5.1 is a classic problem which has been used by many researchers for 2D adaptive mesh refinement schemes. The problem domain is a portion of an infinite plate with a circular hole subjected to unidirectional tensile load. The radius of the hole is equal to 1 unit and a 4×4 portion of the plate is considered (Figure 5.24). The boundary conditions prescribed are such that the symmetry boundary conditions are imposed (Figure 5.24). The exact solution for this problem is smooth everywhere inside the problem domain and is given by

$$\begin{aligned}\sigma_x &= 1 - \frac{1}{r^2} \left(\frac{3}{2} \cos(2\theta) + \cos(4\theta) \right) + \frac{3}{2r^4} \cos(4\theta) \\ \sigma_y &= -\frac{1}{r^2} \left(\frac{1}{2} \cos(2\theta) - \cos(4\theta) \right) - \frac{3}{2r^4} \cos(4\theta) \\ \tau_{xy} &= -\frac{1}{r^2} \left(\frac{1}{2} \sin(2\theta) + \sin(4\theta) \right) + \frac{3}{2r^4} \cos(4\theta)\end{aligned}\tag{5.91}$$

The loadings applied are the tractions correspond to the exact solution for the infinite plate applied on the boundaries. With given plate thickness, this problem can also be used for the 3D case. The thickness of the plate is set to 0.1. Since this example is nearly a 2D problem, the stress components involved in normal direction are trivial (the ratio of the stresses between normal direction and surface direction is less than $1\text{E-}10$) so that only one layer of elements is formed for all the meshes in the refinement steps. Two kinds of input surface elements are used for this example, the first one is triangular element and the second one is quadrilateral element. For each kind of input surface element, three uniform meshes are generated for the analysis (Figure 5.25 and Figure 5.26). As in this example, the structure is a plate so that its normal direction does not change at different locations. Therefore, the recovered stresses obtained by the SPR1 technique will have little difference from the ones obtained by the standard SPR technique.

Stress recovery and error estimation

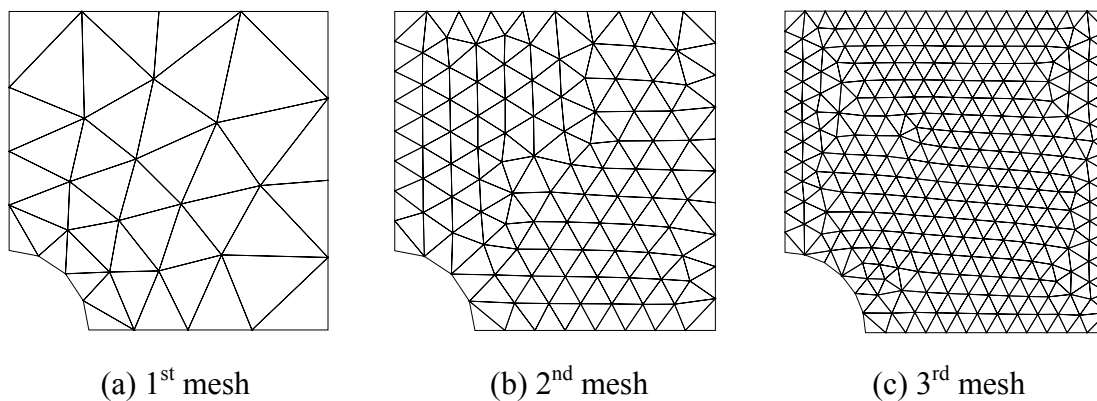


Figure 5.25 Example 5.1: meshes for triangular input elements

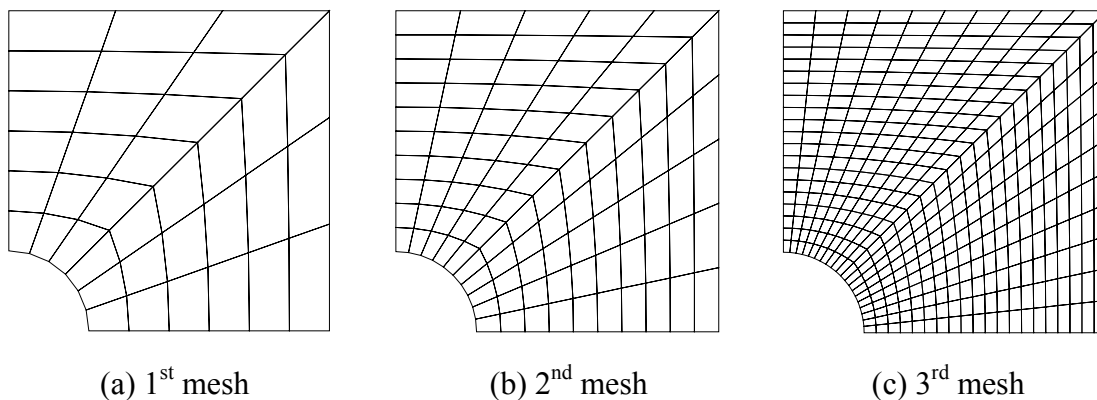


Figure 5.26 Example 5.1: meshes for quadrilateral input elements

Mesh	NTD	$\ e\ $		η		R		θ
		FEM	SPR	FEM	SPR	FEM	SPR	
1	726	1.6750E-03	2.0437E-03	4.08%	4.97%	/	/	1.0873
2	3009	1.2934E-03	1.2767E-03	3.15%	3.11%	0.1819	0.3309	0.8068
3	7656	5.6094E-04	4.4545E-04	1.36%	1.08%	0.8946	1.1275	0.8767

Table 5.7 Example 5.1: Results of triangular input elements

Mesh	NTD	$\ e\ $		η		R		θ
		FEM	SPR	FEM	SPR	FEM	SPR	
1	945	1.0984E-03	1.1339E-03	2.67%	2.76%	/	/	0.8740
2	2409	4.7427E-04	3.2547E-04	1.15%	0.79%	0.8975	1.3338	0.8502
3	9009	1.3425E-04	5.2453E-05	0.33%	0.13%	0.9568	1.3839	0.9363

Table 5.8 Example 5.1: Results of quadrilateral input elements

Some results of this example are shown in Table 5.7 and Table 5.8. Since this problem is almost a 2D example so that the stress components in plate normal direction are close to zero, therefore, in this example, only the total error norms are shown in Table 5.7 and Table 5.8. In each table, error norm $\|\mathbf{e}\|$ consists of two columns of data. Data in FEM column are the norms of the total FEM error of the whole domain (Equation (5.76)) and data in SPR column are the norms of the total recovered error of the whole domain (Equation (5.77)). η is the relative error, which is the ratio between $\|\mathbf{e}\|$ and $\|\mathbf{u}\|$ (Equations (5.43) and (5.44)) written in percentage format. For a series of consecutive meshes in the mesh refinement procedure, the convergence rate, R , can be calculated by

$$R_i = \frac{\log(\eta_i) - \log(\eta_{i-1})}{\log(\text{DOF}_{i-1}) - \log(\text{DOF}_i)} \quad (5.92)$$

where the variable with a subscript of i means the value of this variable is obtained from the i th mesh. θ is the overall effectivity index.

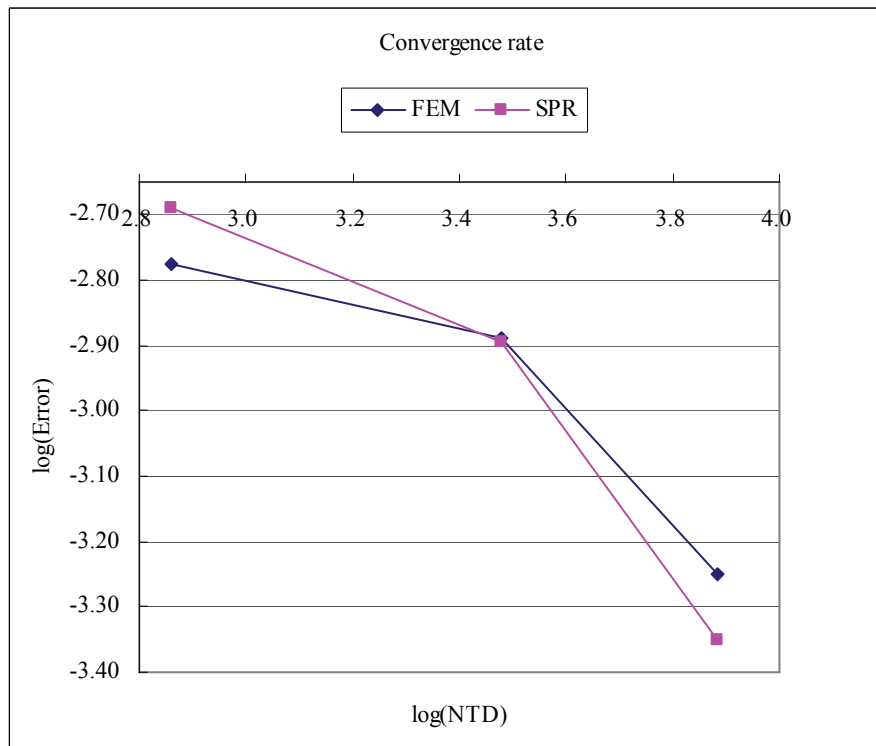


Figure 5.27 Example 5.1: convergence rate for triangular input elements

Stress recovery and error estimation

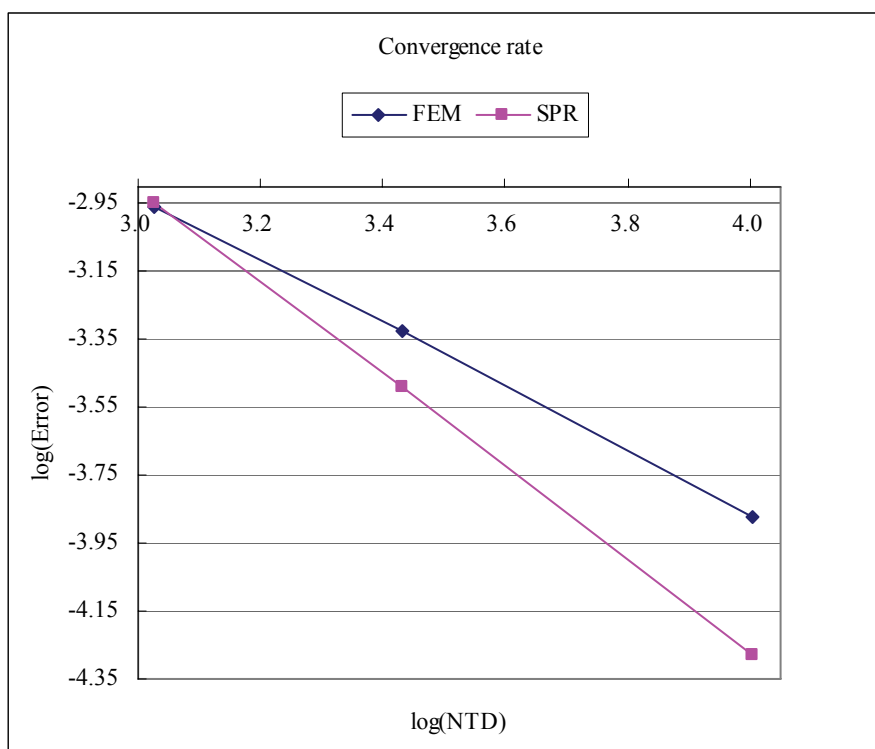


Figure 5.28 Example 5.1: convergence rate for quadrilateral input elements

From the results shown in Table 5.7 and Table 5.8, it can be seen that the recovered stresses are more accurate than the FE stresses. In addition, during the mesh refinement process, the recovered stress obtained by SPR technique can achieve a higher convergence rate than the FE stresses (Figure 5.27 and Figure 5.28).

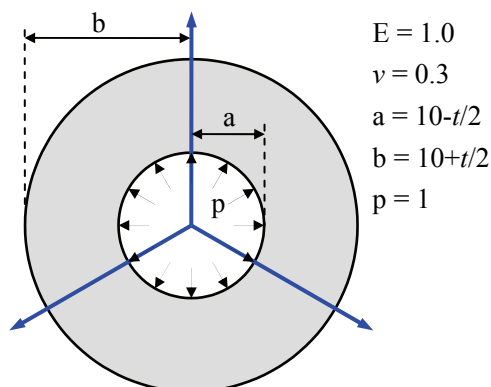


Figure 5.29 Example 5.2: spherical container subjected to uniform internal pressure

Stress recovery and error estimation

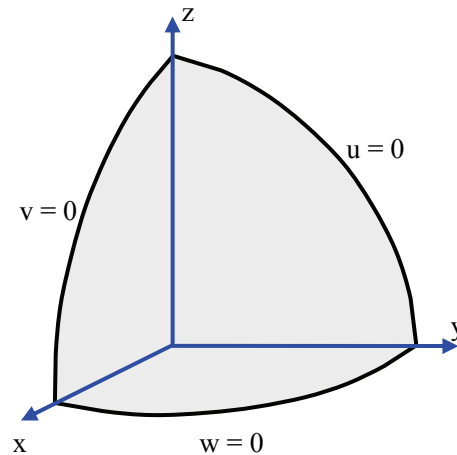


Figure 5.30 Example 5.2: boundary conditions for one-eighth of the shell

Example 5.2 is a sphere shell with an average radius of 10 and a shell thickness of t (Figure 5.29). Three different values of t ($t = 2$, $t = 1$ and $t = 0.5$) are tested for this example. The shell surface in this example is curved so that the shell normal direction is changed everywhere. Uniform unit pressure is prescribed on the internal surface and the value is set to 1. Due to symmetry, only one-eighth of the container was modelled in the finite element analysis. The boundary condition of this example is shown in Figure 5.30. The exact solution for this problem can be found in reference [122]. For each case of thickness, uniform refinements were carried out and the three finite element meshes are generated (Figure 5.31, Figure 5.32 and Figure 5.33). For every mesh, both conventional 3D SPR technique which is based on the GCS and the modified SPR which has been presented in Section 5.3.2 are applied. Note that this example is a truly 3D problem, hence only one layer of elements is not adequate to catch the stress variation along the normal direction. In this example, the number of layers of elements is set to 8 for all the meshes in the refinement steps.

Stress recovery and error estimation

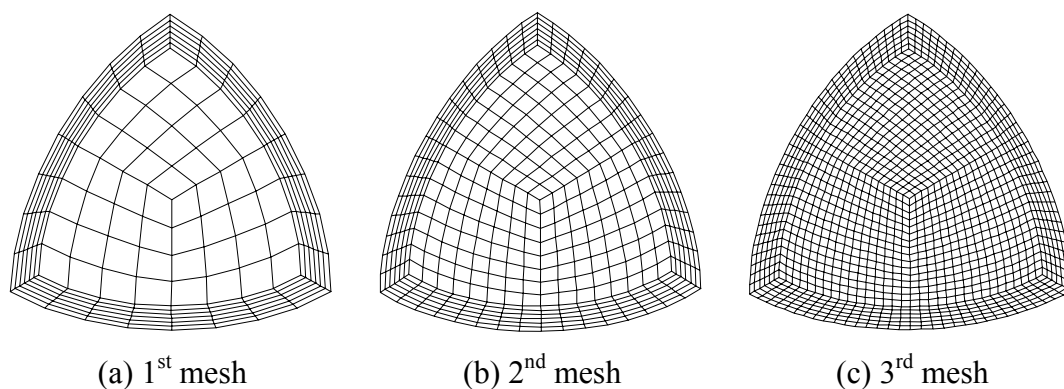


Figure 5.31 Example 5.2: meshes of refinement for case $t = 2$

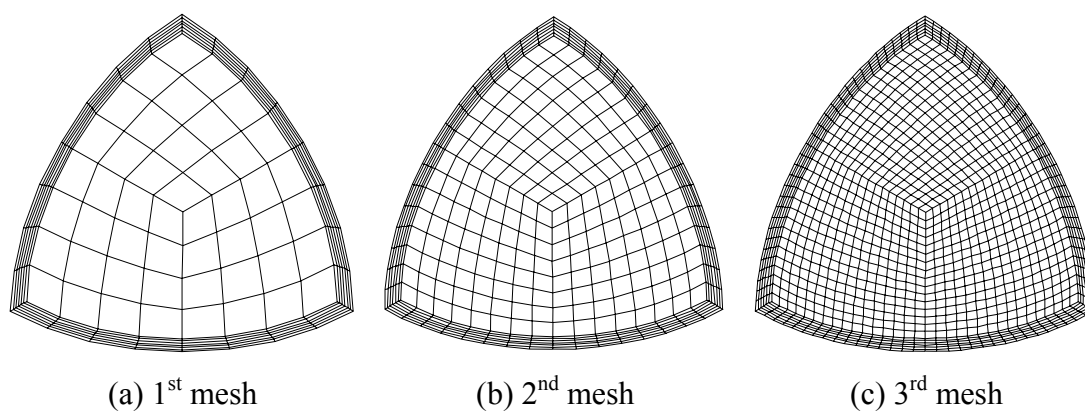


Figure 5.32 Example 5.2: meshes of refinement for case $t = 1$

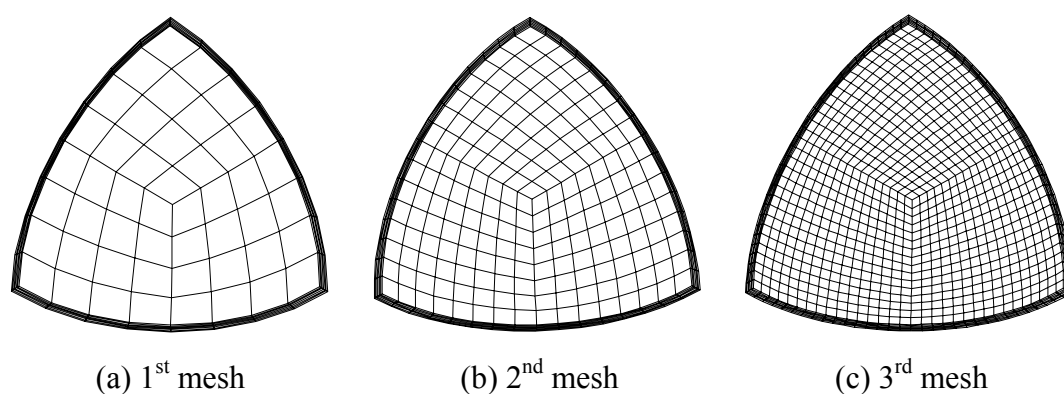


Figure 5.33 Example 5.2: meshes of refinement for case $t = 0.5$

Stress recovery and error estimation

Mesh	NTD	SPR scheme		$\ e\ $		θ	η	
				FEM	Recovered		FEM	Recovered
1	4647	SPR-STD		1.325E-01	1.062E-01	1.1495	28.24%	22.64%
		SPR-TWS	Total	1.325E-01	3.691E-02	0.9268	28.24%	7.87%
			Surface	9.112E-02	2.880E-02	0.9290	19.43%	6.14%
			Normal	9.615E-02	2.308E-02	0.9248	20.50%	4.92%
2	17031	SPR- STD		3.339E-02	1.580E-02	1.0663	7.12%	3.37%
		SPR-TWS	Total	3.339E-02	5.175E-03	0.9732	7.12%	1.10%
			Surface	1.973E-02	4.228E-03	0.9651	4.21%	0.90%
			Normal	2.695E-02	2.983E-03	0.9775	5.74%	0.64%
3	65127	SPR- STD		2.309E-02	2.863E-03	1.0054	4.92%	0.61%
		SPR-TWS	Total	2.309E-02	1.825E-03	0.9923	4.92%	0.39%
			Surface	3.711E-03	1.142E-03	1.1446	0.79%	0.24%
			Normal	2.279E-02	1.424E-03	0.9879	4.86%	0.30%

Table 5.9 Example 5.2: results for case $t = 2$

Mesh	NTD	SPR scheme		$\ e\ $		θ	η	
				FEM	Recovered		FEM	Recovered
1	4647	SPR-STD		1.523E-01	1.048E-01	0.9993	21.72%	14.95%
		SPR-TWS	Total	1.523E-01	3.587E-02	0.9357	21.72%	5.12%
			Surface	9.904E-02	2.407E-02	0.9559	14.13%	3.43%
			Normal	1.157E-01	2.659E-02	0.9206	16.50%	3.79%
2	17031	SPR- STD		3.256E-02	1.899E-02	1.0496	4.65%	2.71%
		SPR-TWS	Total	3.256E-02	5.964E-03	0.9577	4.65%	0.85%
			Surface	2.390E-02	5.000E-03	0.9543	3.41%	0.71%
			Normal	2.212E-02	3.250E-03	0.9616	3.16%	0.46%
3	65127	SPR- STD		9.944E-03	2.485E-03	1.0120	1.42%	0.35%
		SPR-TWS	Total	9.944E-03	9.462E-04	0.9885	1.42%	0.13%
			Surface	5.212E-03	8.261E-04	0.9752	0.74%	0.12%
			Normal	8.469E-03	4.613E-04	0.9934	1.21%	0.07%

Table 5.10 Example 5.2: results for case $t = 1$

Stress recovery and error estimation

Mesh	NTD	SPR scheme		$\ e\ $		θ	η	
				FEM	Recovered		FEM	Recovered
1	4647	SPR-STD		1.676E-01	1.337E-01	0.9698	16.44%	13.11%
		SPR-TWS	Total	1.676E-01	6.034E-02	0.8682	16.44%	5.92%
			Surface	1.063E-01	4.159E-02	0.8555	10.43%	4.08%
			Normal	1.295E-01	4.372E-02	0.8766	12.71%	4.29%
2	17031	SPR- STD		3.577E-02	2.110E-02	1.0003	3.51%	2.07%
		SPR-TWS	Total	3.577E-02	5.697E-03	0.9641	3.51%	0.56%
			Surface	2.483E-02	4.204E-03	0.9742	2.44%	0.41%
			Normal	2.575E-02	3.845E-03	0.9546	2.53%	0.38%
3	65127	SPR- STD		8.099E-03	3.001E-03	0.9904	0.79%	0.29%
		SPR-TWS	Total	8.099E-03	1.059E-03	0.9756	0.79%	0.10%
			Surface	6.067E-03	9.362E-04	0.9705	0.60%	0.09%
			Normal	5.366E-03	4.961E-04	0.9822	0.53%	0.05%

Table 5.11 Example 5.2: results for case $t = 0.5$

Mesh	Thickness	SPR-STD	SPR-TWS	Improvement
		η_1	η_2	$(\eta_1 - \eta_2)/\eta_1$
1 st mesh	$t = 2$	22.64%	78.70%	65.24%
	$t = 1$	14.95%	5.12%	65.75%
	$t = 0.5$	13.11%	5.92%	54.84%
2 nd mesh	$t = 2$	3.37%	1.10%	67.36%
	$t = 1$	2.71%	0.85%	68.63%
	$t = 0.5$	2.07%	0.56%	72.95%
3 rd mesh	$t = 2$	0.61%	0.39%	36.07%
	$t = 1$	0.35%	0.13%	62.86%
	$t = 0.5$	0.29%	0.10%	65.52%

Table 5.12 Example 5.2: comparison of different cases of thickness

The results for different cases of thickness are summarized in Table 5.9, Table 5.10 and Table 5.11. In above tables, SPR-STD refers to the standard super patch recovery method and SPR-TWS refers to the modified SPR (Section 5.3.2). From the results shown in above tables, it is clear that the error of the recovered stresses is less than the error of the FE solution. Furthermore, by comparing the results

Stress recovery and error estimation

obtained from SPR-STD and SPR-TWS, it is found that the SPR-TWS scheme performs better than the SPR-STD technique because it can provide smaller error and achieve better effectivity index value. This also proves that the proposed stress recovery scheme is more suitable for the analysis of TWSs. By comparing the results of different cases of thickness (Table 5.12), it can be found that when the shell thickness is becoming smaller, the improvement of SPR-TWS over SPR-STD is more obvious. This is because the error caused by reduction of polynomial terms (Section 5.3.2.1) is less when the structure is becoming thinner.

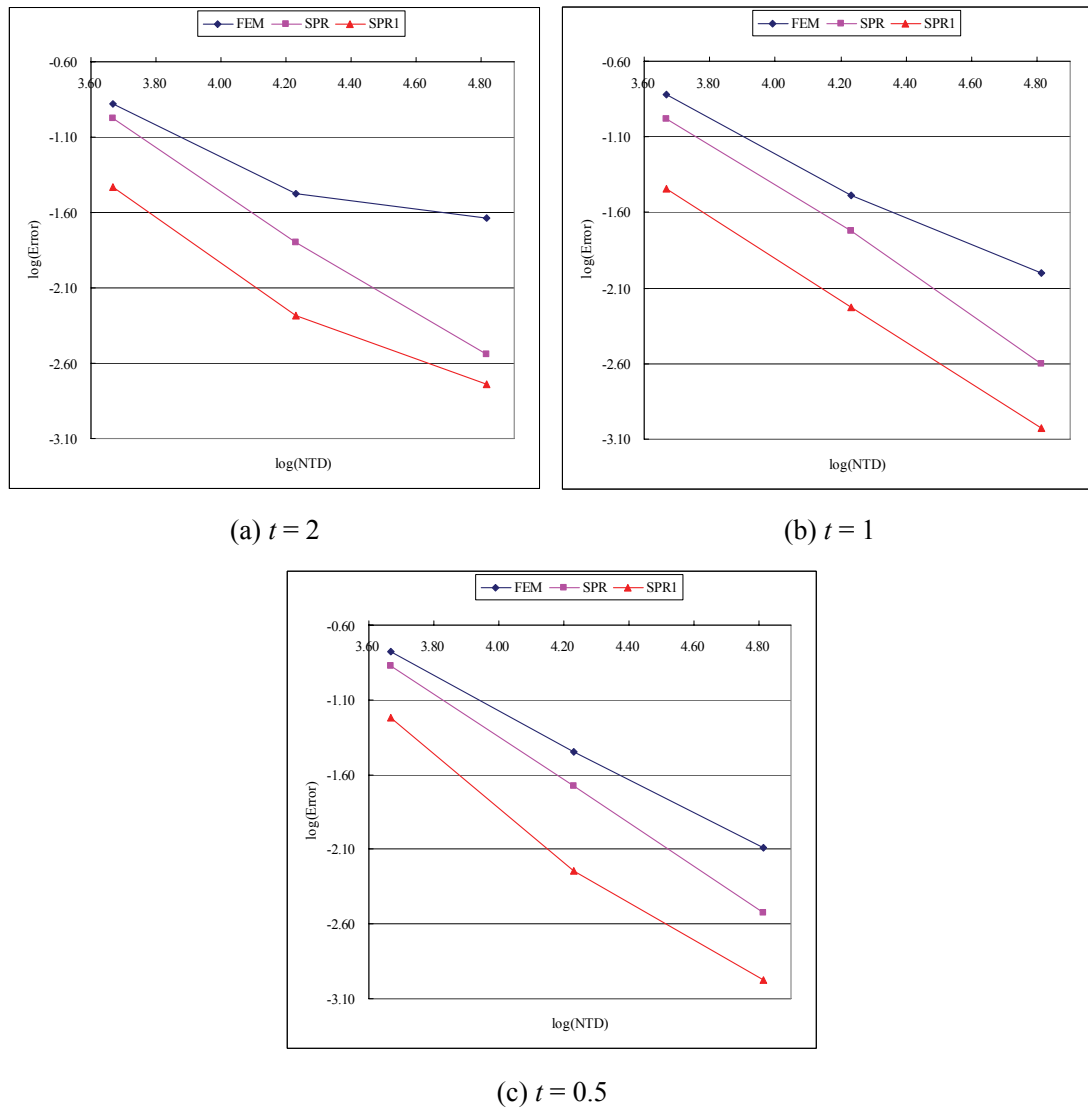


Figure 5.34 Example 5.2: comparison of the convergence rate

From the curves shown in Figure 5.34, it can be seen that error of the recovered stresses converges faster than the one from FE solution when the mesh is refined. And since the SPR-STD technique and the SPR-TWS technique are both derived from the stresses at the superconvergent points, their convergence rates are similar.

5.7 Closure

This chapter is mainly devoted to the design of algorithms for the error estimation of TWSs. In the *a priori* error estimation part, it is pointed out that for TWSs, the error can be estimated in surface direction and normal direction separately. Feasible way to separate the stresses and energy is given. After that, it comes to the major parts of this chapter, stress recovery and *a posteriori* error estimation for TWSs. In the stress recovery procedure, a special scheme which is used for TWSs only is designed based on the SPR technique. Coordinate systems transformation and the boundary patch enhancement are the emphases of this scheme. For the TWS that is constructed by intersected surfaces, it is suggested to divide the whole structure into the junction zone and the shell zone for different analysis. Once the stresses are recovered, the approximated error can be calculated by the Z-Z error estimator. Similar to the stress recovery process, the standard Z-Z error estimator is adjusted to conform to the energy separation scheme.

Two simple examples are provided in the end of this chapter to demonstrate the effect of the proposed error estimation schemes. More examples and more complicated problems will be presented in the next chapter, the design of the adaptive refinement procedure.

CHAPTER 6

ADAPTIVE REFINEMENT

In Chapter 5, the stress recovery algorithm and the corresponding *a posteriori* error estimator for TWSs have been developed. In this chapter, an adaptive refinement scheme will be designed based on this error estimator. Only *h*-refinement is applied in the current study, the orders of elements are kept unchanged. Hence, the implementation of mesh refinement is actually a process of mesh regeneration. For TWS, emphasis of the adaptive refinement scheme is to determine the element sizes in the surface direction and normal direction, and the mesh is refined along surface direction and normal direction separately.

6.1 Overview of the General Adaptive Refinement Scheme

The aim of the *h*-version adaptive refinement is to determine the element sizes for the next refined mesh so that the maximum possible convergence rate can be achieved. The mesh will be refined until the target accuracy is achieved. That is,

$$\bar{\eta} \leq \epsilon \eta \quad (6.1)$$

where $\bar{\eta}$ is the estimated relative error of the current solution and $\epsilon \eta$ is the prescribed relative error tolerance. From Equation (6.1) it can be seen that a *a posteriori* error estimator is required for the adaptive refinement scheme, the main application of the *a posteriori* error estimator is to provide guidelines for the determination of proper element sizes and nodal density for the next refinement. Hence, the *a posteriori* error estimation and adaptive refinement are closely related

Adaptive refinement

and often discussed together in many papers [9, 112, 123-125].

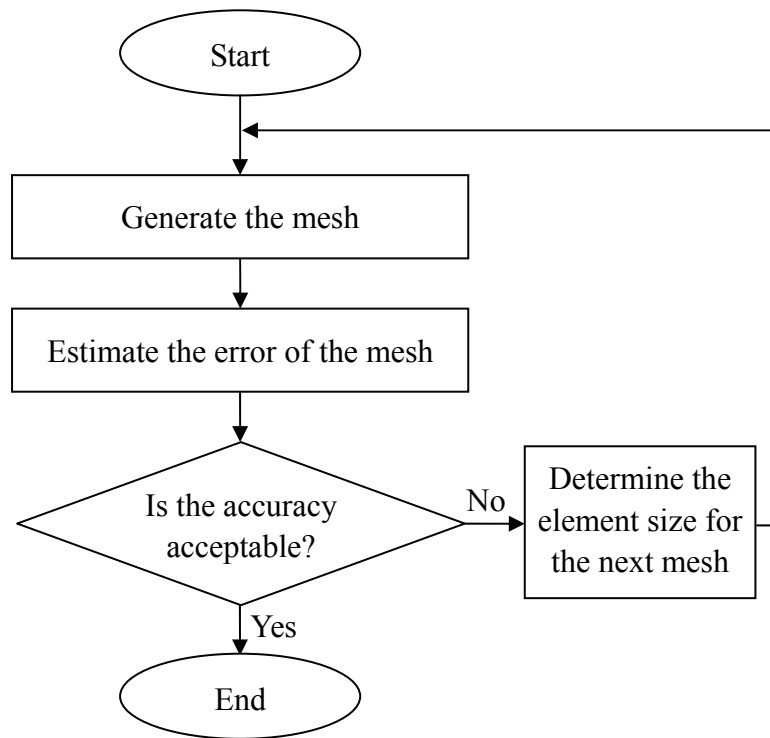


Figure 6.1 Main steps in the adaptive refinement scheme

The main steps in the adaptive refinement scheme are shown in Figure 6.1. Once the current relative error, $\bar{\eta}$ is calculated by the error estimator, the element size for the next refined mesh can be determined by the following steps:

- i) Estimate the expected number of elements required to achieve the target error

$$NE_e = NE_c \left(\bar{\eta} / {}^t\eta \right)^{d/p} \quad (6.2)$$

where NE_c is the current number of elements in the mesh and NE_e is the the expected number of elements for the next mesh. d is the dimension of the domain and p is the order of the element used.

- ii) Define the local allowable total error norm e_a and local refinement indicator ς_i for the i th element

$$e_a = {}^t\eta \| \mathbf{u} \|_{\Omega} / \sqrt{NE_e} \quad \text{and} \quad \varsigma_i = \| \bar{\mathbf{e}}_u \|_{\Omega_i} / e_a \quad (6.3)$$

Adaptive refinement

where $\|\bar{\mathbf{e}}_{\mathbf{u}}\|_{\Omega_i}$ is the estimated error norm of the i th element.

iii) Determine the new element size h_i for the next cycle of analysis

$$h_i^{\text{new}} = \frac{h_i^{\text{old}}}{(\zeta_i)^{1/(\rho+1)}} \quad (6.4)$$

where h_i^{old} is the current element size and ρ is the convergence rate of the solution of the element. If there is no singularity in the problem, $\rho = p$.

The new element size information obtained from Equation (6.4) is then fed into the mesh generator for the generation of a new mesh for the next cycle of analysis. The refinement procedure will be repeated until the total accuracy (Equation (6.1)) is satisfied. In Equation (6.3) it can be found that error of the next mesh is designed to be equally distributed in each element, which can make the adaptive refinement achieve the maximum possible convergence rate.

6.2 Adaptive Refinement Schemes for TWS

For TWSs, the error of the FEM solution is analyzed in surface direction and normal direction (i.e. thickness direction) separately, hence compared to the general adaptive refinement scheme, the procedures will alter accordingly.

6.2.1 Notations used

In this thesis, for the symbols of stress, energy norm and error norm, it is necessary to use super/sub scripts to show whether the energy norm is for the surface direction part or the normal direction part, or whether the error norm is belong to the junction zone or the shell zone. Therefore, in order to give a clear description of the adaptive refinement scheme that will be presented, it is needed to specify consistent notations for the super/sub scripts of symbols used. The

definitions of the super/sub scripts are summarized in Table 6.1.

Group	Properties	Notation	Position	Example
Location of structures	Junction part	junc	Lower RH corner of the symbol	η_{shel}
	Shell part	shel		
	Whole domain	Ω		
	The i th element	Ω_i		
Components of stresses	Total energy	No subscript	Lower LH corner of the symbol	$_{\text{surf}}\eta$
	Surface direction	surf		
	Normal direction	norm		
Different source	From FE solution	$\wedge$	Top of the symbol	$\hat{\eta}$
	From recovered stress	$\sim$		
	Estimated value	-		
	Exact or reference value	No cap		
Expected value	Target error (will be defined later)	t	Top LH corner of the symbol	$^t\eta$
	Immediate target error (will be defined later)	it		

Table 6.1 Notations of superscripts and subscripts

Note that the contents in the last row of the above table will be introduced in the later sections.

In Chapter 5 it has already been presented that overall estimated error norm is separated into the estimated *junction zone error norm* and the estimated *shell zone error norm*, denoted as $\|\bar{\mathbf{e}}\|_{\text{junc}}$ and $\|\bar{\mathbf{e}}\|_{\text{shel}}$, respectively. Additionally, because the FEM stress vector is separated into the surface part vector and normal part vector,

Adaptive refinement

the estimated stress vector will also be partitioned to two vectors, which are denoted as $_{\text{surf}}\tilde{\boldsymbol{\sigma}}$ and $_{\text{norm}}\tilde{\boldsymbol{\sigma}}$. Thus, the estimated shell zone error norm will be further divided into the estimated *shell surface error norm* $_{\text{surf}}\|\tilde{\mathbf{e}}\|_{\text{shell}}$ and the estimated *shell normal error norm* $_{\text{norm}}\|\tilde{\mathbf{e}}\|_{\text{shell}}$. Similar to the error norm, the estimated overall energy norm will include the estimated *junction zone energy norm* $\|\bar{\mathbf{u}}\|_{\text{junc}}$ and the estimated *shell zone energy norm* $\|\bar{\mathbf{u}}\|_{\text{shel}}$, written as

$$\begin{cases} \|\bar{\mathbf{u}}\|_{\Omega} = \sqrt{\|\bar{\mathbf{u}}\|_{\text{junc}}^2 + \|\bar{\mathbf{u}}\|_{\text{shel}}^2} \\ \|\bar{\mathbf{u}}\|_{\text{junc}} = \sqrt{\int_{\text{junc}} \tilde{\boldsymbol{\sigma}}^T \mathbf{D}^{-1} \tilde{\boldsymbol{\sigma}} d\Omega} \\ \|\bar{\mathbf{u}}\|_{\text{shel}} = \sqrt{\int_{\text{shel}} \tilde{\boldsymbol{\sigma}}^T \mathbf{D}^{-1} \tilde{\boldsymbol{\sigma}} d\Omega} \end{cases} \quad (6.5)$$

The estimated *shell zone energy norm* will consist of the estimated *shell surface energy norm* $_{\text{surf}}\|\bar{\mathbf{u}}\|_{\text{shel}}$ and estimated *shell normal energy norm* $_{\text{norm}}\|\bar{\mathbf{u}}\|_{\text{shel}}$, written as

$$\begin{cases} \|\bar{\mathbf{u}}\|_{\text{shel}} = \sqrt{_{\text{surf}}\|\bar{\mathbf{u}}\|_{\text{shel}}^2 + _{\text{norm}}\|\bar{\mathbf{u}}\|_{\text{shel}}^2} \\ _{\text{surf}}\|\bar{\mathbf{u}}\|_{\text{shel}} = \sqrt{\int_{\text{shel}} (_{\text{surf}}\tilde{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} _{\text{surf}}\tilde{\boldsymbol{\sigma}} d\Omega} \\ _{\text{norm}}\|\bar{\mathbf{u}}\|_{\text{shel}} = \sqrt{\int_{\text{shel}} (_{\text{norm}}\tilde{\boldsymbol{\sigma}})^T \mathbf{D}^{-1} _{\text{norm}}\tilde{\boldsymbol{\sigma}} d\Omega} \end{cases} \quad (6.6)$$

The estimated relative errors are then obtained by

$$\bar{\eta}_{\Omega} = \frac{\|\tilde{\mathbf{e}}\|_{\Omega}}{\|\bar{\mathbf{u}}\|_{\Omega}}, \quad \bar{\eta}_{\text{junc}} = \frac{\|\tilde{\mathbf{e}}\|_{\text{junc}}}{\|\bar{\mathbf{u}}\|_{\text{junc}}}, \quad \bar{\eta}_{\text{shel}} = \frac{\|\tilde{\mathbf{e}}\|_{\text{shel}}}{\|\bar{\mathbf{u}}\|_{\text{shel}}} \quad (6.7)$$

and

$$_{\text{surf}}\bar{\eta}_{\text{shel}} = \frac{_{\text{surf}}\|\tilde{\mathbf{e}}\|_{\text{shel}}}{_{\text{surf}}\|\bar{\mathbf{u}}\|_{\text{shel}}} \quad \text{and} \quad _{\text{norm}}\bar{\eta}_{\text{shel}} = \frac{_{\text{norm}}\|\tilde{\mathbf{e}}\|_{\text{shel}}}{_{\text{norm}}\|\bar{\mathbf{u}}\|_{\text{shel}}} \quad (6.8)$$

An ideal case is that the user has defined all the target errors so that the

Adaptive refinement

adaptive refinement schemes can be terminated when all the target errors are satisfied as

$$\bar{\eta}_{\Omega} \leq {}^t\eta_{\Omega}, \quad \bar{\eta}_{\text{junc}} \leq {}^t\eta_{\text{junc}}, \quad \bar{\eta}_{\text{shel}} \leq {}^t\eta_{\text{shel}}, \quad \text{surf} \bar{\eta}_{\text{shel}} \leq \text{surf} {}^t\eta_{\text{shel}}, \quad \text{norm} \bar{\eta}_{\text{shel}} \leq \text{norm} {}^t\eta_{\text{shel}} \quad (6.9)$$

But in case the user only defines part of the target error (e.g. ${}^t\eta_{\Omega}$ only), then it is necessary to design an algorithm to determine the approximate target error for the remaining parts so that optimal results can be obtained.

6.2.2 Target error distribution

In this study, the target error distribution algorithm is designed for two cases:

- 1) The target junction zone error (${}^t\eta_{\text{junc}}$), the target shell zone error (${}^t\eta_{\text{shel}}$) and the target overall error (${}^t\eta_{\Omega}$) have been given.

In this case, the target shell surface error and target shell normal error need to be defined. In order to get reasonable values for $\text{surf} {}^t\eta_{\text{shel}}$ and $\text{norm} {}^t\eta_{\text{shel}}$ based on the given ${}^t\eta_{\text{shel}}$, a feasible way is to use the information of error allocation in different directions from the current mesh. It can be assumed that target errors in different directions have the same allocation as the estimated errors of the current mesh. That is, for a reasonable refinement mesh,

$$\left\{ \begin{array}{l} \frac{\text{surf} {}^t\eta_{\text{shel}}}{{}^t\eta_{\text{shel}}} \approx \frac{\text{surf} \bar{\eta}_{\text{shel}}}{\bar{\eta}_{\text{shel}}} \\ \frac{\text{norm} {}^t\eta_{\text{shel}}}{{}^t\eta_{\text{shel}}} \approx \frac{\text{norm} \bar{\eta}_{\text{shel}}}{\bar{\eta}_{\text{shel}}} \end{array} \right. \quad (6.10)$$

Now, let

$$\frac{\text{surf} \|\bar{\mathbf{e}}\|_{\text{shel}}^2}{\|\bar{\mathbf{e}}\|_{\text{shel}}^2} = \alpha \quad \text{and} \quad \frac{\text{surf} \|\bar{\mathbf{u}}\|_{\text{shel}}^2}{\|\bar{\mathbf{u}}\|_{\text{shel}}^2} = \beta \quad (6.11)$$

Adaptive refinement

then according to the relations

$$\|\mathbf{e}\|_{\text{shel}}^2 = \|\mathbf{e}\|_{\text{surf}}^2 + \|\mathbf{e}\|_{\text{norm}}^2 \quad \text{and} \quad \|\mathbf{u}\|_{\text{shel}}^2 = \|\mathbf{u}\|_{\text{surf}}^2 + \|\mathbf{u}\|_{\text{norm}}^2 \quad (6.12)$$

it can be deduced that

$$\begin{cases} \|\mathbf{e}\|_{\text{surf}} = \sqrt{\alpha} \|\mathbf{e}\|_{\text{shel}} \\ \|\mathbf{e}\|_{\text{norm}} = \sqrt{1-\alpha} \|\mathbf{e}\|_{\text{shel}} \end{cases} \quad \text{and} \quad \begin{cases} \|\mathbf{u}\|_{\text{surf}} = \sqrt{\beta} \|\mathbf{u}\|_{\text{shel}} \\ \|\mathbf{u}\|_{\text{norm}} = \sqrt{1-\beta} \|\mathbf{u}\|_{\text{shel}} \end{cases} \quad (6.13)$$

then the target shell surface error and the target shell normal error can be computed by

$$\eta_{\text{surf}}^t = \frac{\|\mathbf{e}\|_{\text{surf}}}{\|\mathbf{u}\|_{\text{surf}}} \cdot \eta_{\text{shel}}^t = \frac{\sqrt{\alpha} \|\mathbf{e}\|_{\text{shel}}}{\sqrt{\beta} \|\mathbf{u}\|_{\text{shel}}} \cdot \eta_{\text{shel}}^t = \sqrt{\frac{\alpha}{\beta}} \eta_{\text{shel}}^t \quad (6.14)$$

$$\eta_{\text{norm}}^t = \frac{\|\mathbf{e}\|_{\text{norm}}}{\|\mathbf{u}\|_{\text{norm}}} \cdot \eta_{\text{shel}}^t = \frac{\sqrt{1-\alpha} \|\mathbf{e}\|_{\text{shel}}}{\sqrt{1-\beta} \|\mathbf{u}\|_{\text{shel}}} \cdot \eta_{\text{shel}}^t = \sqrt{\frac{1-\alpha}{1-\beta}} \eta_{\text{shel}}^t \quad (6.15)$$

In this way, a given target shell error has been distributed to the shell surface direction part and shell normal direction part. Note that for TWSs, generally, due to the surface direction domination, the shell surface direction energy is much bigger than the shell normal direction energy, while the shell surface direction error is often smaller than the shell normal direction error, hence α is a value smaller than 0.5 (for most of the cases) and β is a value close to 1. Then the value of η_{surf}^t calculated by Equation (6.14) is always smaller than the value of η_{shel}^t , and the value of η_{norm}^t calculated by Equation (6.15) is bigger than the value of η_{shel}^t . Such a distribution of η_{shel}^t is reasonable because most of the focuses should be devoted to the surface direction for the analysis of TWSs.

2) Only the target overall error is given, all the other target errors have not been defined.

Adaptive refinement

In this case, $\bar{\eta}_{\Omega}$ should be firstly allocated to $\bar{\eta}_{\text{junc}}$ and $\bar{\eta}_{\text{shel}}$. Using a similar assumption to Equation (6.10), it can be obtained that for a reasonable refined mesh,

$$\begin{cases} \frac{{}^t\eta_{\text{junc}}}{\bar{\eta}_{\text{junc}}} \approx \frac{\bar{\eta}_{\text{junc}}}{\bar{\eta}_{\Omega}} \\ \frac{{}^t\eta_{\text{shel}}}{\bar{\eta}_{\text{shel}}} \approx \frac{\bar{\eta}_{\text{shel}}}{\bar{\eta}_{\Omega}} \end{cases} \quad (6.16)$$

Now, let

$$\frac{\|\bar{\mathbf{e}}\|_{\text{shel}}^2}{\|\bar{\mathbf{e}}\|_{\Omega}^2} = A \quad \text{and} \quad \frac{\|\bar{\mathbf{u}}\|_{\text{shel}}^2}{\|\bar{\mathbf{u}}\|_{\Omega}^2} = B \quad (6.17)$$

then, it can be deduced that

$${}^t\eta_{\text{shel}} = \frac{\bar{\eta}_{\text{shel}}}{\bar{\eta}_{\Omega}} \cdot {}^t\eta_{\Omega} = \frac{\|\bar{\mathbf{e}}\|_{\text{shel}}}{\|\bar{\mathbf{u}}\|_{\text{shel}}} \cdot \frac{{}^t\eta_{\Omega}}{\bar{\eta}_{\Omega}} = \frac{\sqrt{A}\|\bar{\mathbf{e}}\|_{\Omega}}{\sqrt{B}\|\bar{\mathbf{u}}\|_{\Omega}} \cdot \frac{{}^t\eta_{\Omega}}{\bar{\eta}_{\Omega}} = \sqrt{\frac{A}{B}} {}^t\eta_{\Omega} \quad (6.18)$$

and

$${}^t\eta_{\text{junc}} = \frac{\bar{\eta}_{\text{junc}}}{\bar{\eta}_{\Omega}} \cdot {}^t\eta_{\Omega} = \frac{\|\bar{\mathbf{e}}\|_{\text{junc}}}{\|\bar{\mathbf{u}}\|_{\text{junc}}} \cdot \frac{{}^t\eta_{\Omega}}{\bar{\eta}_{\Omega}} = \frac{\sqrt{1-A}\|\bar{\mathbf{e}}\|_{\Omega}}{\sqrt{1-B}\|\bar{\mathbf{u}}\|_{\Omega}} \cdot \frac{{}^t\eta_{\Omega}}{\bar{\eta}_{\Omega}} = \sqrt{\frac{1-A}{1-B}} {}^t\eta_{\Omega} \quad (6.19)$$

After the target shell zone error is determined by Equation (6.18), it will be further distributed into ${}_{\text{surf}}^t\eta_{\text{shel}}$ and ${}_{\text{norm}}^t\eta_{\text{shel}}$ by using Equations (6.14) and (6.15).

Note that the target errors determined by above Equations are only the approximated target errors which are used as guidance for the adaptive refinement analysis. In the practical implementation, not all the conditions shown in Equation (6.9) must be contented, but only the target errors required by the user need to be satisfied.

6.2.3 Selection of the adaptive scheme

For a coarse initial mesh, generally, it is impossible to achieve the target error

by only one step of refinement, several refinement steps are usually needed to get the final satisfied mesh, of which the FEM error is within the tolerable range. In order to design a good adaptive refinement procedure, two aspects must be considered: 1) the adaptive refinement procedure is able to achieve a final satisfied mesh in a few refinement steps; 2) the final satisfied mesh should be optimized in such a way that the optimal number of DOFs is used for it. To aim these two targets, in this section, some techniques will be used to control the refinement speed and the refinement directions.

6.2.3.1 The immediate target error

A coarse mesh is unable to provide reliable FEM results for a complex problem, hence the error (Equation (6.7)) estimated from it is not as reliable as the one obtained from a fine mesh. In addition, for a coarse mesh, the difference between its estimated relative error and the final target error is big, if the final target error is used directly for the next refinement, the mesh to be generated will often be over-refined. Here the word 'over-refined' does not mean the mesh is fine enough to satisfy the accuracy requirement, but only means that the local error distribution in this mesh is not nearly equally distributed due to the unreliable error estimated from the previous mesh. Compared to a mesh with optimized element sizes, in some locations of an over refined mesh, the element sizes may be too small while in some other locations, the element sizes are too big. In order to control the refinement speed to get a better refinement effect, the *immediate target error*, which is denoted as η^{it} (can refer to the target overall error, target junction zone error, target shell zone error, target shell surface error or target shell normal error by attaching subscripts), will be used as a guide for the next refinement step.

The immediate target errors are not fixed values. They are used only in

Adaptive refinement

generation of the next mesh. As the mesh is refined, the immediate target errors will be redefined at each refinement step. For the i th mesh in the refinement procedure, the immediate target error for the next refined mesh is determined by the following scheme (Box 6.1):

- i) Start with the initial mesh where $i = 1$ and initialize the immediate target error to a big value. In the current study, the initial value of ${}^{\text{it}}\eta_i$ is set to 1.
- ii) If $\bar{\eta}_i \leq {}^{\text{it}}\eta_i$, end and quit.
- iii) If $\bar{\eta}_i > {}^{\text{it}}\eta_i$, ${}^{\text{it}}\eta_{i+1} = \bar{\eta}_i / 2$
 Else ${}^{\text{it}}\eta_{i+1} = \bar{\eta}_i$
- iv) $i = i + 1$
- v) Goto step ii)

Box 6.1 Determination of immediate target errors

The above scheme is suitable for all the relative errors (i.e. overall error, junction zone error, shell zone error, shell surface error and shell normal error). In addition, the target error distribution algorithm described in Section 6.2.2 is also suitable for the immediate target errors. In the case that all the target errors (${}^{\text{t}}\eta$) have been defined by the user, all the corresponding immediate target errors (${}^{\text{it}}\eta$) need to be determined by the scheme given in Box 6.1. Otherwise, this scheme is only used for the immediate target errors which are corresponding to the given target errors. For those immediate target errors whose corresponding target errors are not given, the target error distribution algorithm (Section 6.2.2) will be used for them.

6.2.3.2 Selection of refinement directions

In the current study, even if the refinement procedure is only carried out for the surface mesh, the shell normal error of the solid mesh will also be reduced because the element sizes are getting smaller. Additionally, in case that the surface mesh is coarse, no matter how many layers are used in the normal direction, the error of the final solid mesh is still big because of the surface direction domination. Thus, it is better to refine the mesh only in surface direction in the first several steps until the surface direction error is within an acceptable range, and then start to refine the mesh along the normal direction.

For a special case, when the problem is nearly a 2D problem (either plain stress problem or plain strain problem) or the structure is rather thin (e.g. the length to thickness ratio is less than 1/200) without any junction, the energy in normal direction can almost be ignored. Therefore, only a single layer of element is enough for this kind of structures.

For this reason, two parameters can be established for the selection of mesh refinement directions:

- (1) R_η : limit of the ratio between $_{\text{surf}}\bar{\eta}_{\text{shel}}$ and $_{\text{surf}}^t\eta_{\text{shel}}$. If $_{\text{surf}}\bar{\eta}_{\text{shel}} / _{\text{surf}}^t\eta_{\text{shel}} > R_\eta$,

the current mesh is regarded to be too coarse in surface direction. Then there is no need to refine in normal direction for the next mesh.

- (2) R_u : limit of the ratio between $_{\text{norm}}\|\mathbf{u}\|_{\text{shel}}^2$ and $\|\mathbf{u}\|_{\text{shel}}^2$. If $_{\text{norm}}\|\mathbf{u}\|_{\text{shel}}^2 / \|\mathbf{u}\|_{\text{shel}}^2 < R_u$,

the energy in normal direction is regarded as trivial. Then, it is not necessary to refine the current mesh in normal direction.

By considering both these two parameters, the algorithm of selection of mesh refinement directions is described in Box 6.2.

Adaptive refinement

$$\text{if } \frac{\text{surf } \bar{\eta}_{\text{shel}}}{\text{surf } \eta_{\text{shel}}} > R_{\eta} \text{ or } \frac{\text{norm } \|\mathbf{u}\|_{\text{shel}}^2}{\|\mathbf{u}\|_{\text{shel}}^2} < R_u$$

only refine the shell zone of the mesh in surface direction

else

refine in both surface direction and normal direction

end if

Box 6.2 Selection of mesh refinement directions

Numerical tests show that a value of 5 for R_{η} and a value of 1% for R_u perform well for the selection of refinement directions and thus are adopted in the current study.

6.2.4 Algorithm to define the new element size

In the adaptive refinement scheme, if any of the target accuracy given by the users is not satisfied, the current mesh needs to be refined. Equations (6.2) to (6.4) will be used to determine the new element sizes for the next mesh that will be generated. In the mesh generation procedure, the sizes in the thickness direction of solid elements are controlled by the number of dividing layers, if the shell thickness is a constant for a given local position, the element size in thickness direction is inversely proportional to the number of layers in thickness direction. Hence, in the current study, the new element size will consist of two parts of information: 1) surface element sizes for surface mesh generation and 2) numbers of element layers for solid mesh generation. For a structure with given thicknesses, the element sizes in thickness direction can be easily calculated once the numbers of element layers are known and vice versa. For a solid element located in the junction zone, since it is considered as a general 3D element, the surface element size and the normal

element size can be set to be equal, hence only a single value for the new element sizes is needed. While for a solid element coming from the shell zone, the new element sizes must be calculated separately in surface direction and normal direction.

6.2.4.1 Element sizes determination

In the mesh generation process, the surface elements are first extruded to volume elements and then the volume elements are layered into solid elements. So for every solid element, there must be only one surface element corresponding to it. On the contrary, for every surface element, there may be several solid elements corresponding to it (there will be only one corresponding solid element when 1 layer in thickness direction is used). For this reason, the surface direction size for any solid element can be defined as the size of its corresponding element in the surface mesh. The size of a surface element is calculated by

$$_{\text{surf}} h = \sqrt{A} \quad (6.20)$$

where A is the area of the surface element size. Then, for a solid element within the shell zone, the new surface element size will be calculated by

$$_{\text{surf}} \mathbf{e}_{\text{shel}}^e = \frac{_{\text{it}} \eta_{\text{shel}}}{\sqrt{\text{NE}_{\text{shel}} \left(\frac{_{\text{surf}} \bar{\eta}_{\text{shel}}}{_{\text{it}} \eta_{\text{shel}}} \right)^{d/p}}} \quad \text{and} \quad _{\text{surf}} h_i^{\text{new}} = \frac{_{\text{surf}} h_i^{\text{old}}}{\left(\frac{_{\text{surf}} \|\mathbf{e}\|_{\text{shel}}}{_{\text{surf}} \mathbf{e}_{\text{shel}}^e} \right)^{1/(p+1)}} \quad (6.21)$$

where $_{\text{surf}} \mathbf{e}_{\text{shel}}^e$ is the local allowable error norm in the surface direction of the shell zone. NE_{shel} is the number of the elements in the shell zone in the current mesh. d is the dimension of the elements and p is the polynomial order of shape functions of the elements. For an element from the junction zone, the new surface element size can be calculated by

Adaptive refinement

$$\mathbf{e}_{\text{junc}}^e = {}^{\text{it}}\eta_{\text{junc}} \frac{\|\bar{\mathbf{u}}\|_{\text{junc}}}{\sqrt{\text{NE}_{\text{junc}} \left(\frac{{}^{\text{it}}\bar{\eta}_{\text{junc}}}{\eta_{\text{junc}}} \right)^{d/p}}} \quad \text{and} \quad {}_{\text{surf}}h_i^{\text{new}} = \frac{{}_{\text{surf}}h_i^{\text{old}}}{\left(\frac{\|\bar{\mathbf{e}}\|_{\text{junc}}}{\mathbf{e}_{\text{junc}}^e} \right)^{1/(p+1)}} \quad (6.22)$$

where $\mathbf{e}_{\text{junc}}^e$ is the local allowable error norm of the junction zone. NE_{junc} is the number of the elements in the junction zone in the current mesh. In Equations (6.21) and (6.22), d (dimension of the elements) is 3 and p (polynomial order for shape functions) is 2 since solid quadratic elements are used in the current study.

The algorithm of determination of the new element size in the thickness direction is not straightforward as the one for surface element size. Note that in the mesh generation process, the element size in normal direction is controlled by the number of layers of element defined at the nodal points of the surface mesh (denoted as NEL). The thickness direction element size at the i th node in the surface mesh is calculated by

$${}_{\text{norm}}h_i = \frac{t_i}{\text{NEL}_i} \quad (6.23)$$

where t_i is the shell thickness at the i th node. The thickness direction element size of a surface element can be calculated by

$${}_{\text{norm}}h = \frac{1}{n} \cdot \sum_{j=1}^n {}_{\text{norm}}h_j \quad (6.24)$$

where n is the number of the nodes of this surface element and h_j is the thickness direction element size at the j th element node. Hence, for a solid element within the shell zone, the new thickness direction element size obtained from it can be calculated by

$${}_{\text{norm}}\mathbf{e}_{\text{shel}}^e = {}_{\text{norm}}\hat{\eta}_{\text{shel}} \frac{{}_{\text{norm}}\|\bar{\mathbf{u}}\|_{\text{shel}}}{\sqrt{\text{NE}_{\text{shel}} \left(\frac{{}_{\text{norm}}\bar{\eta}_{\text{shel}}}{\tilde{\eta}_{\text{shel}}} \right)^{d/p}}} \quad \text{and} \quad {}_{\text{norm}}h_i^{\text{new}} = \frac{{}_{\text{norm}}h_i^{\text{old}}}{\left(\frac{{}_{\text{norm}}\|\bar{\mathbf{e}}\|_{\text{shel}}}{{}_{\text{norm}}\mathbf{e}_{\text{shel}}^e} \right)^{1/(p+1)}} \quad (6.25)$$

where ${}_{\text{norm}}\mathbf{e}_{\text{shel}}^e$ is the local allowable error norm in the normal direction of the shell

zone. Again, d is set to 3 and p is set to 2 in the above equation. For an element within the junction zone, since it is regarded as a general 3D solid element and there is no distinct surface direction and normal direction for it, hence there is no need to calculate the element size in normal direction particularly. In consider of that the thickness to length ratio of an ideal 3D element should be close to 1, in this study, the thickness direction size of an element from the junction zone is just set to be equal to its surface direction size.

6.2.4.2 Element sizes adjustment

It is found that although immediate target errors are used to slow down the refinement speed, it is still possible to refine the mesh too fast, especially for the initial refinement step. Therefore, it is necessary to adjust the new element sizes determined in Section 6.2.4.1 to control the refinement speed by

$$\frac{\text{surf } h_i^{\text{new}}}{\text{surf } h_i^{\text{old}}} \geq \text{surf RH}_{\min} \quad \text{and} \quad \frac{\text{norm } h_i^{\text{new}}}{\text{norm } h_i^{\text{old}}} \geq \text{norm RH}_{\min} \quad (6.26)$$

where $\text{surf RH}_{\min}$ and $\text{norm RH}_{\min}$ are the minimum ratios between the new element size and old element size in the surface direction and the normal direction, respectively. In this study, $\text{surf RH}_{\min}$ is set to 0.2 and $\text{norm RH}_{\min}$ is set to 0.5. Numerical examples show that this kind of setting performs well in the refinement speed control.

Note that although the current overall error is bigger than the target error, the new element sizes calculate by Equations (6.21), (6.22) and (6.25) are not always smaller than the old sizes. This is because that the mesh node distribution is not optimized so that the error is not equally distributed in every element. Thus, by comparing the current mesh and the next generated mesh, the mesh may be refined in some local areas while de-refined in other local areas. If some current errors

Adaptive refinement

(junction zone error, shell zone error, shell surface error or shell normal error) are already smaller than their corresponding target errors, then there is no need to reduce the element sizes of these zones or directions. In addition, in order to save the DOFs, these sizes can be enlarged if the calculated new sizes are bigger. Note that for elements from the junction zone, there is no need to calculate the element size in thickness direction particularly hence the adjustment for thickness direction element size is only used for the elements from the shell zone. The algorithm to adjust the new surface element sizes and new thickness direction element sizes are summarized in Box 6.3.

1) *if need to refine in surface direction*

$$\text{if } h_{i,\text{surf}}^{\text{new}} < h_{i,\text{surf}}^{\text{old}} / 5$$

$$h_{i,\text{surf}}^{\text{new}} = h_{i,\text{surf}}^{\text{old}} / 5$$

end if

else

$$\text{if } h_{i,\text{surf}}^{\text{new}} < h_{i,\text{surf}}^{\text{old}}$$

$$h_{i,\text{surf}}^{\text{new}} = h_{i,\text{surf}}^{\text{old}}$$

end if

end if

2) *if need to refine in normal direction*

$$\text{if } h_{i,\text{norm}}^{\text{new}} < h_{i,\text{norm}}^{\text{old}} / 2$$

$$h_{i,\text{norm}}^{\text{new}} = h_{i,\text{norm}}^{\text{old}} / 2$$

end if

else

$$\text{if } h_{i,\text{norm}}^{\text{new}} < h_{i,\text{norm}}^{\text{old}}$$

$$h_{i,\text{norm}}^{\text{new}} = h_{i,\text{norm}}^{\text{old}}$$

end if

end if

Box 6.3 Adjustment of the new element sizes

As a surface element is corresponding to several solid elements, hence, the final size of a surface element is determined by

Adaptive refinement

$$h_{\text{surf}}^{\text{new}} = \frac{1}{n} \cdot \sum_{j=1}^n h_j^{\text{new}} \quad (6.27)$$

where n is the total number of the solid elements corresponding to it and h_j^{new} is the surface element size calculated from the j th corresponding solid elements. The final thickness direction element size is similarly determined by

$$h_{\text{norm}}^{\text{new}} = \frac{1}{n} \cdot \sum_{j=1}^n h_j^{\text{new}} \quad (6.28)$$

where n is total number of the solid elements corresponding to the surface element and h_j^{new} is the thickness direction size calculated from the j th corresponding solid elements.

6.2.5 Mesh refinement

After the new element sizes are determined, a new solid mesh will be generated for the next step of analysis. As it has been mentioned before, the mesh generation process includes two steps: surface mesh generation and solid mesh generation. In the surface mesh generation step, element sizes are controlled by the nodal metric tensors [3], while in the solid mesh generation step, the element sizes in normal direction are controlled by the number of element layers. For this reason, the newly determined information about the element sizes needs to be converted to two parts: 1) nodal metric tensors, $\mathbf{M}$ and 2) nodal number of layers of elements, NEL.

6.2.5.1 Formation of nodal element sizes

Note that the new element sizes are defined for every surface element, while in the mesh generation step, either the surface element size controlling parameter $\mathbf{M}$ or the normal element size controlling parameter NEL are defined at each nodal point

Adaptive refinement

of the surface mesh, hence, it is needed to convert the surface element sizes into the nodal sizes.

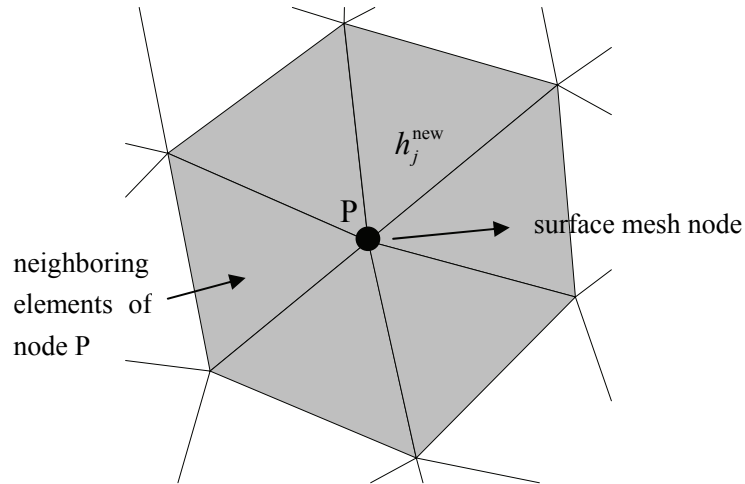


Figure 6.2 Determine the nodal element size in surface direction

The nodal element size in surface direction, $_{\text{surf}}H^{\text{new}}$, can be calculated by taking the average of all the surface sizes of its neighboring surface elements (Figure 6.2)

$$_{\text{surf}}H^{\text{new}} = \frac{1}{m} \sum_{j=1}^m _{\text{surf}}h_j^{\text{new}} \quad (6.29)$$

where $_{\text{surf}}h_j^{\text{new}}$ is the new size of the j th neighboring surface element (Section 6.2.4) and m is the number of the adjacent surface elements. Similarly, the nodal element size in thickness direction, is obtained by

$$_{\text{norm}}H^{\text{new}} = \frac{1}{m} \sum_{j=1}^m _{\text{norm}}h_j^{\text{new}} \quad (6.30)$$

Note that for the solid elements from the junction, the new nodal element size in thickness direction is set to be equal to the nodal surface element size

$$_{\text{norm}}H^{\text{new}} = _{\text{surf}}H^{\text{new}} \quad (6.31)$$

6.2.5.2 Adjustment of nodal element size in thickness direction

Note that the refinement process is now carried out in two directions, when the surface size and normal size differ too much, the element will be over-distorted and affect the accuracy of the FE analysis. In order to avoid this kind of element, some further treatments should be applied. In this study, the surface element sizes will be kept unchanged because the surface direction accuracy is more important for a TWS, hence, only the element size in thickness direction determined by Equations (6.30) and (6.31) will be adjusted. Let Q be the ratio between $_{\text{norm}}H^{\text{new}}$ and $_{\text{surf}}H^{\text{new}}$

$$Q = \frac{_{\text{norm}}H^{\text{new}}}{_{\text{surf}}H^{\text{new}}} \quad (6.32)$$

It is required to find a proper range for Q so that when

$$Q_{\min} \leq Q \leq Q_{\max} \quad (6.33)$$

the element can be regarded as not over-distorted. Thus, the thickness direction element size adjustment procedure is summarized as follows:

- i) If $_{\text{norm}}H^{\text{new}} < _{\text{surf}}H^{\text{new}} \cdot Q_{\min}$, it means that the new element is too thin (or too short) in thickness direction, then the element size in thickness direction at this node is redefined by

$$_{\text{norm}}H^{\text{new}} = Q_{\min} \cdot _{\text{surf}}H^{\text{new}} \quad (6.34)$$

- ii) If $_{\text{norm}}H^{\text{new}} > _{\text{surf}}H^{\text{new}} \cdot Q_{\max}$, it means that the new element is too thick (or too tall) in thickness direction, then the element size in thickness direction at this node is redefined by

$$_{\text{norm}}H^{\text{new}} = Q_{\max} \cdot _{\text{surf}}H^{\text{new}} \quad (6.35)$$

Note that in the final solid meshes for TWSs, there are different types of solid elements (tetrahedron, pyramid, prism and hexahedron) used for FE analysis, hence

it is impossible to find a uniform $Q_{\min}$ and $Q_{\max}$ for all those elements analytically. In this study, according to the numerical tests, $Q_{\min}$ is set to 0.05 and $Q_{\max}$ is set to 20 roughly. It is found that such limit values for Q can provide good performance for most of the cases.

6.2.5.3 Generation of the new mesh

To generate the surface mesh for the next refinement, the current surface mesh will be used as an input background mesh for the surface mesh generator [3]. In the surface mesh generation procedure, the element sizes are controlled by the nodal metric tensor. According to the definition of the nodal metric tensor (Equation (3.10)), the new metric $\mathbf{M}$ defined at the nodal point in the input mesh is written as

$$\mathbf{M}_{3D} = \begin{bmatrix} \hat{\mathbf{V}}_1 & \hat{\mathbf{V}}_2 & \hat{\mathbf{V}}_3 \end{bmatrix} \begin{bmatrix} \frac{1}{(\text{surf } H^{\text{new}})^2} & 0 & 0 \\ 0 & \frac{1}{(\text{surf } H^{\text{new}})^2} & 0 \\ 0 & 0 & \frac{1}{(\text{surf } H^{\text{new}})^2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_1 & \hat{\mathbf{V}}_2 & \hat{\mathbf{V}}_3 \end{bmatrix}^T \quad (6.36)$$

where $\hat{\mathbf{V}}_1$, $\hat{\mathbf{V}}_2$ and $\hat{\mathbf{V}}_3$ are the axis vectors of the local coordinate system. If the surface node is within the shell zone, NCS (Section 5.2.2.1) will be used as the local coordinate system. If the surface node is within the junction, then GCS will be accepted as the local coordinate system. After the new nodal metric tensors are determined, the surface mesh generator will be employed to create the new surface mesh.

In this study, the element size in the thickness direction is controlled by the nodal number of element layers (NEL). Based on the input nodal element sizes determined in Equations (6.30), (6.31) (6.34) and (6.35), the NEL values for the input surface mesh are determined by

$$NEL_i = \frac{t_i}{\text{norm } H_i} \quad (6.37)$$

where NEL_i is the nodal number of element layers at the i th node in the input surface mesh. t_i and $\text{norm } H_i$ are the shell thickness and thickness direction element size at the i th surface mesh node. A subroutine to get the NEL values for the newly generated nodes from the NEL values of the input mesh (i.e. background mesh) is developed by the author. In this subroutine, a simple linear interpolation [3] method is used to determine the NEL of a node within an input surface element. Note that since NEL is an integer, all the calculated values for NEL will be rounded. With the newly obtained NEL values, the generated surface mesh will be converted to solid mesh by the algorithms introduced in Chapter 3 and Chapter 4.

6.3 Retrieve of the Stress from the Reference Mesh

For those problems without available exact solution, in order to judge the performance of the adaptive refinement procedure, it is necessary to generate a fine mesh which can provide much more accurate solutions than the meshes used in adaptive analysis. Such a mesh is referred as the *reference mesh*, which can be obtained by using the adaptive refinement procedure until the estimated error is within an acceptable range. Compared to the refined mesh that can achieve the user defined accuracy for the first time in the adaptive refinement procedures (such a mesh is referred as the *final mesh*), the reference mesh must lead to less error, and thus it will have much more DOFs. Usually, the estimated error of the reference mesh must be at least less than the half of the estimated error of the final mesh.

For any position within the element of the mesh in an adaptive refinement step, how to retrieve its corresponding "exact" stresses from the reference mesh is a critical problem. In the current study, all the element patches of the corner nodal

points (not including the mid-side nodal points) of the reference mesh will be saved as the reference data. For an integration point within the element of the current mesh, an algorithm will be designed to locate the nearest patch which encloses this point from the reference mesh. And then the patch coordinates of this integration point will be calculated by Equation (5.59), hence the "exact" stresses of this point can be obtained by Equation (5.45). In order to locate the nearest belonging patch in the reference mesh, the element that encloses this integration point will be determined first, and then the nearest patch can be obtained by selecting the element node nearest to that integration point. Note that the nearest patch must be selected from the element that encloses the integration points, but not the node nearest to this point from the reference mesh. The reason for this measure can be explained by an example shown in Figure 6.3. In this example, there is a crack within the domain. For an integration point P in the reference mesh, the element enclosing P is ABC and B will be the assembly node of the nearest patch that encloses P. However, if using the method by searching the nearest node, then D will be found as the patch assembly node and hence wrong results will be produced.

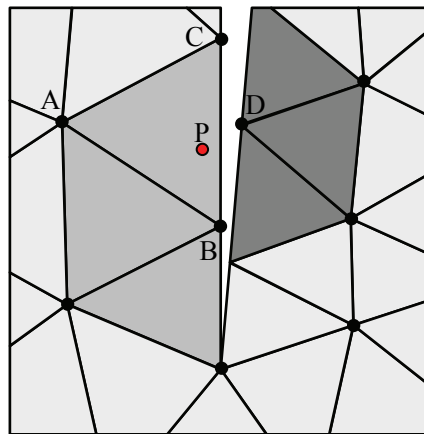


Figure 6.3 The nearest belonging patch of point P

After the nearest patch is determined, the "exact stress" at this point can be determined from the patch. The detailed algorithms will be introduced in Appendix

B.

6.4 Adaptive Refinement Examples

In this section, several numerical examples will be exhibited to show the effect of the developed adaptive refinement scheme. In order to show the performance of the estimated error, both the results of *adaptive refinement scheme* (ARS) and *uniform refinement scheme* (URS) are provided for comparison. The graphics of the distribution of overall effectivity index (i.e. θ) are also provided for some examples. The meaning of the colors used in these graphics is defined as following:

	: $ \theta-1 \leq 0.05$
	: $0.05 < \theta-1 \leq 0.25$
	: $0.25 < \theta-1 \leq 0.50$
	: $0.50 < \theta-1 \leq 0.75$
	: $ \theta-1 > 0.75$

6.4.1 Examples of structures formed by single surfaces

In this section, all the examples that will be modeled are constructed by single surfaces. Examples 6.1 to 6.2 are used to demonstrate that the presented adaptive refinement scheme can be successfully applied to 2D problems. Examples 6.3 to 6.5 are employed to show that the adaptive refinement scheme can be used for shell structures. Among them, Examples 6.3 and 6.4 are plate bending problems while Example 6.5 is a general shell structure formed by a curved surface.

The first example is the one that was used in the previous chapter (Example 5.1). Exact solution of this problem is available (Equation (5.91)), hence no reference mesh is needed for this example. Since this example can be regarded as a 2D plain stress problem, the exact stress components involved in normal direction

Adaptive refinement

are zero. FEM results show that the values of these stress components are so small that can be ignored, hence there is no need to analyze the error in normal direction particularly and only the overall error is considered. The target error for this example is set to 1%. Similar to Example 5.1, only one layer of elements is formed in the thickness direction for this example. The meshes generated in ARS and URS are shown in Figure 6.4 and Figure 6.5, respectively. The graphics of the distribution of effectivity index for each mesh in the refinement step are also given in Figure 6.4 and Figure 6.5. Some results are given in Table 6.2 and Table 6.3, for ARS and URS respectively. For all the examples used in this chapter, $\|\mathbf{e}\|$ means the error norm of the FEM stresses; $\|\mathbf{u}\|$ means the energy norm of the exact stress (if no exact solution available, use the one from the reference mesh instead). R is the convergence rate.

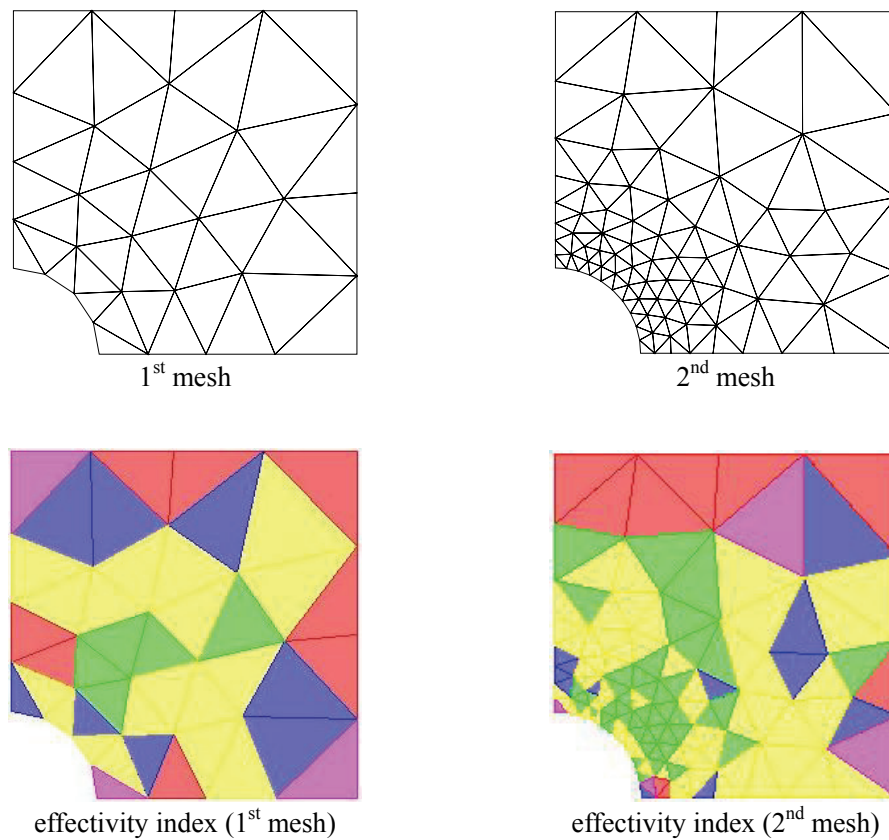


Figure 6.4 Example 6.1: adaptive refinement

Adaptive refinement

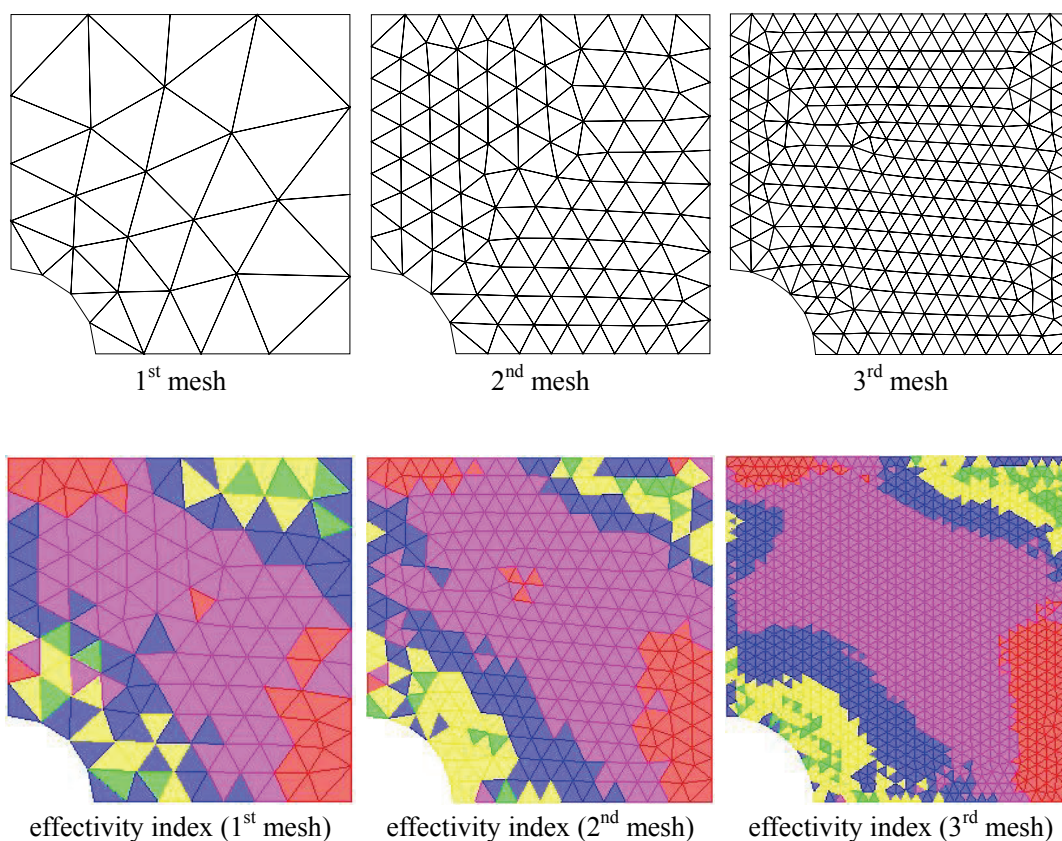


Figure 6.5 Example 6.1: uniform refinement

Mesh	NTD	$\ e\ $	$\ u\ $	η	θ	R
A1	726	1.1192E-03	4.1098E-02	2.72%	1.0873	/
A2	2046	2.3447E-04	4.1098E-02	0.57%	1.3554	1.5086

Table 6.2 Example 6.1: results of the adaptive refinement scheme

Mesh	NTD	$\ e\ $	$\ u\ $	η	θ	R
U1	726	1.6750E-03	4.1098E-02	4.08%	1.0873	/
U2	3009	1.2934E-03	4.1098E-02	3.15%	0.8068	0.1819
U3	7656	5.6094E-04	4.1098E-02	1.36%	0.8767	0.8946

Table 6.3 Example 6.1: results of the uniform refinement scheme

Adaptive refinement

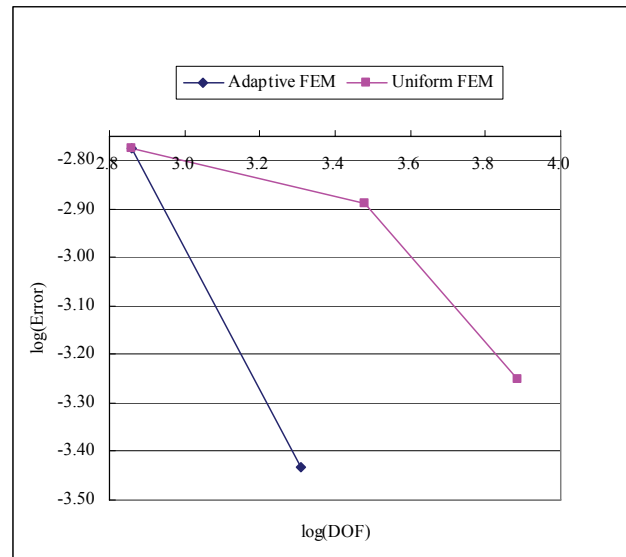


Figure 6.6 Example 6.1: comparison of convergence rate

From the results shown in Table 6.2 and Table 6.3, it can be seen that starting from a coarse mesh with 726 DOFs, the ARS takes only 1 step to reach the accuracy requirement and the final mesh has only 2046 DOFs. While the URS takes 2 steps to reach an error that is close to the accuracy requirement and the final mesh has 7566 DOFs. Thus, it is obvious that ARS can provide a higher converge rate than URS (Table 6.2 and Table 6.3, the last column). This conclusion can also be observed from Figure 6.6. From the graphics of distribution of effectivity indexes shown in Figure 6.4 and Figure 6.5, it can be found that the θ values at the domain boundary are not as good as the ones in the interior part, which shows the stress recovery effect of the inner patch is better than the boundary patch (Section 5.3.2.4). From Figure 6.4 and Figure 6.5, it can also be seen that ARS provides a better distribution of θ compared to URS. For comparison, the results of the shell analysis on this problem can be found in reference [10]. It can be seen that the presented ARS performs as good as the one proposed by Zienkiewicz and Zhu [10].

Adaptive refinement

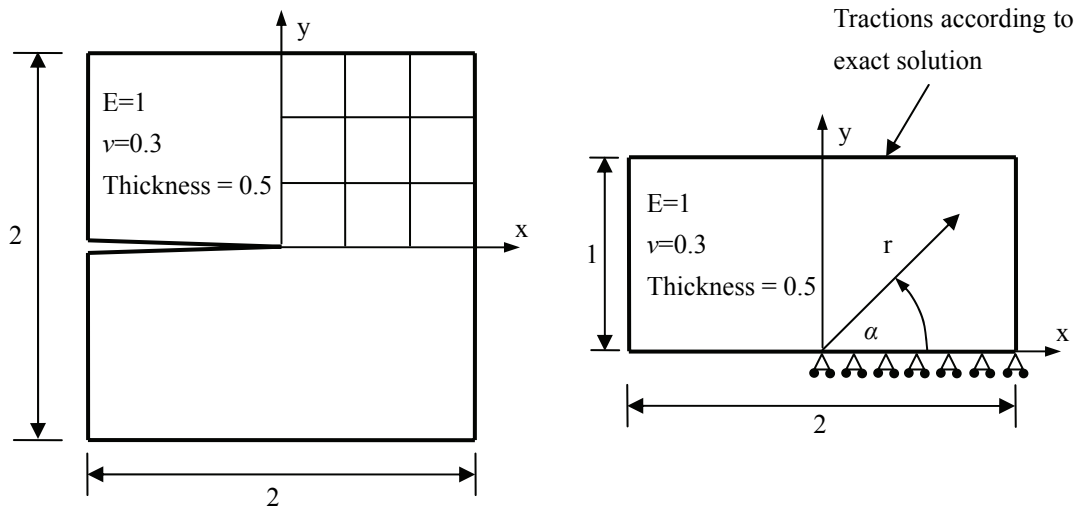


Figure 6.7 Example 6.2: region near a crack tip

The problem of the Example 6.2 is a region near the tip of a crack (Figure 6.7) on a plate. It is loaded by tractions given by an exact solution, which is expressed by

$$\begin{aligned}\sigma_x &= \frac{1}{\sqrt{r}} \cos \frac{\alpha}{2} \left(1 - \sin \frac{\alpha}{2} \sin \frac{3\alpha}{2} \right) \\ \sigma_y &= \frac{1}{\sqrt{r}} \cos \frac{\alpha}{2} \left(1 + \sin \frac{\alpha}{2} \sin \frac{3\alpha}{2} \right) \\ \tau_{xy} &= \frac{1}{\sqrt{r}} \sin \frac{\alpha}{2} \cos \frac{\alpha}{2} \cos \frac{3\alpha}{2}\end{aligned}\tag{6.38}$$

Exploiting symmetry, only a half of the domain is modeled. The thickness of the plate is set to 0.5. Due to the existence of a crack tip, there is a singular point in this example and the position of this singular point is $x = 0, y = 0$.

This problem is still a 2D problem (plain stress), hence only the overall error is considered and only one layer of elements is formed for all the meshes in the refinement step. The target error for this example is set to 5%. The meshes and the

Adaptive refinement

corresponding distribution of effectivity indexes in ARS and URS are shown in Figure 6.8 and Figure 6.9, respectively. Related results are listed in Table 6.4 and Table 6.5.

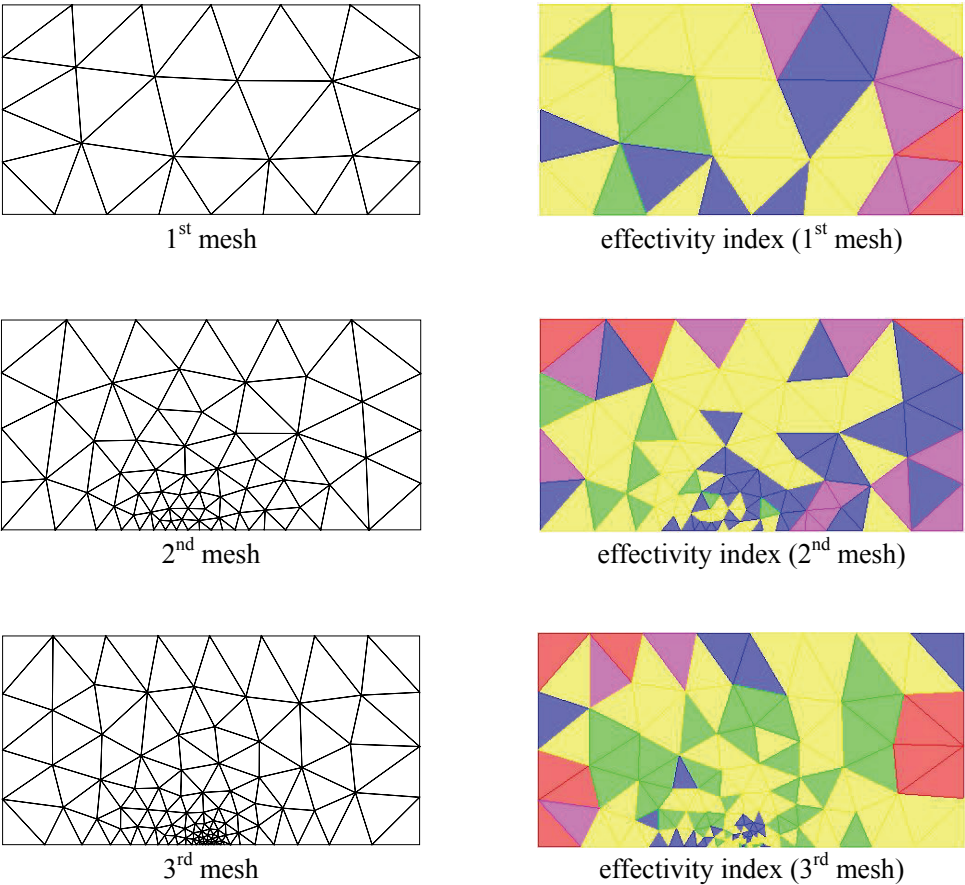
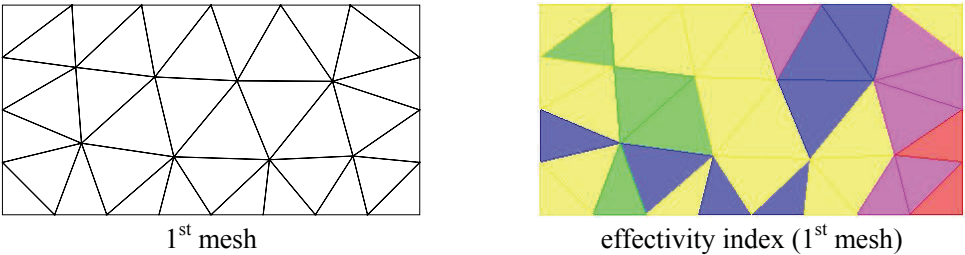


Figure 6.8 Example 6.2: adaptive refinement



Adaptive refinement

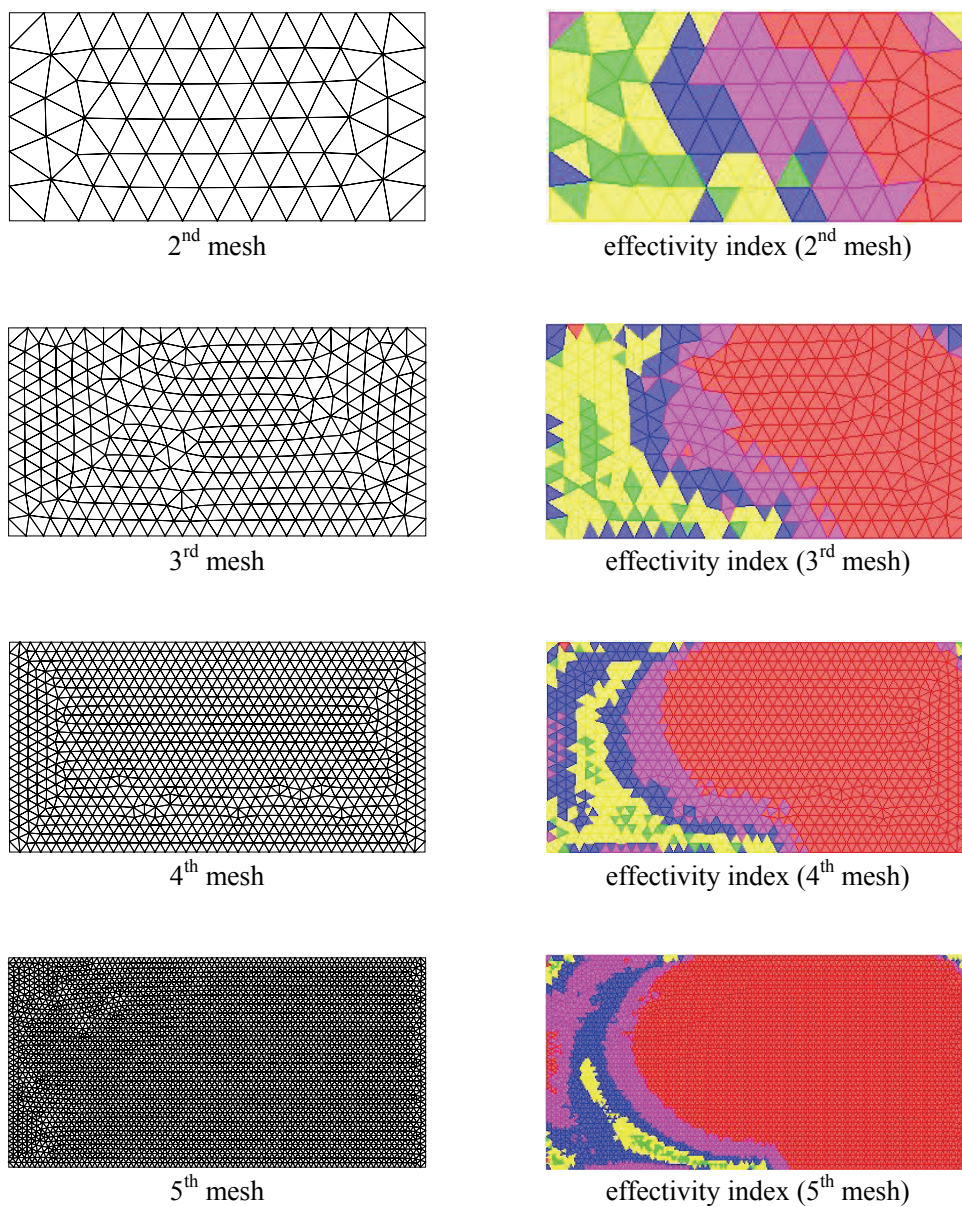


Figure 6.9 Example 6.2: uniform refinement

Mesh	NTD	$\ e\ $	$\ u\ $	η	θ	R
A1	660	6.6313E-02	4.2501E-01	15.60%	1.1061	/
A2	1815	3.8083E-02	4.2501E-01	8.96%	1.0216	0.5483
A3	2769	2.0485E-02	4.2501E-01	4.82%	0.9545	1.4680

Table 6.4 Example 6.2: results of the adaptive refinement scheme

Adaptive refinement

Mesh	NTD	$\ e\ $	$\ u\ $	η	θ	R
U1	660	6.6313E-02	4.2501E-01	15.60%	1.1061	/
U2	2088	6.1132E-02	4.2501E-01	14.38%	1.0330	0.0706
U3	8148	4.3436E-02	4.2501E-01	10.22%	1.0365	0.2510
U4	25263	3.4000E-02	4.2501E-01	8.00%	1.0288	0.2164
U5	100035	2.4981E-02	4.2501E-01	5.88%	1.0044	0.2240

Table 6.5 Example 6.2: results of the uniform refinement scheme

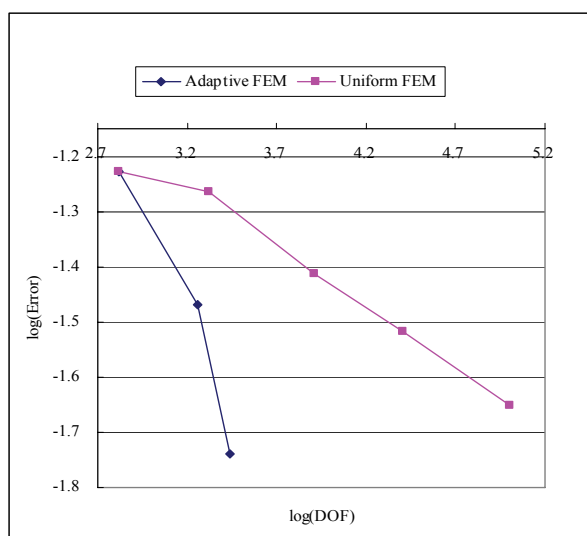


Figure 6.10 Example 6.2: comparison of convergence rate

From the results shown in Table 6.4 and Table 6.5, it can be seen that for this example, ARS can reach the target error at much less cost of DOF compared to URS (2769 DOFs in ARS while 100035 DOFs in URS), i.e. it provides a much higher converge rate (Table 6.4 and Table 6.5, the last column). Figure 6.10 also proves this conclusion. By comparing the distribution of θ , it can be found that the effectivity index distribution of ARS is better than that of URS (Figure 6.8 and Figure 6.9). From the above results, it can be found that for a problem with singular point, the URS cannot achieve a good effectivity index even for the final mesh. Similar results can be found for the shell analysis in reference [126].

Adaptive refinement

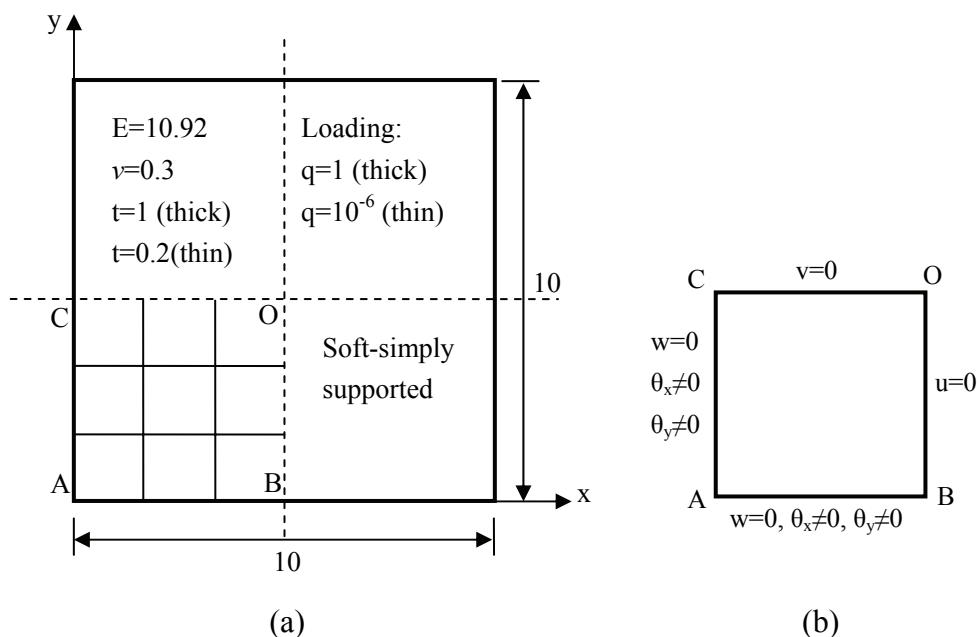


Figure 6.11 Example 6.3: a square plate under uniform pressure with soft-simply support on all sides:

(a) whole structure; (b) a quadrant of the structure and the corresponding boundary conditions

Example 6.3 is a plate bending problem and the problem involved is a square plate under uniform lateral pressure (Figure 6.11a). All the sides of the plate are *soft-simply supported* [127] and hence only the lower left symmetric quadrant was taken out for the analysis. The boundary conditions of the analyzed quadrant of the whole structure are shown in Figure 6.11b. For this problem, there is no exact 3D solution available, so a pre-refined mesh will be used as the reference mesh to provide the approximate “exact” solution. As the plate is soft-simply supported on all sides, boundary layers will be present along the domain boundary [128]. In order to observe the boundary layers effect, two values of thickness of the plates were used and they are $t/L = 0.02$ ($t = 0.2$) for the thin case and $t/L = 0.1$ ($t = 1$) for the thick case. As this problem is not a 2D problem, the performance in either the surface direction or the normal direction should be taken into consideration. Thus, the target errors will include the target surface direction error and the target normal direction error.

Adaptive refinement

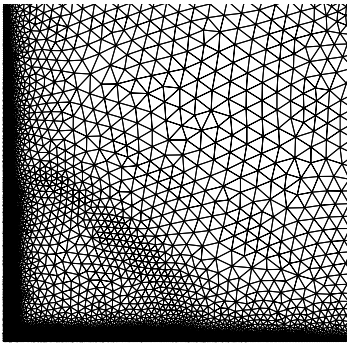


Figure 6.12 Example 6.3, Case 1: 2D view of the reference mesh

NTD	$\ u\ $			η		
	surf	normal	overall	surf	normal	overall
772542	2.32E-01	2.54E-02	2.33E-01	0.53%	8.04%	1.02%

Table 6.6 Example 6.3, Case 1: details of the reference mesh

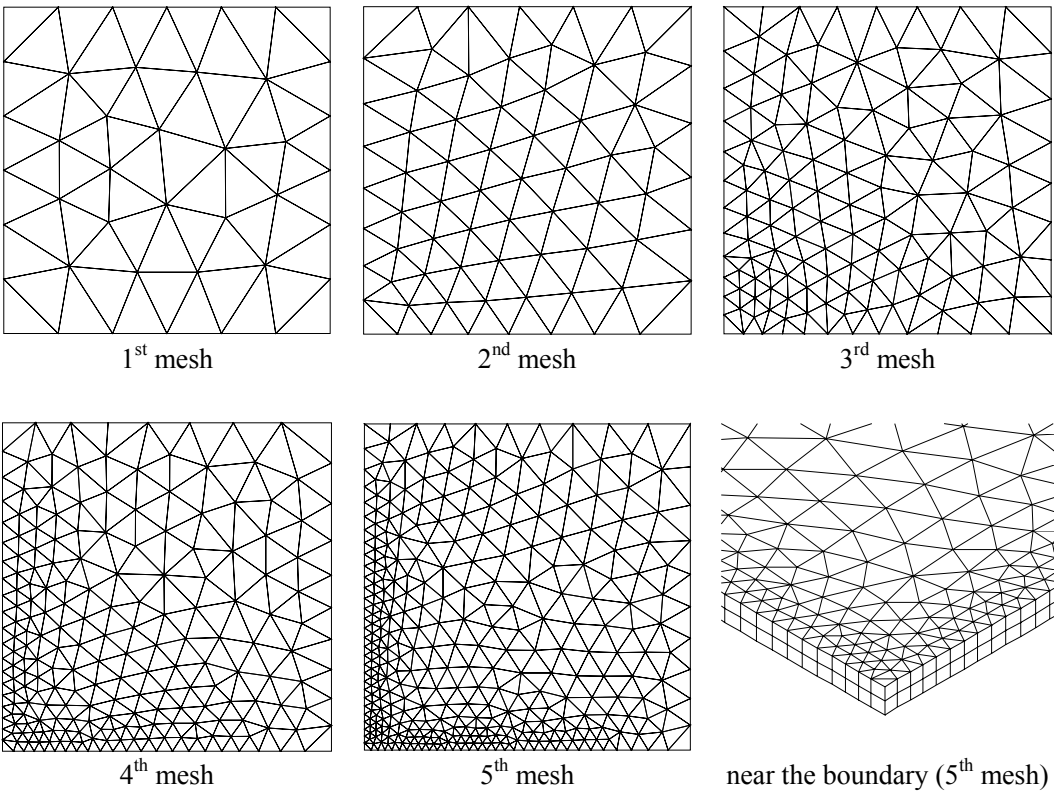


Figure 6.13 Example 6.3, Case 1: adaptive refinement

Adaptive refinement

For the first case ($t = 0.2$), the target overall error is set to 5%, the target surface direction error and target normal direction error are set to 2% and 30%, respectively. Note the target normal direction error is much bigger than the target surface direction error, this is because the plate is thin for this case so that the energy contribution from the stress components in normal direction is small. A pre-refined mesh with 772542 DOFs is used as the reference mesh (Figure 6.12 and Table 6.6) for this case. The meshes for each refinement step are shown in Figure 6.13 and Figure 6.14. The corresponding results are given in Table 6.7 and Table 6.8.

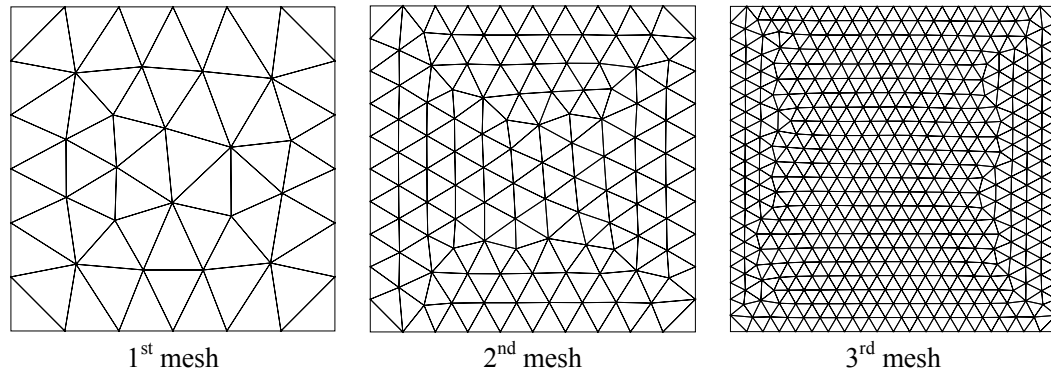


Figure 6.14 Example 6.3, Case 1: uniform refinement

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
1	999	1.34E-02	3.65E-02	3.88E-02	5.77%	143.82%	16.67%	0.8398	1.0569	1.0264
2	1758	1.20E-02	2.43E-02	2.71E-02	5.20%	95.66%	11.62%	0.7585	0.8903	0.8658
3	3171	9.27E-03	1.60E-02	1.85E-02	4.00%	63.14%	7.94%	0.7652	0.7975	0.7895
4	5670	5.43E-03	1.25E-02	1.36E-02	2.34%	49.26%	5.84%	0.7174	0.6824	0.6881
5	13821	3.22E-03	5.96E-03	6.77E-03	1.39%	23.49%	2.91%	1.0733	1.3325	1.2784

Table 6.7 Example 6.3, Case 1: results of the adaptive refinement scheme

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
1	999	1.34E-02	3.65E-02	3.88E-02	5.77%	143.82%	16.67%	0.8398	1.0569	1.0264
2	3252	1.08E-02	1.45E-02	1.81E-02	4.67%	57.29%	7.77%	0.7596	0.9896	0.9141
3	19833	6.27E-03	5.89E-03	8.60E-03	2.70%	23.23%	3.69%	0.7201	1.1946	0.9720

Table 6.8 Example 6.3, Case 1: results of the uniform refinement scheme

Adaptive refinement

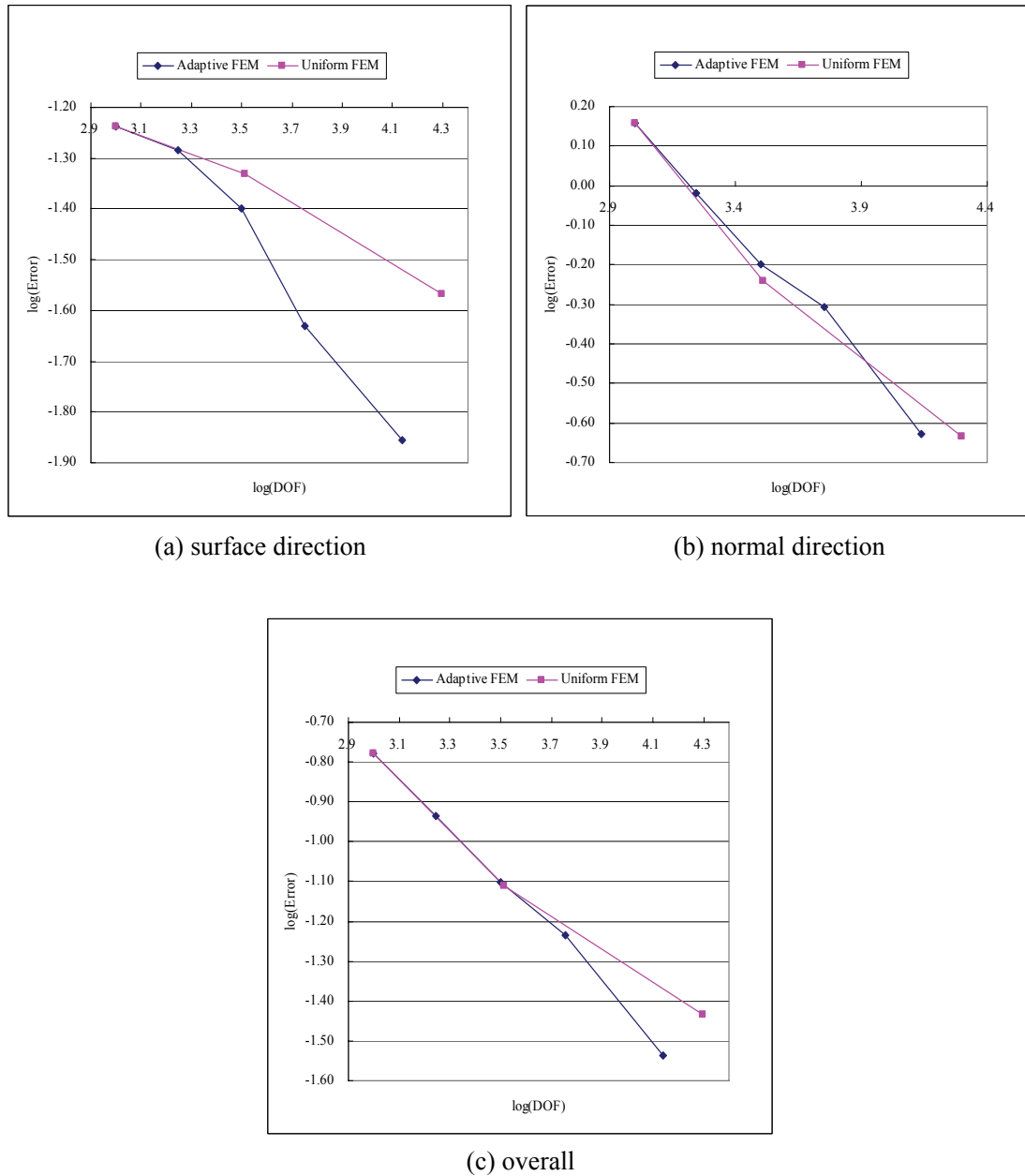


Figure 6.15 Example 6.3, Case 1: comparison of convergence rate

The results listed in Table 6.7 and Table 6.8 show that ARS can achieve the target error at less DOFs compared to URS. From Figure 6.15, it can be seen that the convergence rate of ARS in surface direction is much higher than the one of URS (Figure 6.15a). Although the convergence rates in normal direction of both

Adaptive refinement

ARS and URS are similar (Figure 6.15b), because of the surface direction predomination, the overall convergence rate of ARS is still higher than the one of URS (Figure 6.15c), which is also proved by the results in Table 6.7 and Table 6.8. Such results prove that the treatment for putting more efforts onto the refinement in surface direction in ARS is valid and efficient. Additionally, because the plate is soft-simply supported on all side, strong boundary layers effect is detected in Figure 6.13, while URS is unable to catch this stress concentration effect. Similar results of shell analysis on this problem were obtained by Lee [128].

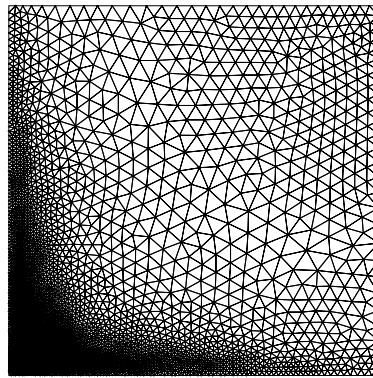


Figure 6.16 Example 6.3, Case 2: 2D view of the reference mesh

NTD	$\ u\ $			η		
	surf	normal	overall	surf	normal	overall
520524	2.10E+00	6.60E-01	2.20E+00	0.44%	1.50%	0.62%

Table 6.9 Example 6.3, Case 2: details of the reference mesh

In the second case ($t = 1$), the target surface direction error, normal direction error and the overall error are set to 1%, 5% and 2%, respectively. A pre-refined mesh with 520524 DOFs is used as a reference mesh (Figure 6.16 and Table 6.9). The meshes for each refinement step are shown in Figure 6.17 and Figure 6.18. The corresponding results are given in Table 6.10 and Table 6.11.

Adaptive refinement

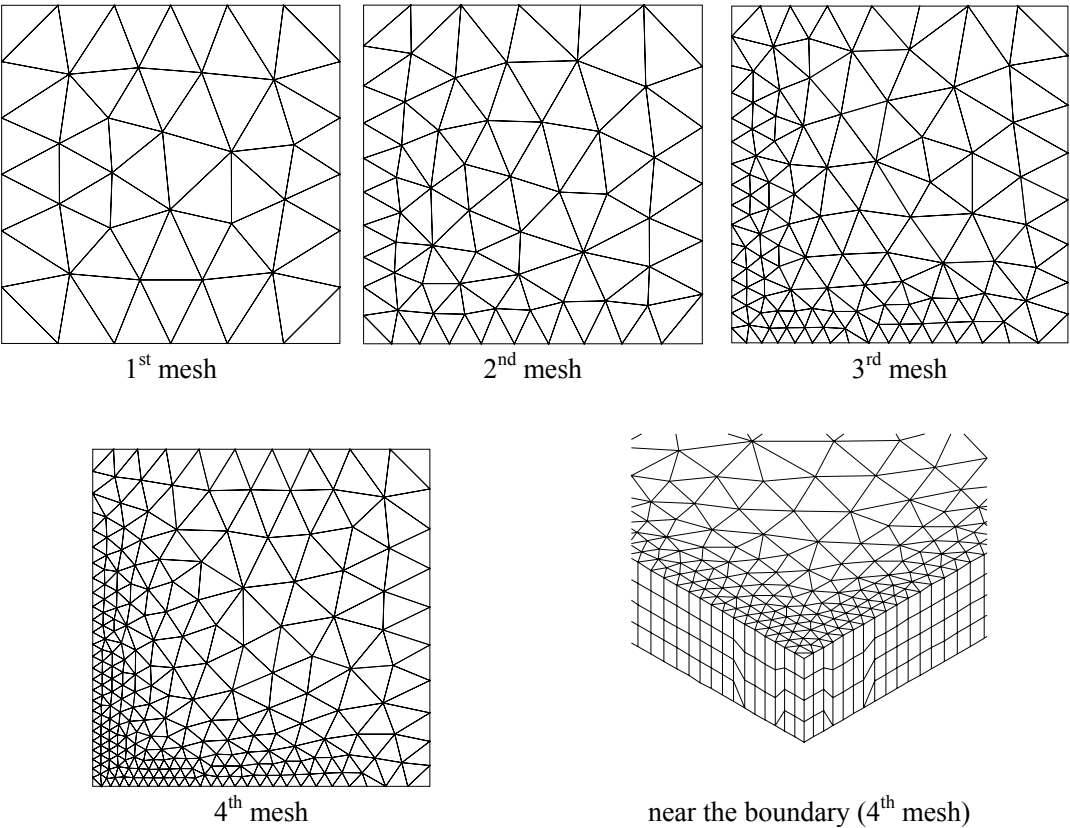


Figure 6.17 Example 6.3, Case 2: adaptive refinement

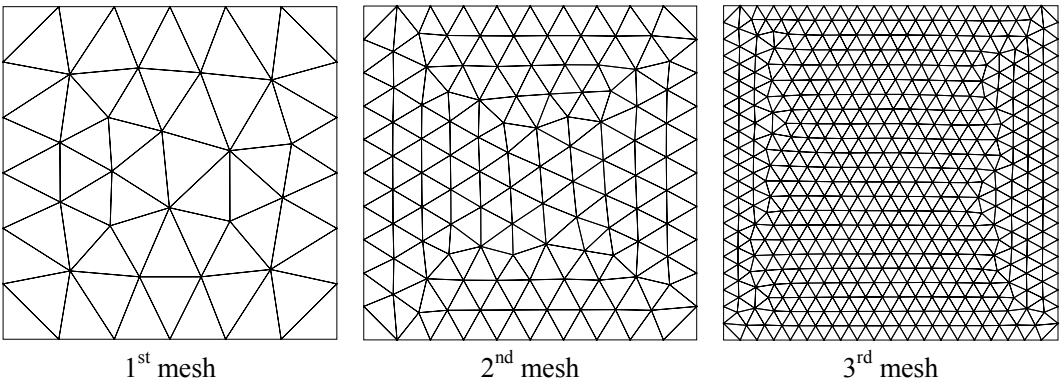


Figure 6.18 Example 6.3, Case 2: uniform refinement

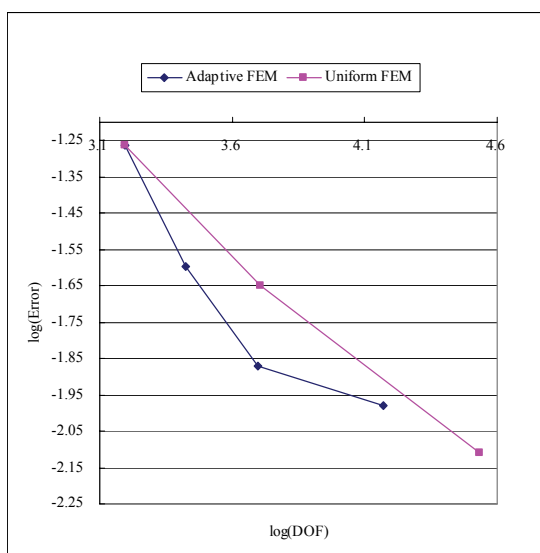
Adaptive refinement

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
A1	1563	1.15E-01	1.09E-01	1.58E-01	5.47%	16.49%	7.19%	0.7139	1.5930	1.2131
A2	2667	5.32E-02	7.74E-02	9.39E-02	2.54%	11.72%	4.27%	0.9767	2.0943	1.8120
A3	4959	2.82E-02	7.01E-02	7.56E-02	1.34%	10.62%	3.44%	1.4939	2.2659	2.1751
A4	14754	2.20E-02	3.33E-02	3.99E-02	1.05%	5.04%	1.82%	1.4898	2.1264	1.9543

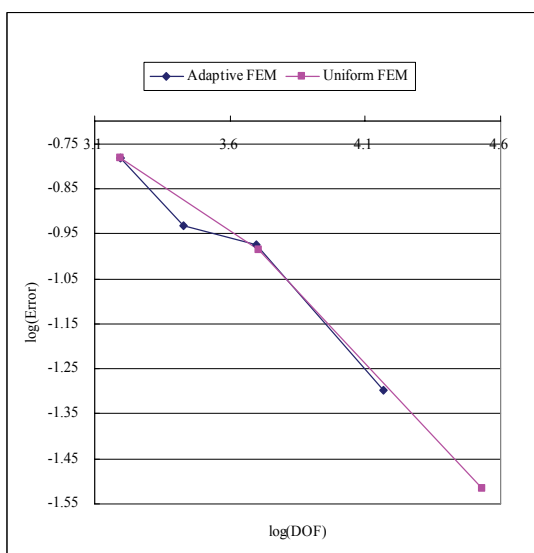
Table 6.10 Example 6.3, Case 2: results of the adaptive refinement scheme

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	all	surf	normal	all	surf	normal	all
U1	1563	1.15E-01	1.09E-01	1.58E-01	5.47%	16.49%	7.19%	0.7139	1.5930	1.2131
U2	5073	4.70E-02	6.81E-02	8.27E-02	2.24%	10.31%	3.76%	1.0454	2.3108	1.9925
U3	34035	1.63E-02	2.01E-02	2.59E-02	0.78%	3.04%	1.18%	1.3573	2.3120	1.9879

Table 6.11 Example 6.3, Case 2: results of the uniform refinement scheme

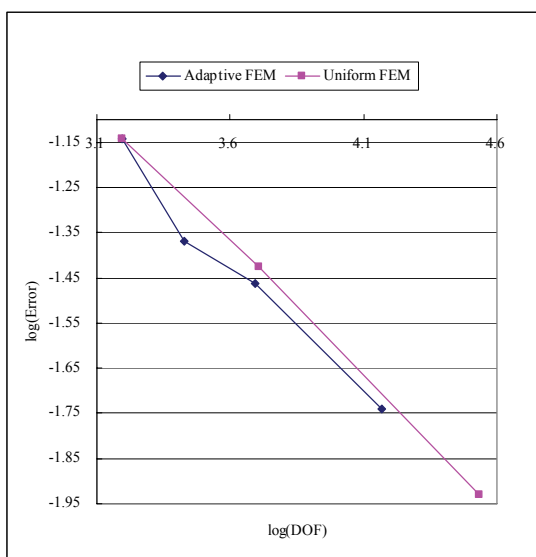


(a) surface direction



(b) normal direction

Adaptive refinement



(c) overall

Figure 6.19 Example 6.3, Case 2: comparison of convergence rate

The results given in Table 6.10 and Table 6.11 and the curves shown in Figure 6.19 show that ARS can achieve the target accuracy at less DOFs than URS. Compared to Case 1, because this plate is thick in this case, although all the sides are soft-simply supported, the boundary layers effect is not that strong (Figure 6.17).

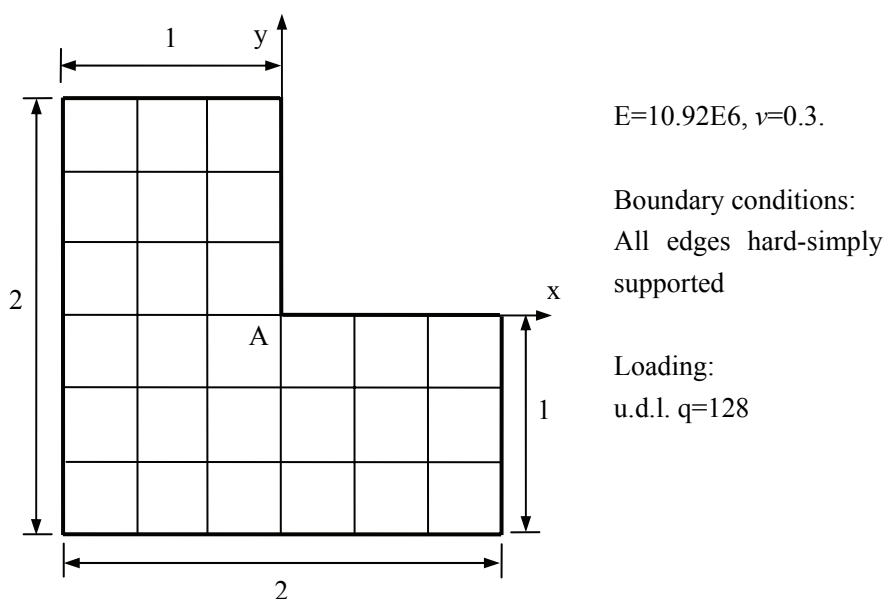


Figure 6.20 Example 6.4: L-shaped plate

Adaptive refinement

The problem of Example 6.4 is an L-shaped plate under uniform lateral pressure (Figure 6.20). This problem is also a plate bending problem. The plate is hard-simply supported on all sides. For this problem, there is a singular point located at the origin (point A, Figure 6.20). The thickness of this plate is set to be 0.04. The target overall error is set to 10%. The target surface direction error is set to 10% and the target normal direction error is set to 50%. A mesh with 734631 DOFs is used as a reference mesh (Figure 6.21 and Table 6.12) for this example. The meshes and distribution of effectivity indexes for each refinement step are shown in Figure 6.22 and Figure 6.23. The corresponding results are given in Table 6.13 and Table 6.14.

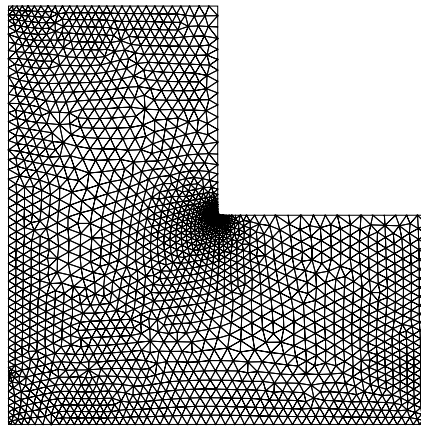


Figure 6.21 Example 6.4: 2D view of the reference mesh

NTD	$\ u\ $			η		
	surf	normal	overall	surf	normal	overall
734631	1.87E+00	3.92E-01	1.91E+00	2.79%	6.22%	3.02%

Table 6.12 Example 6.4: details of the reference mesh

Adaptive refinement

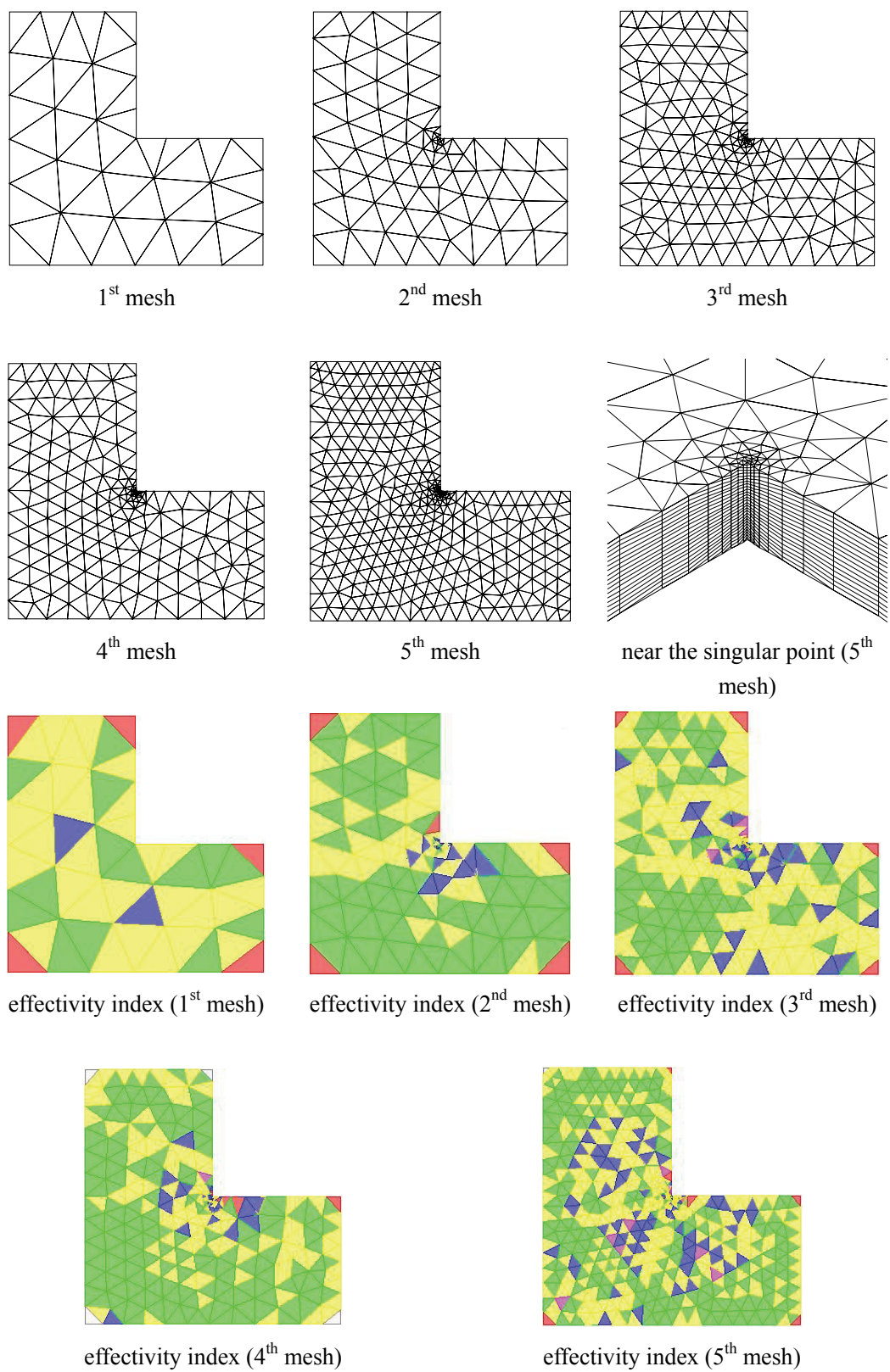


Figure 6.22 Example 6.4: adaptive refinement

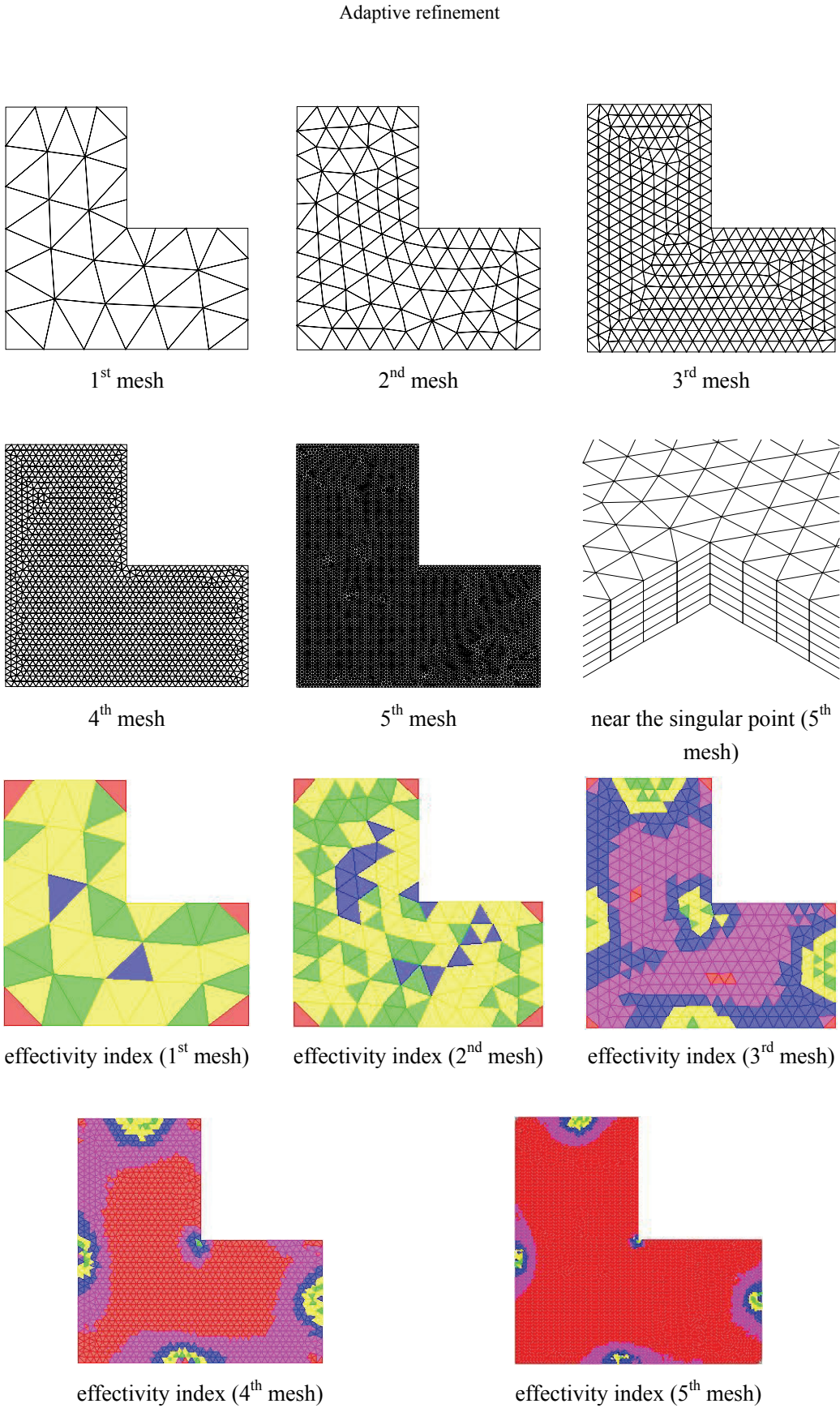


Figure 6.23 Example 6.4: uniform refinement

Adaptive refinement

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
A1	1401	8.46E-01	1.29E+00	1.54E+00	45.15%	328.94%	80.52%	0.5940	1.1384	1.0059
A2	6267	4.00E-01	5.57E-01	6.86E-01	21.38%	142.40%	35.87%	0.7769	1.1193	1.0158
A3	18300	2.63E-01	2.87E-01	3.90E-01	14.06%	73.39%	20.37%	0.8440	1.0527	0.9631
A4	18603	2.03E-01	2.56E-01	3.27E-01	10.84%	65.45%	17.09%	0.9680	1.0485	1.0182
A5	63714	1.22E-01	1.36E-01	1.82E-01	6.49%	34.67%	9.52%	0.9790	1.0303	1.0078

Table 6.13 Example 6.4: results of the adaptive refinement scheme

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
U1	1401	8.46E-01	1.29E+00	1.54E+00	45.15%	328.94%	80.52%	0.5940	1.1384	1.0059
U2	4113	6.64E-01	5.95E-01	8.91E-01	35.43%	152.03%	46.59%	0.6010	1.1285	0.8764
U3	25395	5.52E-01	3.17E-01	6.36E-01	29.45%	80.84%	33.24%	0.5963	1.3146	0.8340
U4	100803	4.25E-01	1.75E-01	4.60E-01	22.71%	44.67%	24.03%	0.6554	1.5272	0.8396
U5	565743	3.20E-01	1.12E-01	3.39E-01	17.08%	28.72%	17.72%	0.7416	1.4634	0.8514

Table 6.14 Example 6.4: results of the uniform refinement scheme

The effectivity index distribution shown in Figure 6.22 and Figure 6.23 proves that the θ values of ARS are better than those of URS, which can also be observed from the results shown in Table 6.13 and Table 6.14. From Figure 6.24b, it can be seen that since there is no normal direction refinement in the initial steps, the convergence rate of ARS is not obviously higher than one of URS. After the 3rd mesh, the refinement in normal direction is started, and then the convergence rate of ARS in normal direction surpasses the one of URS evidently. Since there is a singular point in this example, an obvious singular effect can be detected in Figure 6.22. It is also found that it is hard to control the error in normal direction for this example. This phenomenon is quite reasonable because the values of the stress components in normal direction are much smaller compared to the ones in surface direction.

Adaptive refinement

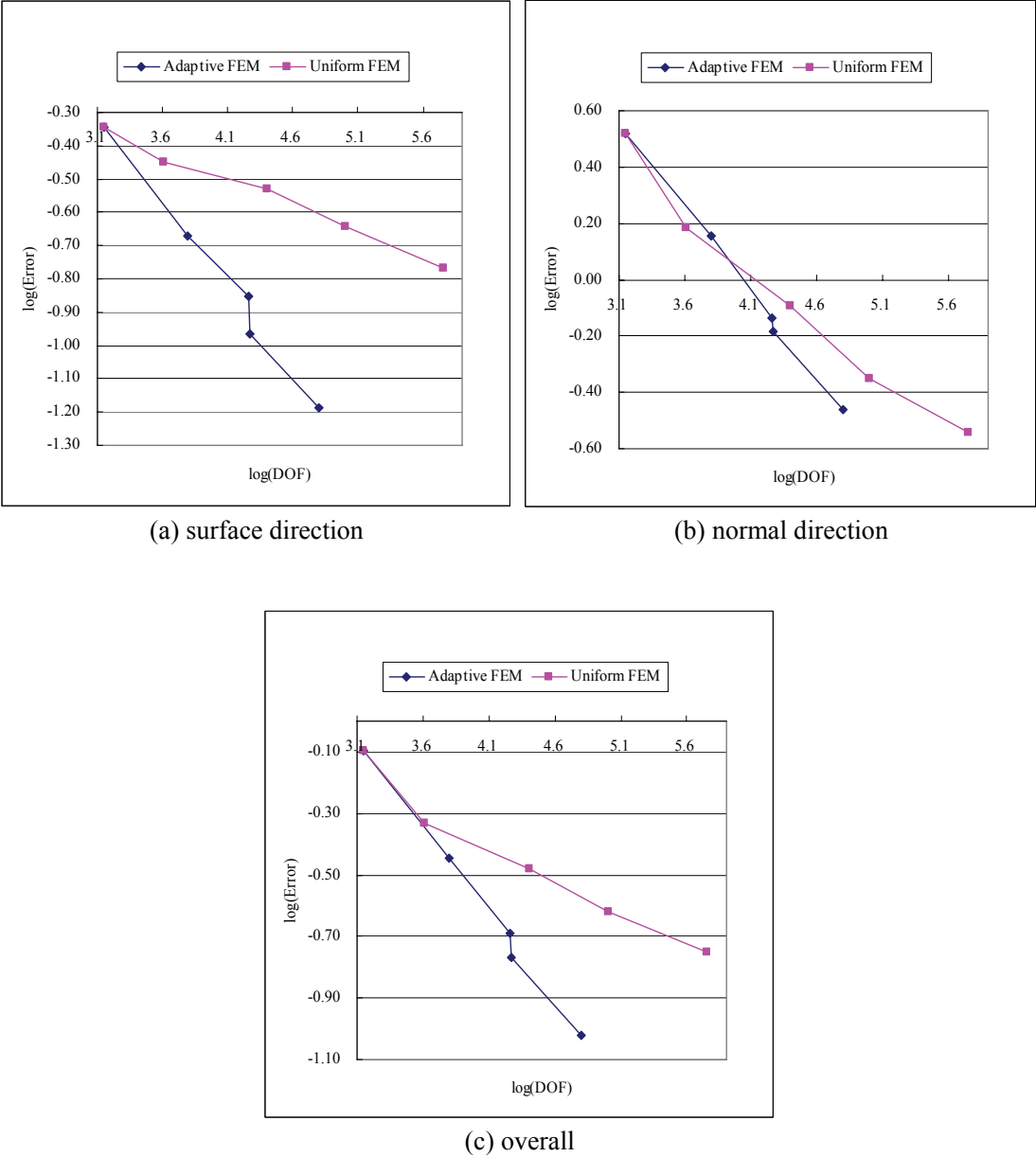


Figure 6.24 Example 6.4: comparison of convergence rate

Adaptive refinement

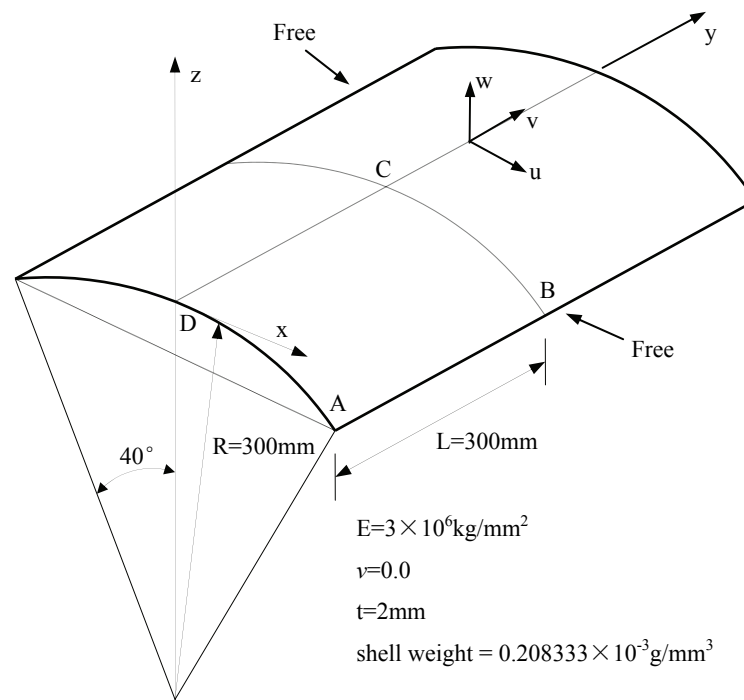


Figure 6.25 Example 6.5: cylindrical shell roof under self-weight

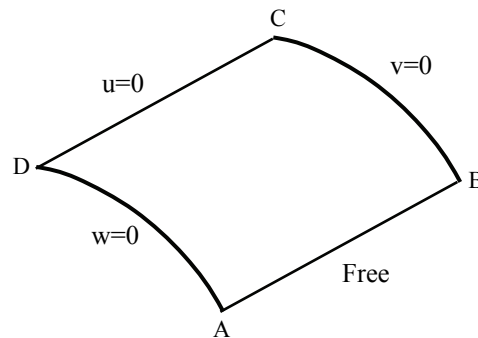


Figure 6.26 Example 6.5: boundary conditions.

Example 6.5 has been used by many researchers for the shell analysis. The shell is supported by diaphragms at the two curved ends and is free along the other two straight edges (Figure 6.25). Exploiting symmetry, only a quadrant of the roof is analyzed. The detailed boundary conditions of the quadrant are shown in Figure 6.26. The roof is subjected to loading due to its self-weight of $0.208333 \times 10^{-3} \text{ g/mm}^3$. The thicknesses of this shell is set to 2. This example is used to simulate the shell analysis so that the Possion ratio is set to 0. The target error is set to 2%. The

Adaptive refinement

target errors in surface direction and normal direction are not defined, hence, they will be determined by the algorithms presented in Section 6.2.2. A reference mesh (Figure 6.27 and Table 6.15) with 846303 DOFs is used for this example. The meshes for each refinement step are shown in Figure 6.28 and Figure 6.29. The corresponding results are given in Table 6.16 and Table 6.17.

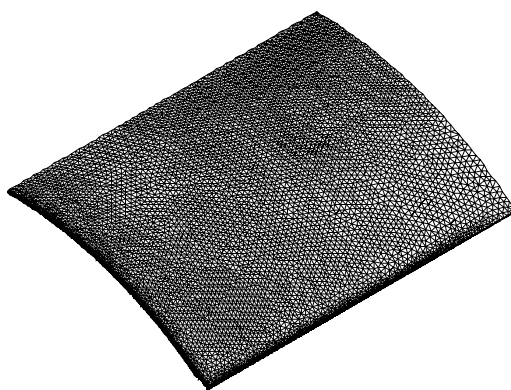
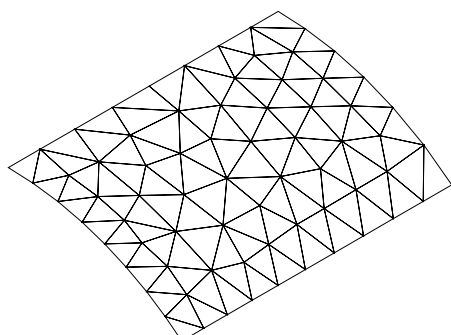


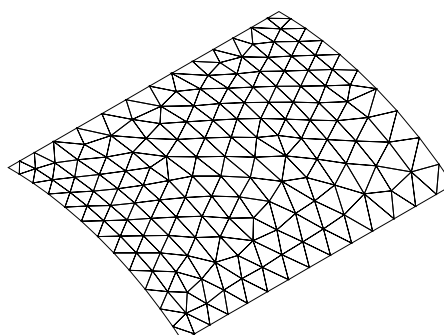
Figure 6.27 Example 6.5: reference mesh

NTD	$\ u\ $			η		
	surf	normal	overall	surf	normal	overall
846303	1.84E-01	5.67E-03	1.84E-01	0.28%	14.99%	0.54%

Table 6.15 Example 6.5: details of the reference mesh



1st mesh



2nd mesh

Adaptive refinement

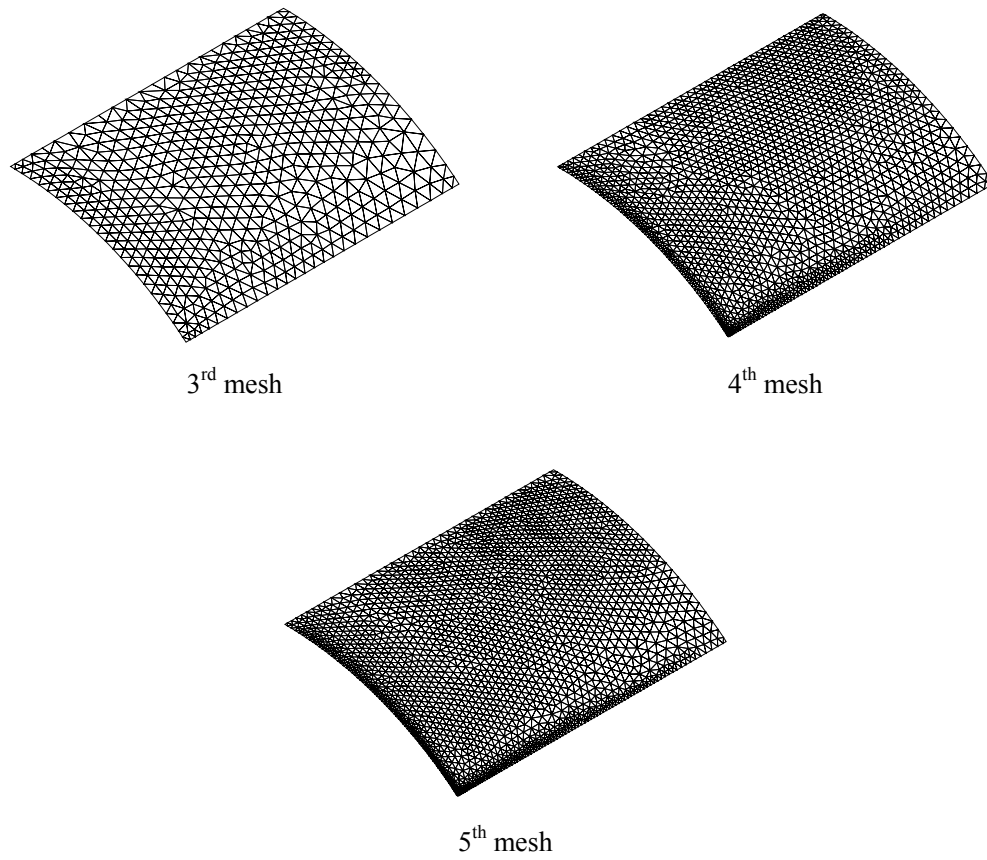
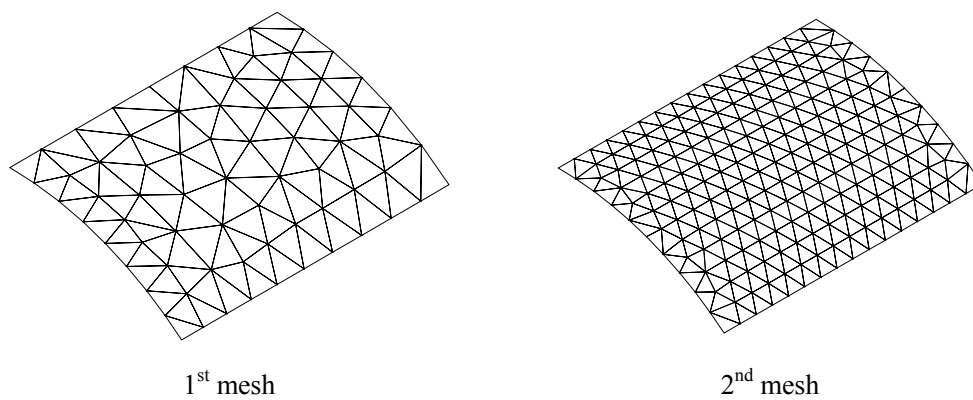


Figure 6.28 Example 6.5: adaptive refinement



Adaptive refinement

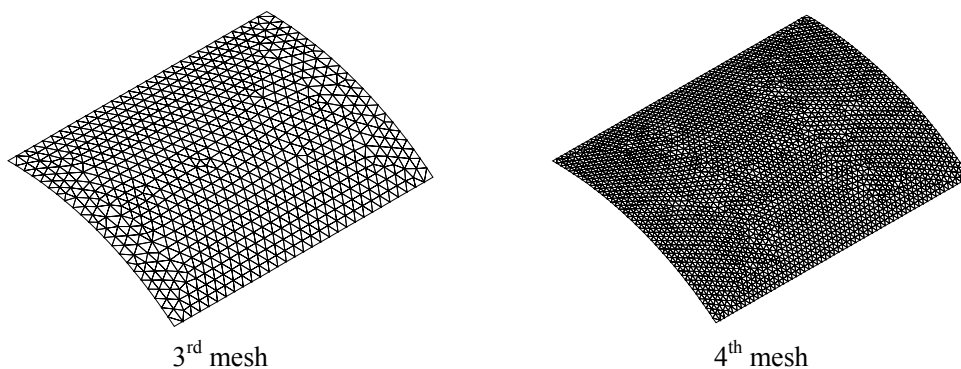


Figure 6.29 Example 6.5: uniform refinement

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
A1	2709	4.98E-02	5.00E-02	7.06E-02	27.05%	883.20%	38.32%	0.7341	1.0602	0.9125
A2	6999	1.80E-02	2.50E-02	3.08E-02	9.76%	442.11%	16.73%	0.8869	0.9318	0.9168
A3	24687	5.76E-03	1.01E-02	1.16E-02	3.13%	178.49%	6.32%	0.9187	0.8824	0.8914
A4	69621	2.67E-03	3.58E-03	4.47E-03	1.45%	63.23%	2.43%	0.9740	1.2561	1.1631
A5	90753	2.12E-03	2.74E-03	3.46E-03	1.15%	48.28%	1.88%	0.8751	1.0376	0.9796

Table 6.16 Example 6.5: results of the adaptive refinement scheme

Mesh	NTD	$\ e\ $			η			θ		
		surf	normal	overall	surf	normal	overall	surf	normal	overall
U1	2709	4.98E-02	5.00E-02	7.06E-02	27.05%	883.20%	38.32%	0.7341	1.0602	0.9125
U2	8367	1.61E-02	2.44E-02	2.92E-02	8.74%	429.88%	15.85%	1.4410	1.0746	1.1980
U3	30327	5.50E-03	9.98E-03	1.14E-02	2.99%	176.07%	6.18%	1.1781	1.0512	1.0822
U4	120567	2.79E-03	3.52E-03	4.49E-03	1.52%	62.11%	2.44%	0.9455	1.3054	1.1795

Table 6.17 Example 6.5: results of the uniform refinement scheme

Adaptive refinement

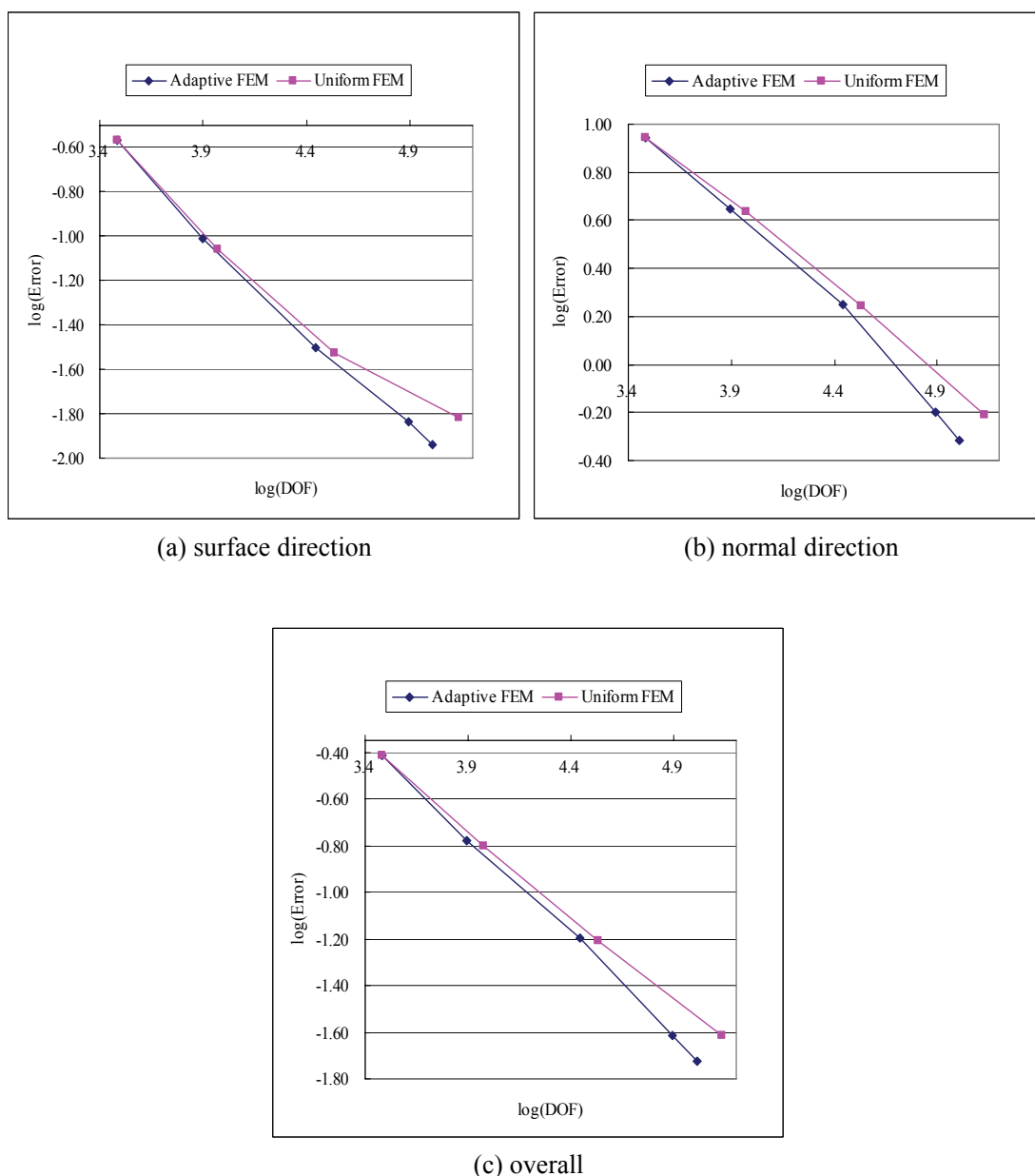


Figure 6.30 Example 6.5: comparison of convergence rate

The results shown in Table 6.16, Table 6.17 and Figure 6.30 show that the convergence rate of ARS is higher than that of URS in both the surface direction and normal direction. Further more, the overall effectivity index of ARS is much better than that of URS. Since the roof is soft-simply supported at edge DA (Figure 6.26) and the roof is thin, a strong boundary layer effect can be found in Figure 6.28. Due to the boundary layer effect, the error in normal direction is much bigger than the

error in surface direction. In order to reduce the error in normal direction, smaller elements would be needed along the free and soft-simply supported edges of the roof. In shell analysis, similar results were obtained in reference [111].

6.4.2 Examples of structures formed by multi-surface

In this section, several examples constructed by multi-surface are tested. Examples 6.6 and 6.7 are formed by two intersected flat plates. These two examples are similar but only the construction at the surface junctions is different. Example 6.8 is constructed by a curved surface and a plate while Example 6.9 is formed by two curved surfaces. All these examples are used to demonstrate that the presented adaptive refinement scheme can be successfully applied to the TWSs formed by intersected surfaces.

For all the examples formed by multi-surface, the computational time for both ARS and URS will be listed in tables. In order to compare the computational time for different examples, all the examples are carried out on the same computer which is equipped with an AMD ATHLON 64 3800+ X2 processor and 4096MB on board memory. It should be mentioned that because the P13 element is defined by the author using ABAQUS user defined subroutine, it cannot perform as fast as the standard elements which are already integrated into ABAQUS element library. Therefore, compared to a mesh with similar DOFs in URS, more computational time is needed for the mesh in ARS.

Example 6.6 is a structure with two intersected plates. The vertical plate is fixed on side AG and the horizontal plate is restricted in z-direction movement on side IC (Figure 6.31). Exploiting symmetry, only a half of the structure is analyzed. The detailed boundary conditions of the half are shown in Figure 6.32. The structure is subjected to a uniform pressure loading on one of the plates. The value of the

Adaptive refinement

pressure is 1g/mm^2 . The thickness of the structure is set to 1mm.

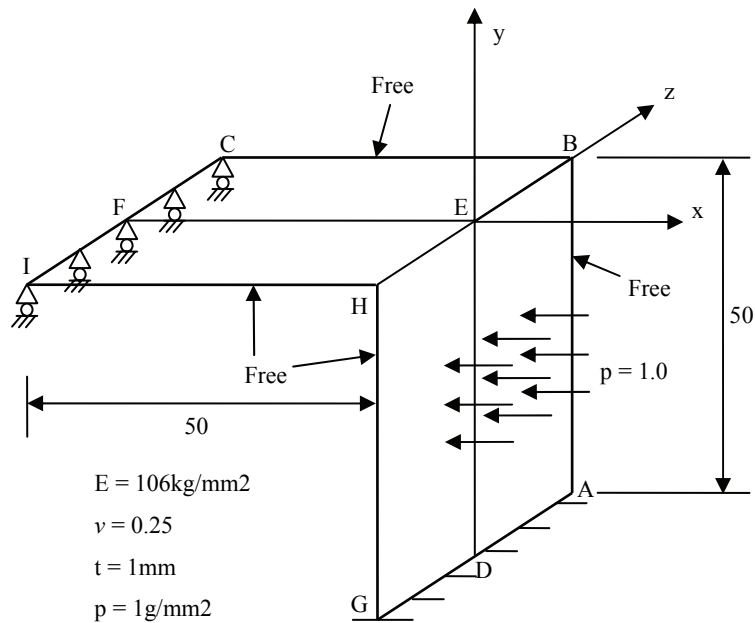


Figure 6.31 Example 6.6: two intersected plates under horizontal pressure

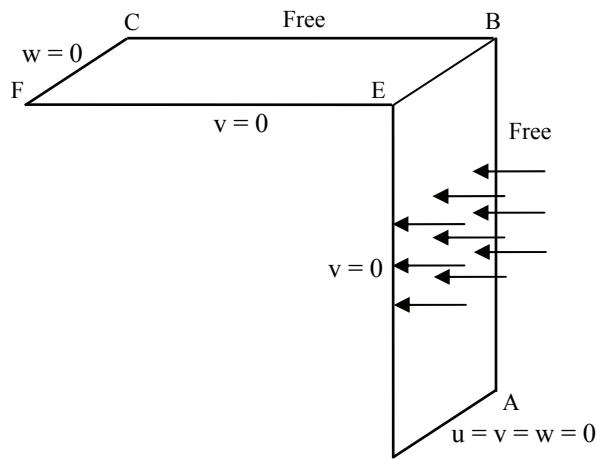


Figure 6.32 Example 6.6: boundary conditions

Adaptive refinement

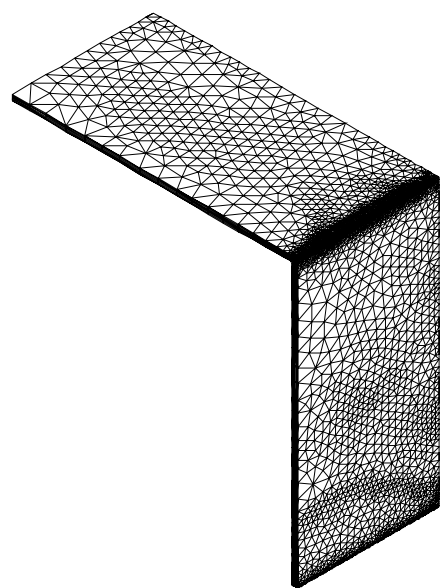
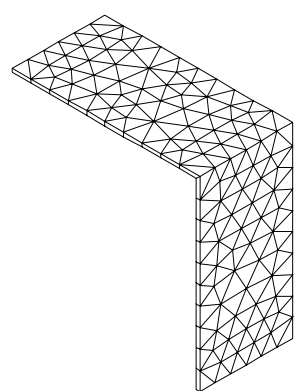


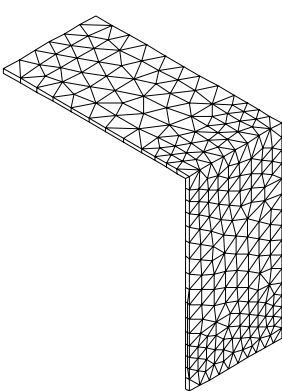
Figure 6.33 Example 6.6: reference mesh

NTD	$\ u\ $					η				
	surf	normal	shell	solid	overall	surf	normal	shell	solid	overall
892248	5.15E+01	2.43E+00	5.15E+01	6.72E+00	5.20E+01	0.61%	22.80%	1.24%	10.81%	1.86%

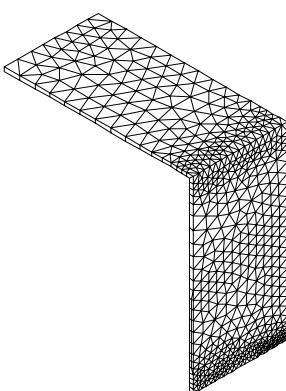
Table 6.18 Example 6.6: details of the reference mesh



1st mesh



2nd mesh



3rd mesh

Adaptive refinement

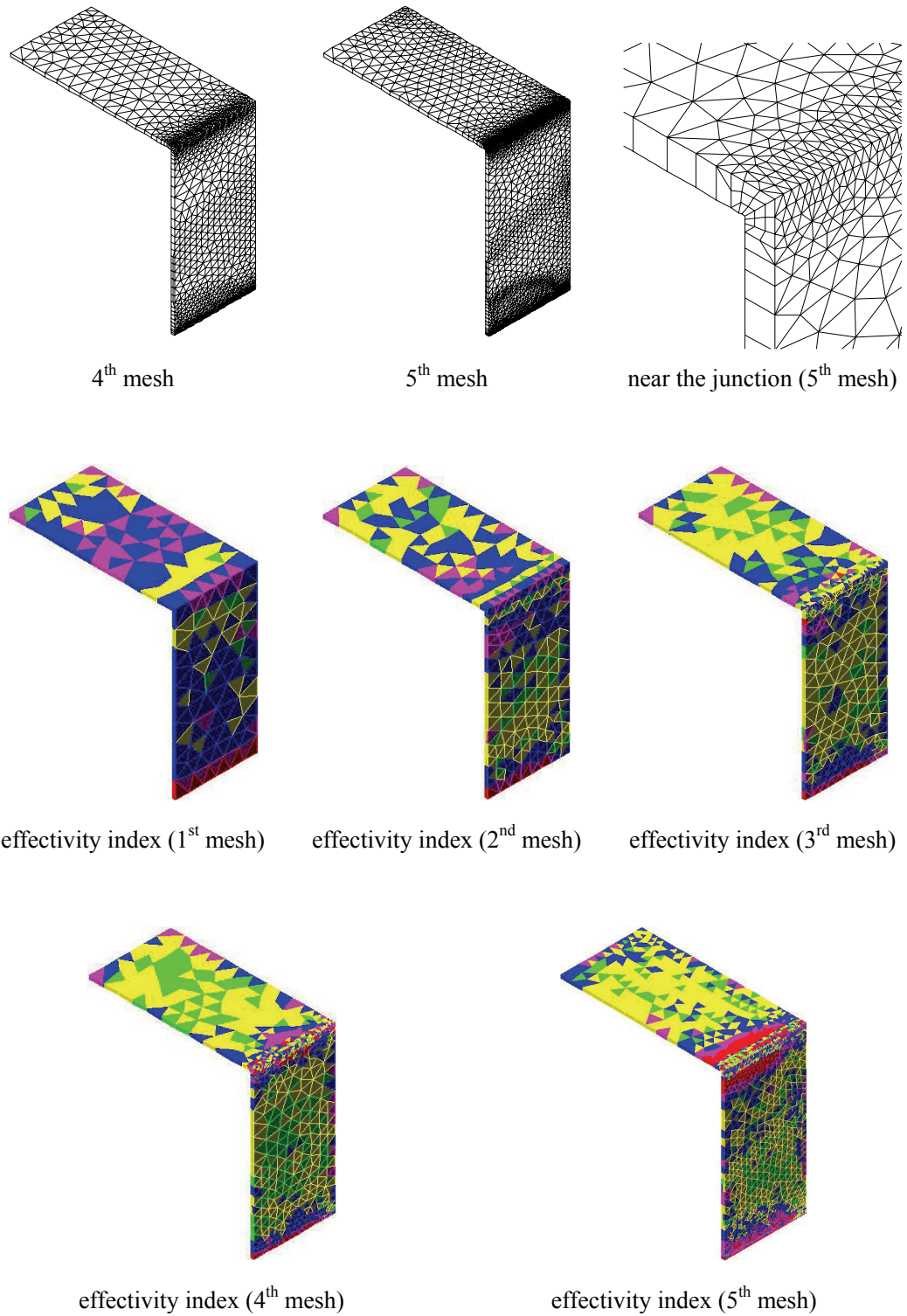
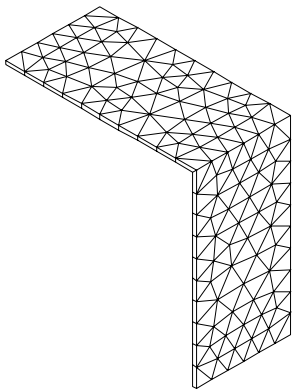
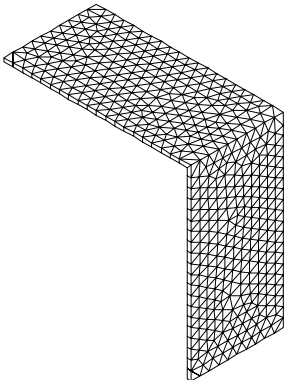


Figure 6.34 Example 6.6: adaptive refinement

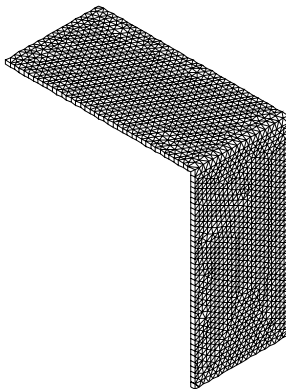
Adaptive refinement



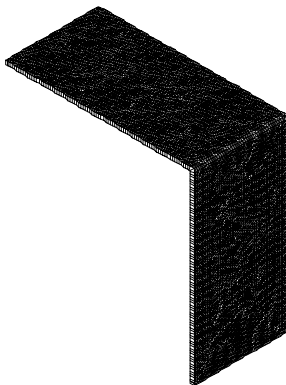
1st mesh



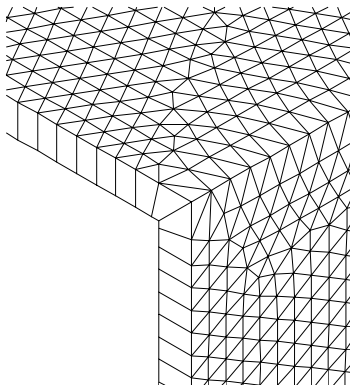
2nd mesh



3rd mesh



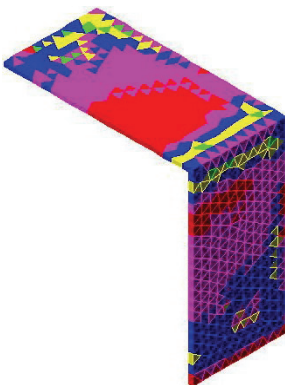
4th mesh



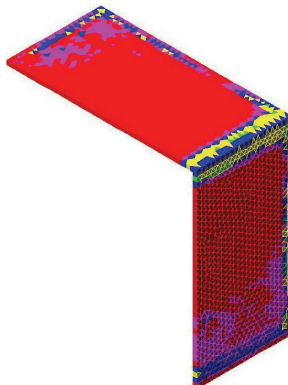
near the junction (4th mesh)



effectivity index (1st mesh)



effectivity index (2nd mesh)



effectivity index (3rd mesh)

Adaptive refinement

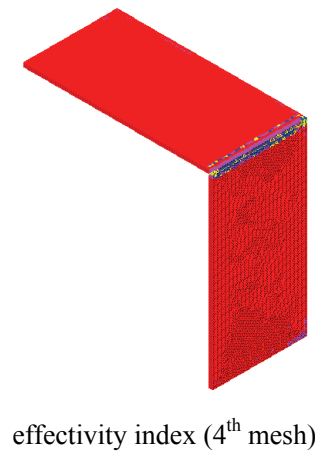


Figure 6.35 Example 6.6: uniform refinement

The target error for this case is set to 5%. Since this structure consists of a junction, the error should be analyzed in shell part and solid part separately. The shell error will be further divided into surface error and normal error. In this example, the target error of the shell zone is set to 5%, and the target errors in surface direction and normal direction of the shell zone are set to 5% and 50% respectively. The target error of the junction zone is set to 25%. A reference mesh (Figure 6.33 and Table 6.18) with 892248 DOFs is used for this example. The meshes and graphics of effectivity index distribution for each refinement step are shown in Figure 6.34 and Figure 6.35. Results of ARS and URS are listed in Table 6.19 and Table 6.20. The computational time of ARS and URS is shown in Table 6.21.

Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	overall	surf	normal	shell	junction	overall
A1	3423	6.60E+00	5.71E+00	8.73E+00	9.19E+00	1.27E+01	12.82%	234.59%	16.93%	136.87%	24.38%
A2	6927	4.96E+00	3.22E+00	5.91E+00	5.43E+00	8.03E+00	9.63%	132.17%	11.47%	80.82%	15.44%
A3	9309	3.98E+00	2.65E+00	4.78E+00	3.36E+00	5.84E+00	7.72%	109.03%	9.27%	49.98%	11.24%
A4	20793	2.89E+00	1.99E+00	3.51E+00	1.79E+00	3.94E+00	5.61%	81.61%	6.80%	26.63%	7.57%
A5	58260	9.63E-01	1.21E+00	1.54E+00	1.68E+00	2.29E+00	1.87%	49.59%	3.00%	25.09%	4.40%

Table 6.19 Example 6.6: results of the adaptive refinement scheme

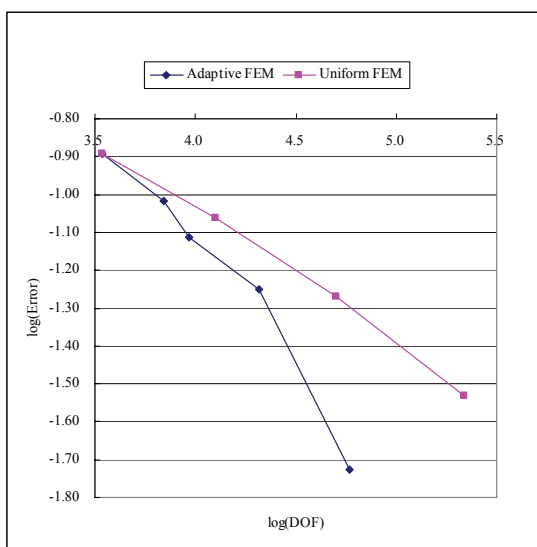
Adaptive refinement

Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	overall	surf	normal	shell	junction	overall
U1	3423	6.60E+00	5.71E+00	8.73E+00	9.19E+00	1.27E+01	12.82%	234.59%	16.93%	136.87%	24.38%
U2	12612	4.46E+00	2.37E+00	5.05E+00	6.09E+00	7.91E+00	8.67%	97.27%	9.80%	90.63%	15.22%
U3	50217	2.77E+00	1.32E+00	3.07E+00	3.71E+00	4.82E+00	5.39%	54.28%	5.96%	55.28%	9.27%
U4	218349	1.51E+00	1.20E+00	1.93E+00	2.11E+00	2.86E+00	2.93%	49.14%	3.74%	31.42%	5.50%

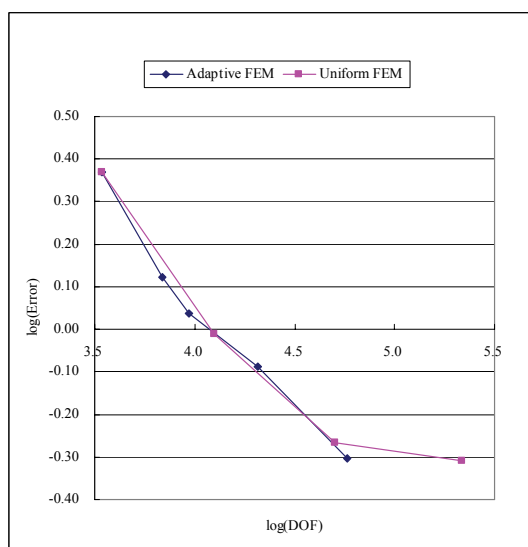
Table 6.20 Example 6.6: results of the uniform refinement scheme

Mesh	Mesh Generation	FE Analysis	Stress Recovery	Error Estimation	Adaptive Scheme Design	Total
A1	1	1	4	8	1	15
A2	1	2	8	18	1	30
A3	1	3	12	24	1	41
A4	2	11	27	56	1	97
A5	4	35	87	165	2	293
U1	1	1	4	8	/	14
U2	1	1	4	9	/	15
U3	1	2	14	35	/	52
U4	2	11	66	165	/	244

Table 6.21 Example 6.6: comparison of computational time (time unit: second)

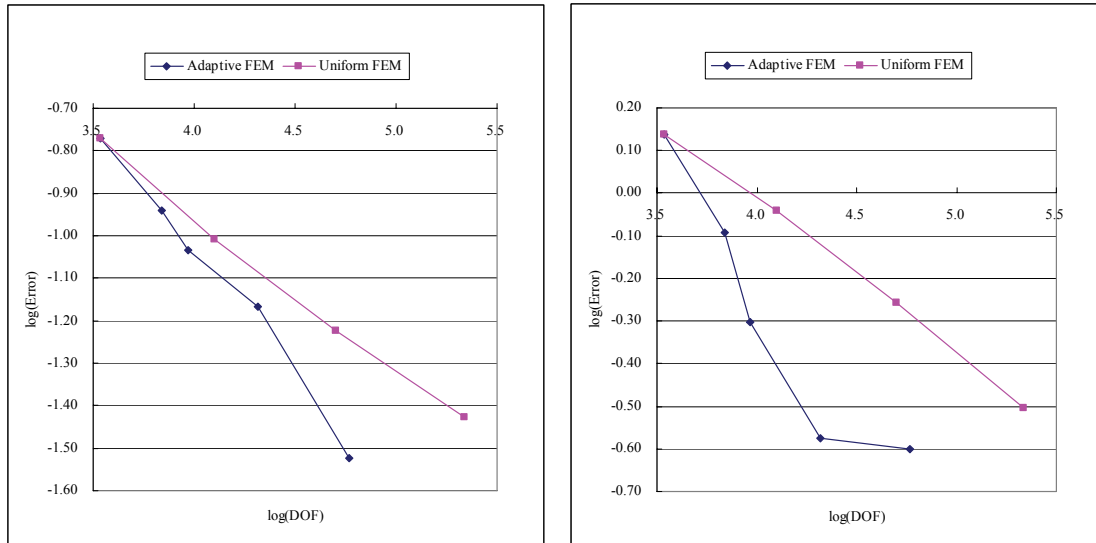


(a) shell zone: surface direction



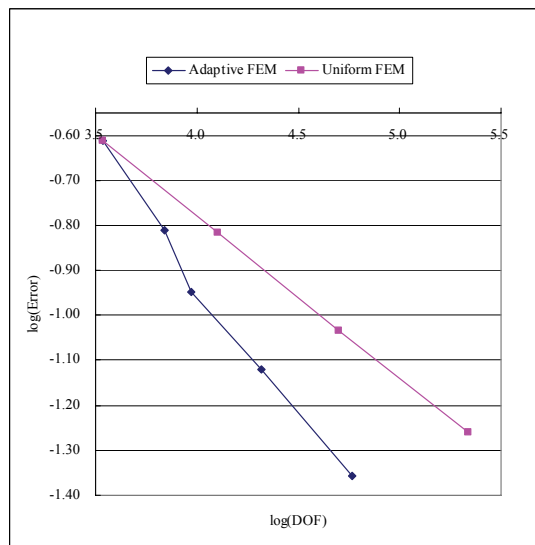
(b) shell zone: normal direction

Adaptive refinement



(c) shell zone: total

(d) junction zone



(e) whole mesh

Figure 6.36 Example 6.6: comparison of convergence rate

From Figure 6.34 it can be seen that most of the DOFs are devoted to the region near the junction because of the stress concentration. But since the stress values at the junction are near to be unbounded, it is still difficult to reduce the error of junction zone to a low level. From the graphics of effectivity index distribution shown in Figure 6.34 and Figure 6.35, it is clear that ARS can provide better θ

Adaptive refinement

values than URS. It is also found that in the picture of effectivity index distribution of URS, the θ values near the junction are better than the values elsewhere. Such a phenomenon will not happen if the exact solutions are available. For the θ values obtained from the reference mesh, since the error of the junction from the reference mesh is not small enough (in this example is about only a half of the target error), the junction error of the final mesh in URS is not much bigger than the one of the reference mesh, hence, the θ values obtained appear “good”. It is also found that the θ values along the boundary are worse than the values from the interior elements. This is because the recovered stresses of the boundary patches can not be as good as the ones of the interior patches, which has already been indicated in Section 5.3.2.4.

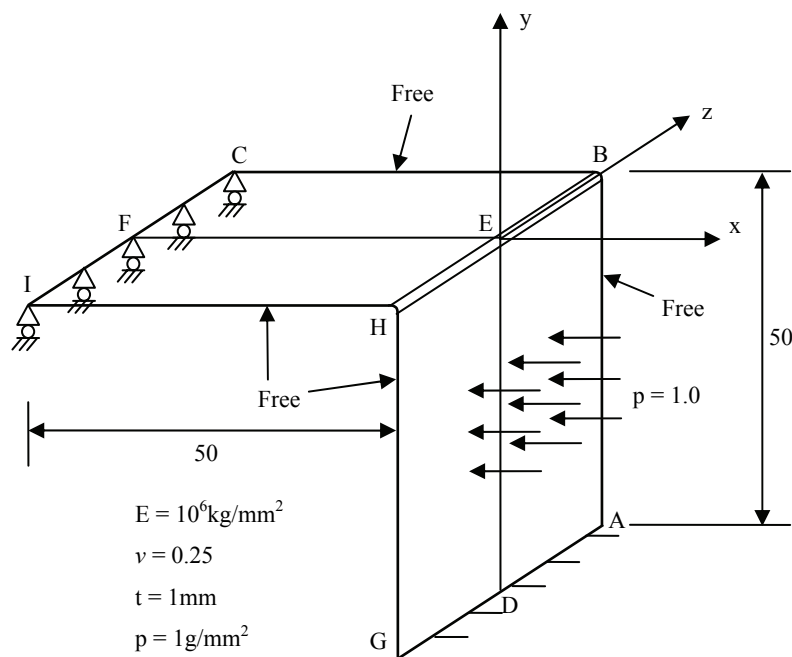


Figure 6.37 Example 6.7: two intersected plates under horizontal pressure

Example 6.7 is derived from the previous example. In Figure 6.31, the two plates are connected directly so that there is a right angle formed at the junction, which will result in singular solutions along the junction. In this example, a smooth connection is used instead of the direct connection (Figure 6.37). Now, the two

Adaptive refinement

plates are connected by a cylindrical shell with the radius of 1 (Figure 6.38). The boundary condition and the load are not changed.

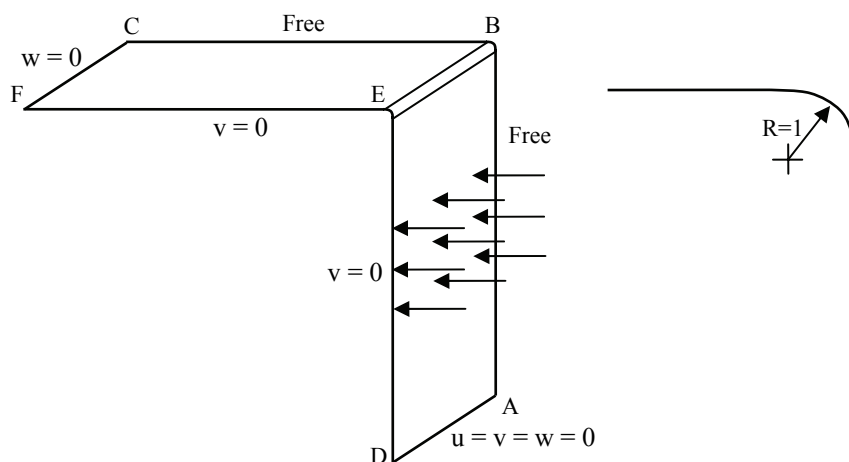


Figure 6.38 Example 6.7: boundary conditions

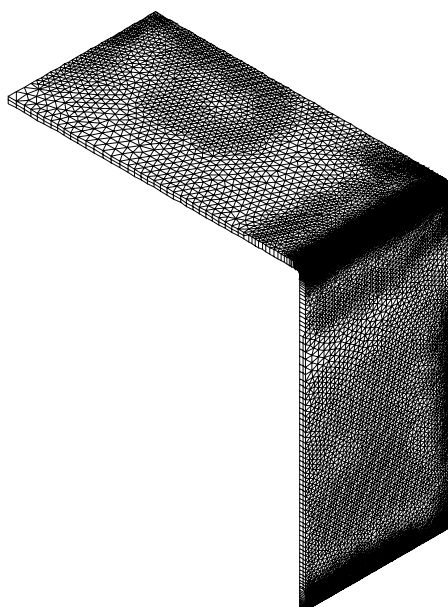


Figure 6.39 Example 6.7: reference mesh

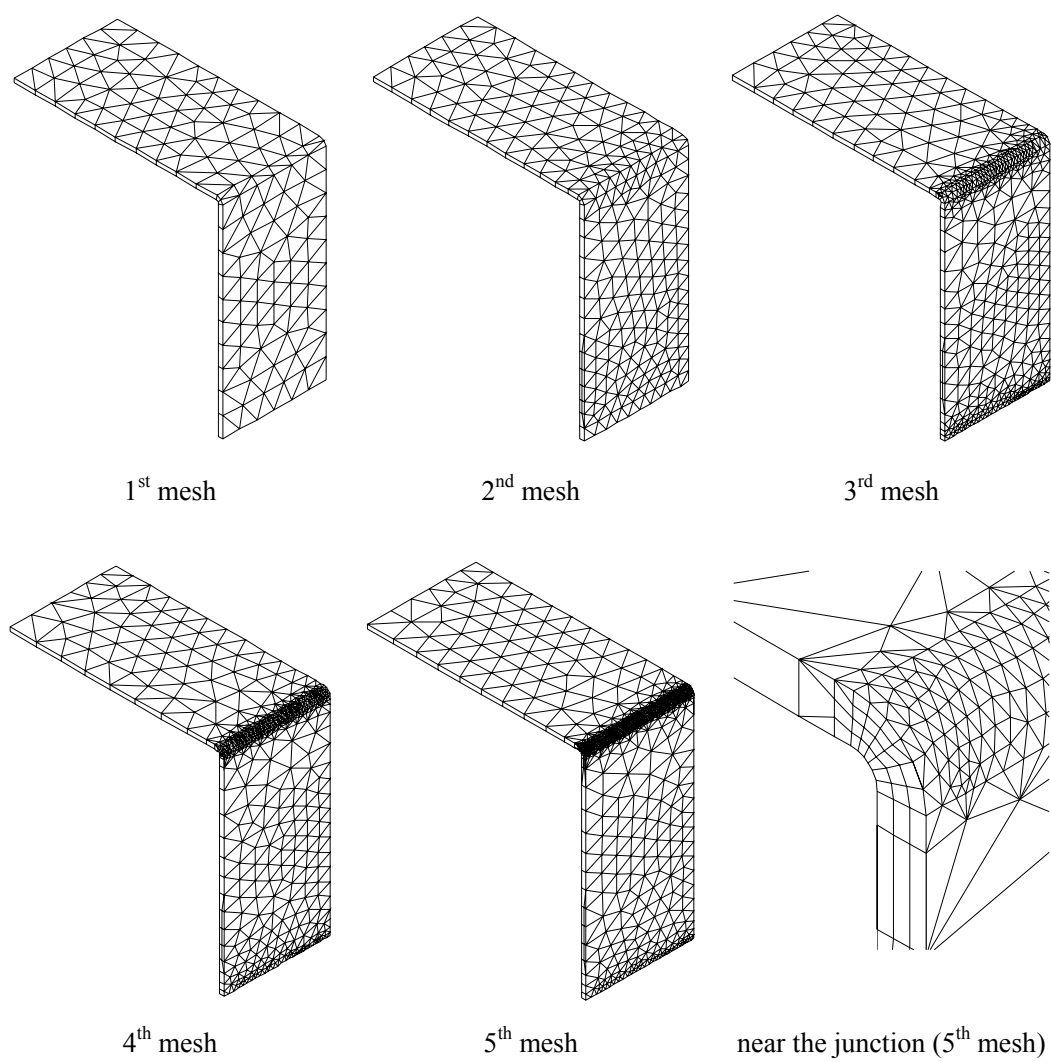
The target error for this case is set to 5%. The target error of the shell zone is set to 10%, and the target errors in shell surface direction and shell normal direction are set to 5% and 50% respectively. The target error of the junction zone is set to 10%.

Adaptive refinement

A reference mesh (Figure 6.39 and Table 6.22) with 1097796 DOFs is used for this example. The meshes and graphics of effectivity index distribution for each refinement step are shown in Figure 6.40 and Figure 6.41. The corresponding results are given in Table 6.23 and Table 6.24. The computational time spent on ARS and URS is listed in Table 6.25.

NTD	$\ u\ $					η				
	surf	normal	shell	solid	overall	surf	normal	shell	solid	overall
1097796	4.83E+01	2.37E+00	4.83E+01	6.70E+00	4.87E+01	0.81%	33.65%	1.84%	3.27%	1.88%

Table 6.22 Example 6.7: details of the reference mesh



Adaptive refinement

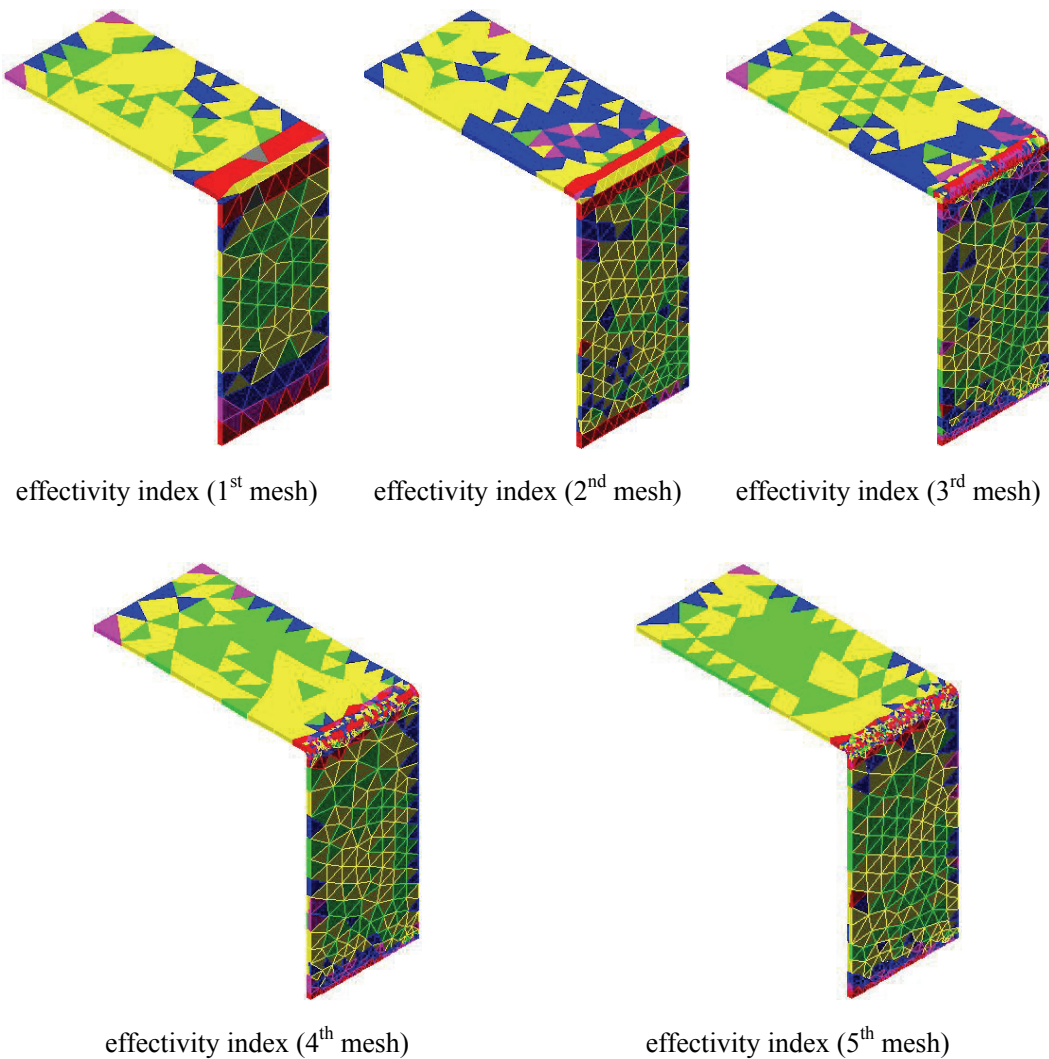
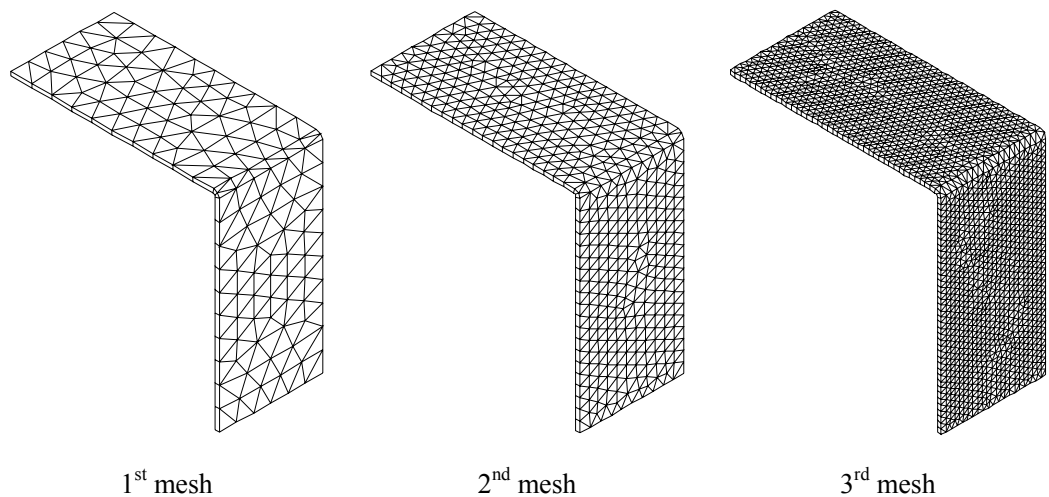


Figure 6.40 Example 6.7: adaptive refinement



Adaptive refinement

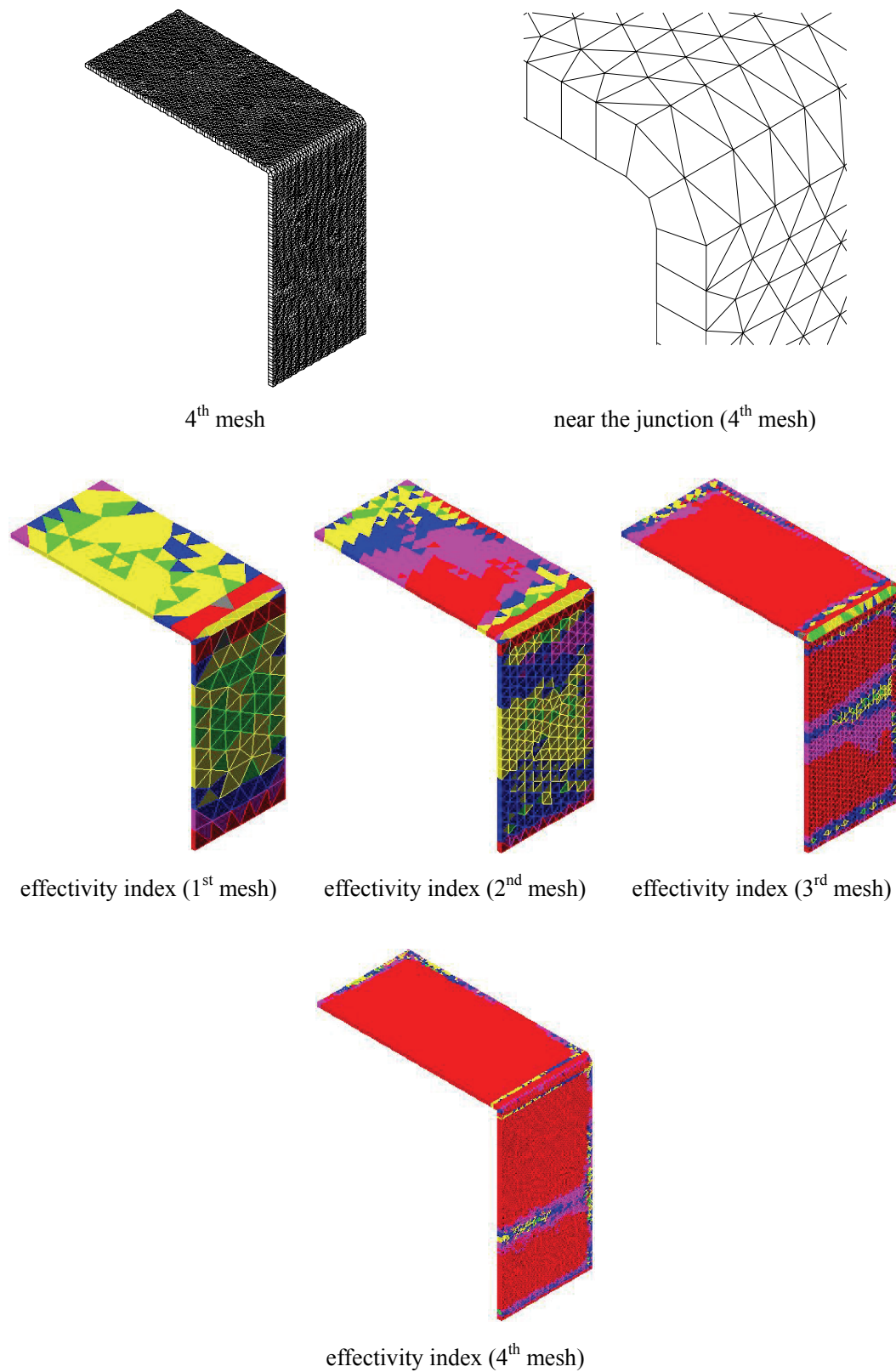


Figure 6.41 Example 6.7: uniform refinement

Adaptive refinement

Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	overall	surf	normal	shell	junction	overall
A1	3561	6.18E+00	5.01E+00	7.96E+00	4.35E+00	9.07E+00	12.81%	210.87%	16.49%	64.94%	18.62%
A2	7524	4.30E+00	2.50E+00	4.97E+00	4.55E+00	6.74E+00	8.90%	105.40%	10.31%	67.85%	13.84%
A3	13887	2.21E+00	2.37E+00	3.24E+00	2.05E+00	3.83E+00	4.57%	99.87%	6.71%	30.63%	7.87%
A4	21900	2.31E+00	2.42E+00	3.34E+00	7.00E-01	3.42E+00	4.78%	101.91%	6.93%	10.45%	7.01%
A5	59190	2.35E+00	2.50E+00	3.43E+00	4.37E-01	3.46E+00	4.86%	105.47%	7.12%	6.52%	7.10%

Table 6.23 Example 6.7: results of the adaptive refinement scheme

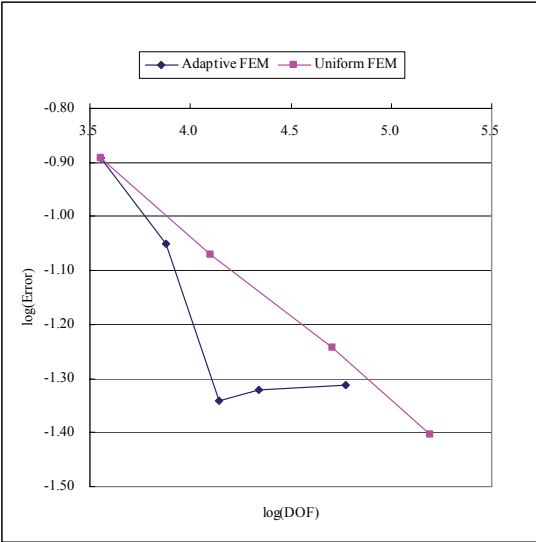
Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	all	surf	normal	shell	junction	all
U1	3561	6.19E+00	5.00E+00	7.96E+00	4.35E+00	9.07E+00	12.82%	210.71%	16.49%	64.86%	18.61%
U2	12669	4.11E+00	1.72E+00	4.46E+00	4.58E+00	6.39E+00	8.52%	72.43%	9.24%	68.29%	13.12%
U3	50625	2.76E+00	1.03E+00	2.94E+00	4.81E+00	5.64E+00	5.71%	43.45%	6.10%	71.71%	11.57%
U4	156822	1.90E+00	1.02E+00	2.16E+00	1.76E+00	2.79E+00	3.94%	43.15%	4.48%	26.29%	5.73%

Table 6.24 Example 6.7: Results of the uniform refinement scheme

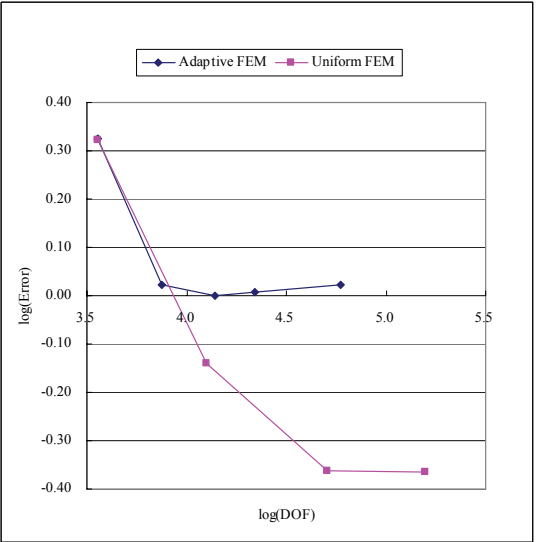
Mesh	Mesh Generation	FE Analysis	Stress Recovery	Error Estimation	Adaptive Scheme Design	Total
A1	1	1	3	8	1	14
A2	1	2	8	21	1	33
A3	1	3	14	35	1	54
A4	1	10	25	58	1	95
A5	3	41	83	187	1	315
U1	1	1	3	8	/	13
U2	1	2	14	38	/	55
U3	2	12	60	168	/	242
U4	8	51	191	523	/	773

Table 6.25 Example 6.7: comparison of computational time (time unit: second)

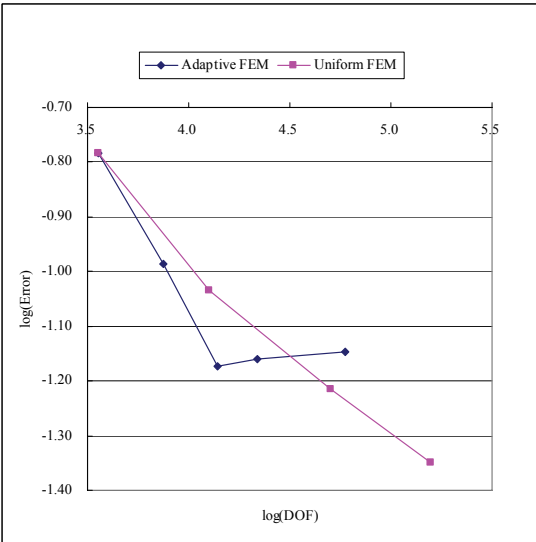
Adaptive refinement



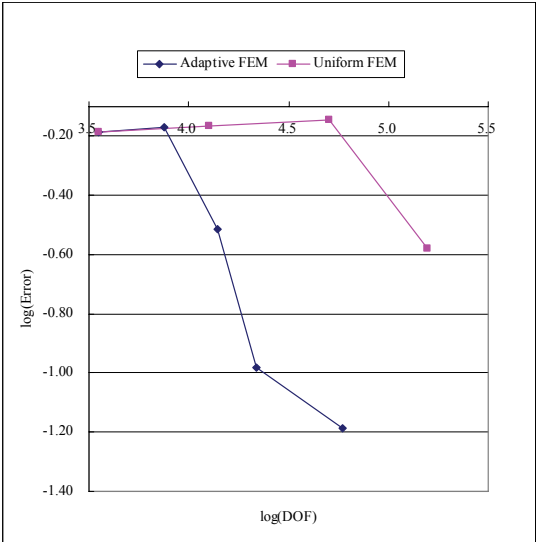
(a) shell zone: surface direction



(b) shell zone: normal direction

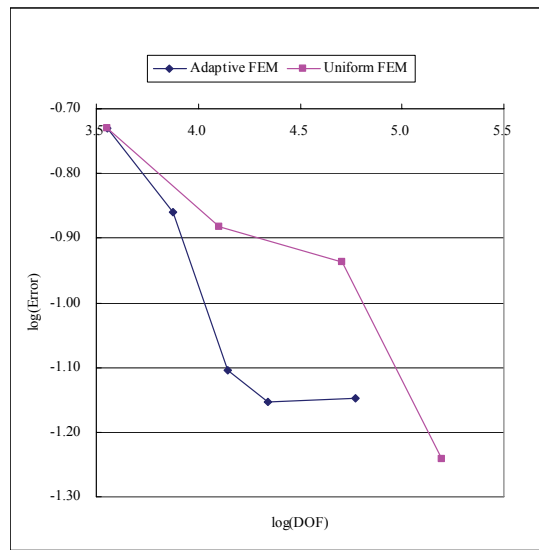


(c) shell zone



(d) junction zone

Adaptive refinement



(e) whole mesh

Figure 6.42 Example 6.7: comparison of convergence rate

From the results shown in Table 6.23 and Table 6.24, it is clear that compared to Example 6.6, the error of the junction zone becomes much smaller in this example. This is because that the strength of singularity is weakened for a smoothed connection. It is also found that the normal direction error is very hard to control. The reason is that the normal error of the reference mesh is not that small. Due to the limitation of the computer used, it is impossible to get a better reference mesh which can provide a good accuracy in normal direction. But considering that the structure is very thin and the energy contribution in normal direction is trivial, there is no need to care too much about the error in normal direction.

Example 6.8 is a water container, which is composed of a tube and a circular bottom plate. The dimension of this structure is shown in Figure 6.43. Exploiting symmetry, only a quarter of the structure is analyzed. The detailed boundary conditions of the quarter part are shown in Figure 6.44. The structure is subjected to an inner pressure from the water. The value of the pressure is liner distributed along the vertical direction, with the value from 0 to $0.98 \times 99 \text{ N/m}^2$ (Figure 6.43). The

Adaptive refinement

thickness of the structure is 2mm.

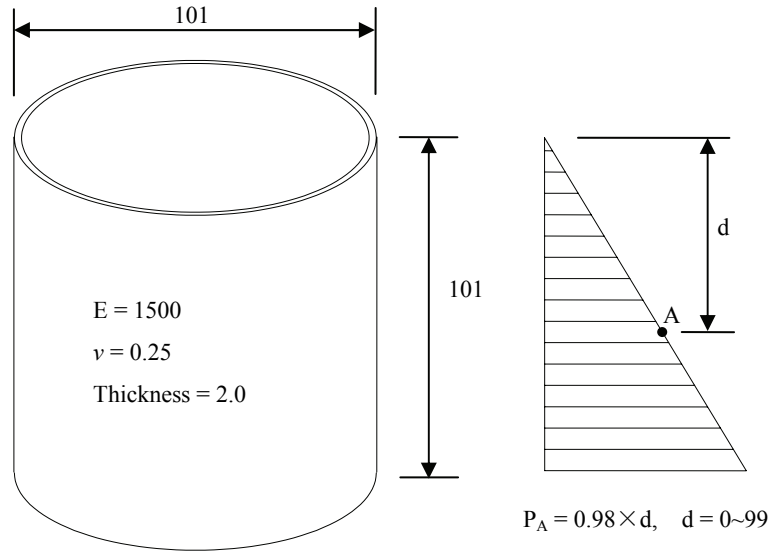


Figure 6.43 Example 6.8: two intersected plates under horizontal pressure

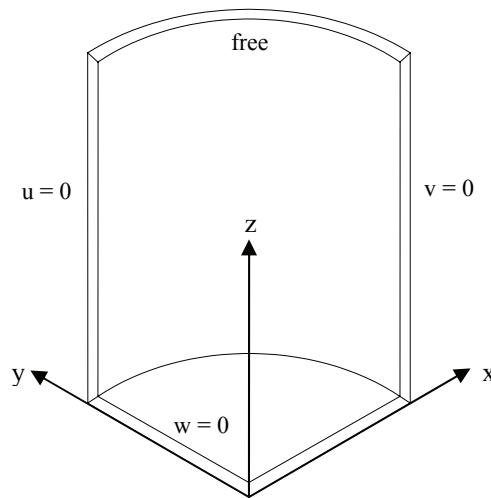


Figure 6.44 Example 6.8: boundary conditions of a quarter of the structure

The target total error for this example is set to 5%. The target error of the shell zone is set to 3%, and the target error of the junction zone is set to 20%. The target errors in surface direction and normal direction of the shell zone are not defined, hence, they will be determined by the algorithms introduced in Section 6.2.2. A

Adaptive refinement

reference mesh (Figure 6.45 and Table 6.26) with 870009 DOFs is used for this example. The meshes for each refinement step are shown in Figure 6.46 and Figure 6.47. The corresponding results are given in Table 6.27 and Table 6.28. Table 6.29 shows the computational time spent on ARS and URS.

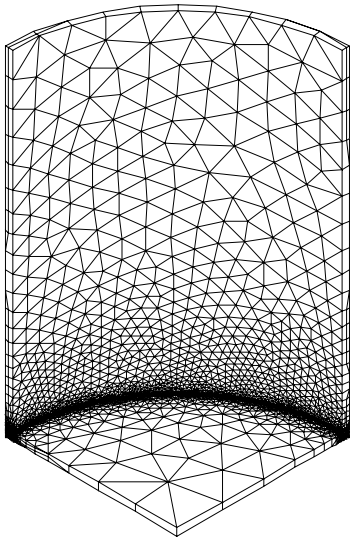
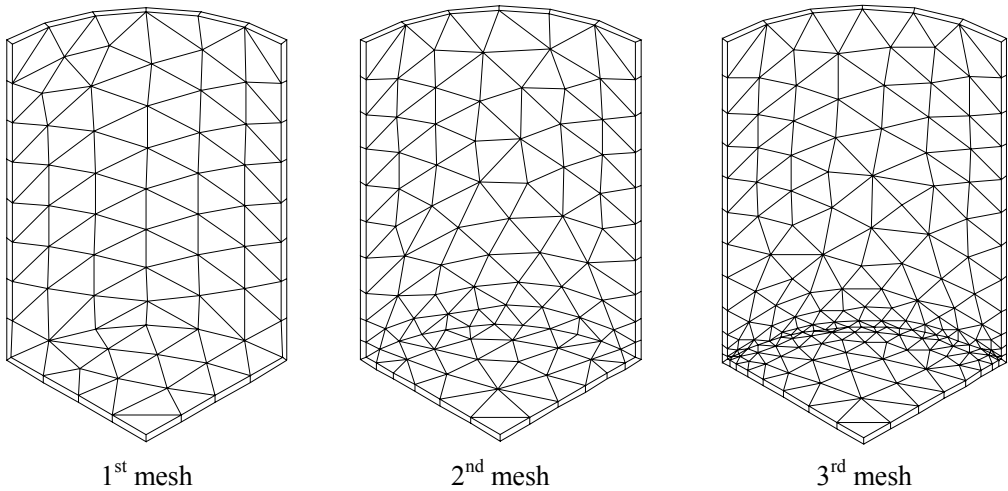


Figure 6.45 Example 6.8: reference mesh

NTD	$\ u(t)\ $					η				
	surf	normal	shell	solid	overall	surf	normal	shell	solid	overall
870009	3.99E+03	2.33E+02	4.00E+03	6.71E+02	4.05E+03	0.41%	17.46%	1.10%	10.94%	2.11%

Table 6.26 Example 6.8: details of the reference mesh



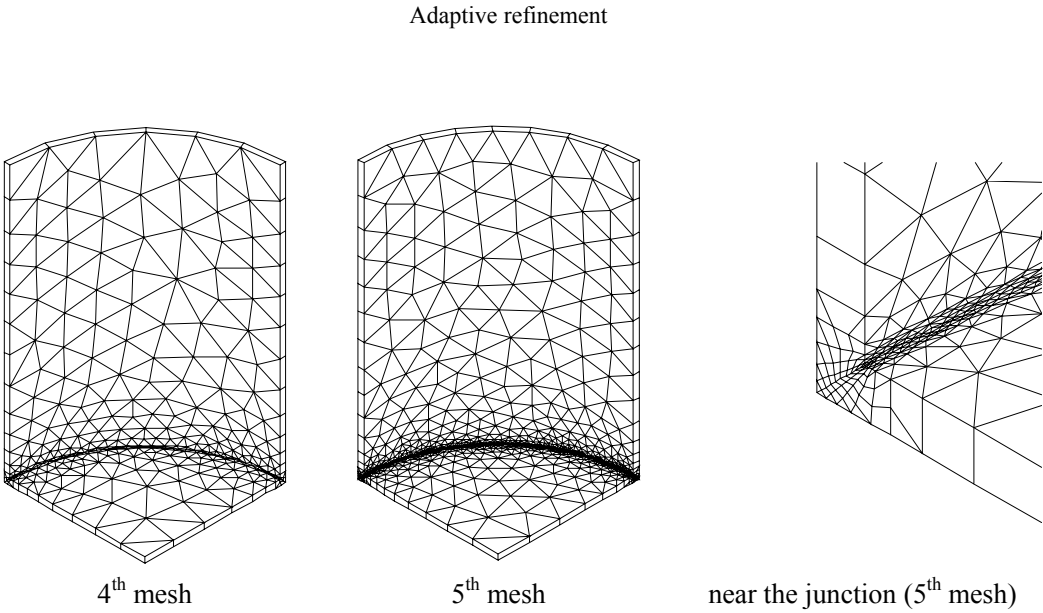


Figure 6.46 Example 6.8: adaptive refinement

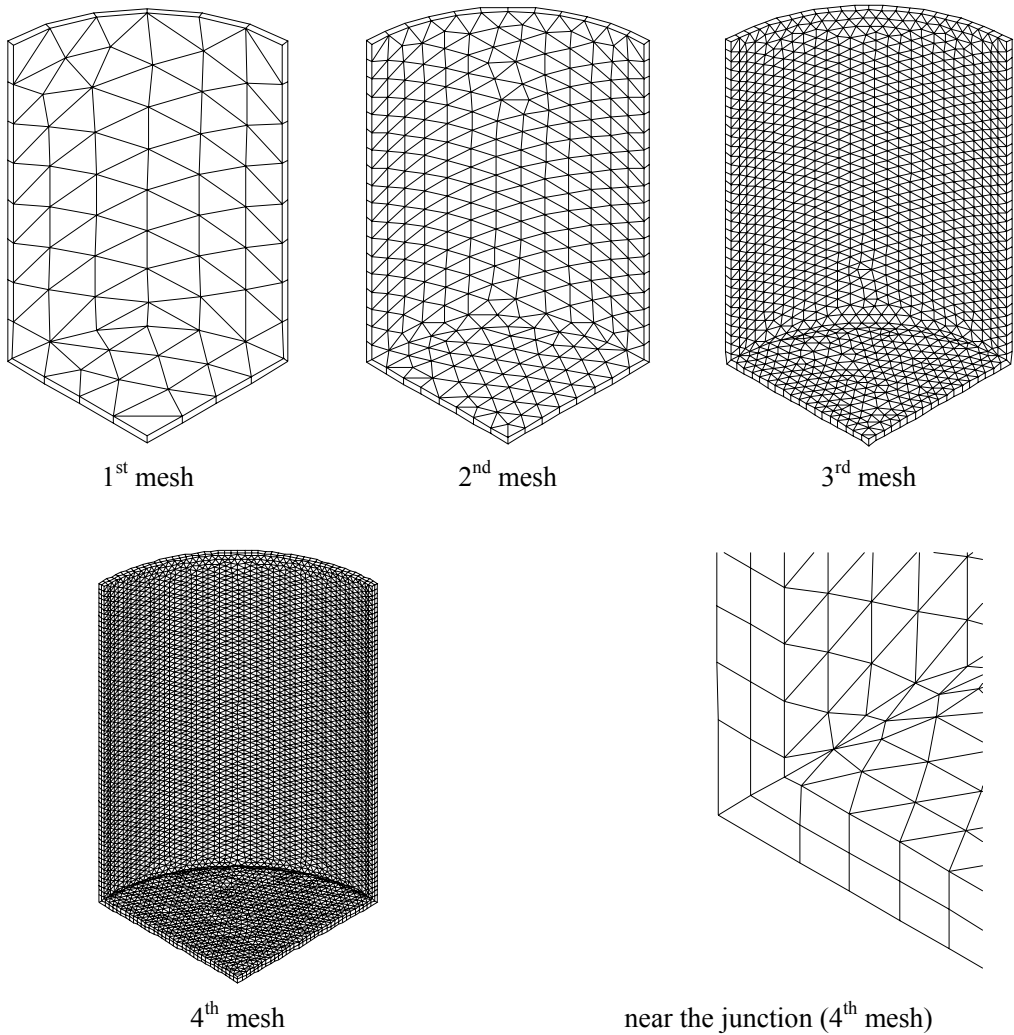


Figure 6.47 Example 6.8: uniform refinement

Adaptive refinement

Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	overall	surf	normal	shell	junction	overall
A1	1827	4.47E+02	2.61E+02	5.18E+02	1.23E+03	1.34E+03	11.21%	112.20%	12.95%	183.71%	32.96%
A2	4566	1.68E+02	1.79E+02	2.46E+02	4.13E+02	4.80E+02	4.22%	77.05%	6.15%	61.56%	11.85%
A3	8424	7.78E+01	1.15E+02	1.39E+02	2.66E+02	3.00E+02	1.95%	49.36%	3.47%	39.71%	7.40%
A4	22476	4.41E+01	9.41E+01	1.04E+02	1.84E+02	2.11E+02	1.10%	40.48%	2.60%	27.41%	5.21%
A5	80256	4.03E+01	8.28E+01	9.21E+01	1.27E+02	1.57E+02	1.01%	35.60%	2.30%	18.88%	3.86%

Table 6.27 Example 6.8: results of the adaptive refinement scheme

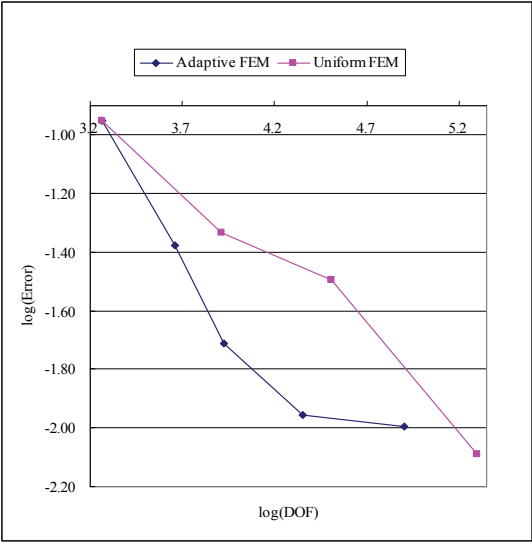
Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	overall	surf	normal	shell	junction	overall
U1	1827	4.47E+02	2.61E+02	5.18E+02	1.23E+03	1.34E+03	11.21%	112.20%	12.95%	183.71%	32.96%
U2	8157	1.86E+02	1.32E+02	2.28E+02	5.47E+02	5.93E+02	4.66%	56.77%	5.71%	81.54%	14.61%
U3	32073	1.27E+02	9.92E+01	1.61E+02	3.86E+02	4.18E+02	3.19%	42.68%	4.04%	57.52%	10.31%
U4	199011	3.25E+01	4.86E+01	5.84E+01	2.00E+02	2.09E+02	0.81%	20.88%	1.46%	29.90%	5.15%

Table 6.28 Example 6.8: results of the uniform refinement scheme

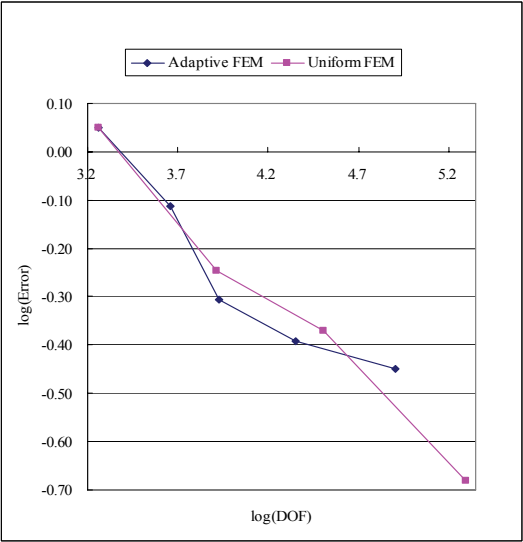
Mesh	Mesh Generation	FE Analysis	Stress Recovery	Error Estimation	Adaptive Scheme Design	Total
A1	1	1	3	6	1	12
A2	1	1	8	17	1	28
A3	1	3	14	33	1	52
A4	5	16	43	99	0	163
A5	26	74	188	533	2	823
U1	1	1	3	6	/	11
U2	1	3	9	25	/	38
U3	3	9	39	102	/	153
U4	24	118	334	800	/	1276

Table 6.29 Example 6.8: comparison of computational time (time unit: second)

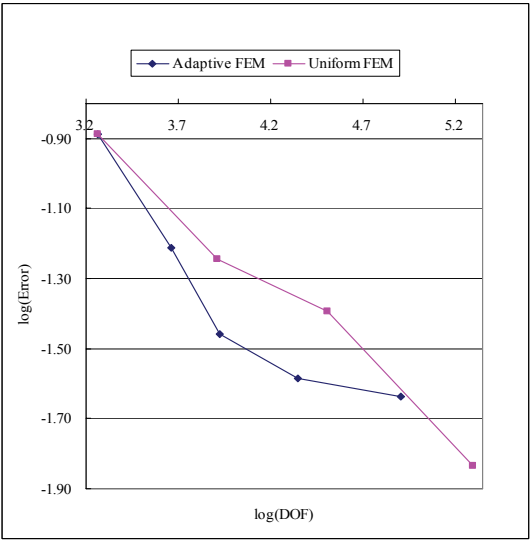
Adaptive refinement



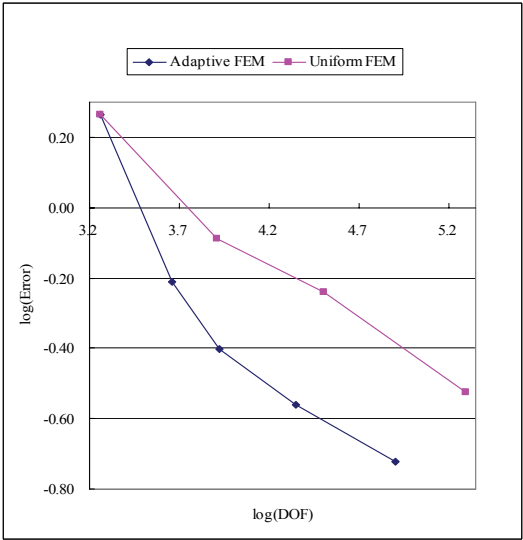
(a) shell zone: surface direction



(b) shell zone: normal direction

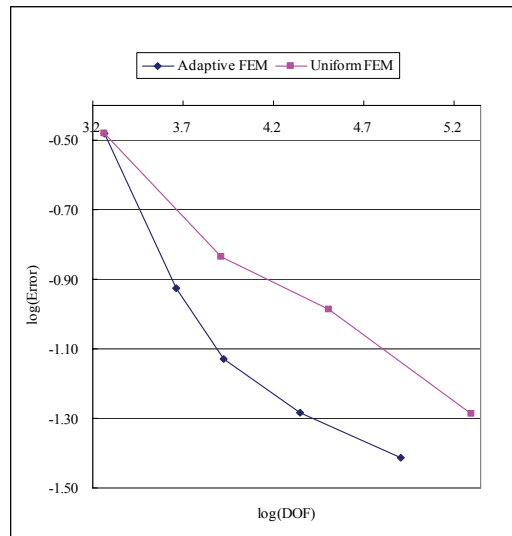


(c) shell zone



(b) junction zone

Adaptive refinement



(e) whole mesh

Figure 6.48 Example 6.8: comparison of convergence rate

From the results listed in Table 6.27 and Table 6.28, it can be seen that in order to achieve the target accuracy, 199011 DOFs are used for the last mesh in URS, while only 80256 DOFs are used for the last mesh in ARS. Hence, the convergence rate of ARS is higher than that of URS, which can also be observed in Figure 6.48. In addition, for the structure with junctions, ARS can catch the characteristic of the stress concentration near the junction so that most of the DOFs are given to the junction part.

The problem to be modeled in the last example is a tubular joint, which is composed of two intersected tubes. The dimension of this structure is shown in Figure 6.49. Exploiting symmetry, only a quarter of the structure is analyzed. The detailed boundary conditions of the quarter part are shown in Figure 6.50. The structure is subjected to a uniform pressure from the top of the vertical tube. The thickness of the structure is 1mm.

Adaptive refinement

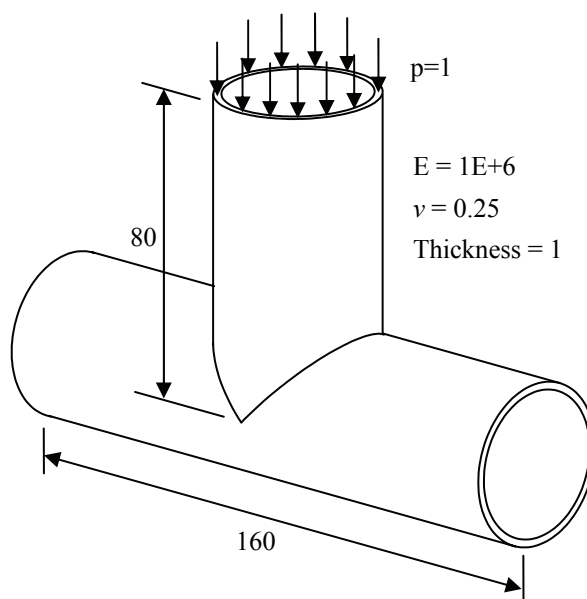


Figure 6.49 Example 6.9: two intersected plates under horizontal pressure

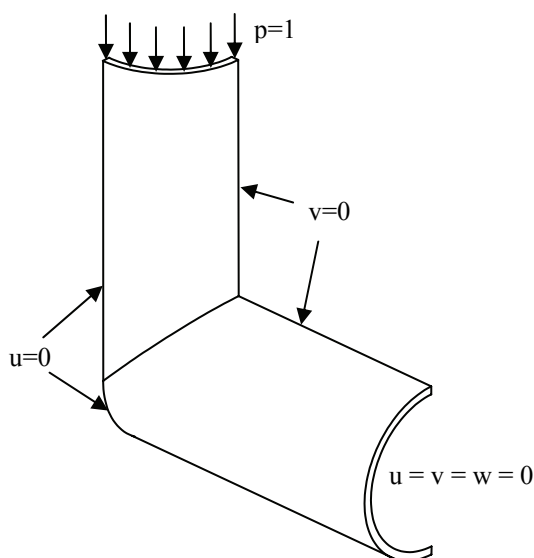


Figure 6.50 Example 6.9: boundary conditions of a quarter of the structure

In this example, only the target total error is given, which is 10%. The target error of the junction zone, the target error of the shell zone, the target errors in surface direction and normal direction of the shell zone will be determined by the presented ARS. A reference mesh (Figure 6.51 and Table 6.30) with 863958 DOFs is

Adaptive refinement

used for this example. The meshes for each refinement step are shown in Figure 6.52 and Figure 6.53. The corresponding results are given in Table 6.31 and Table 6.32. The computational time spent on this example is shown in Table 6.33.

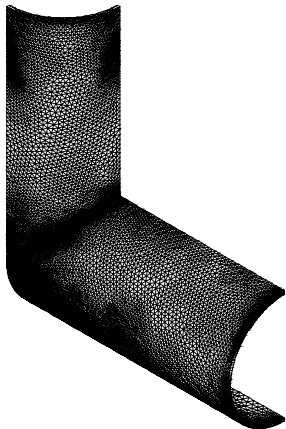
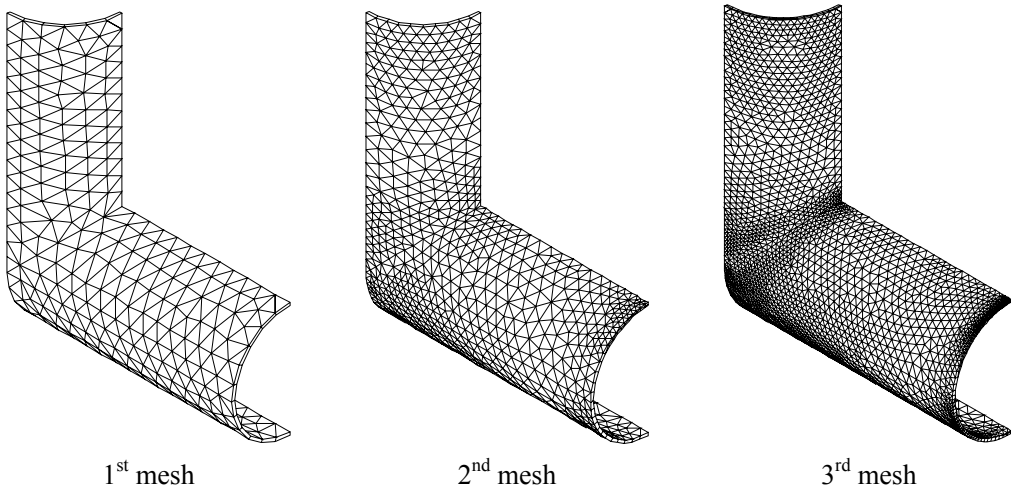


Figure 6.51 Example 6.9: reference mesh

NTD	$\ u(t)\ $					η				
	surf	normal	shell	solid	overall	surf	normal	shell	solid	overall
863958	1.40E-01	7.25E-03	1.40E-01	1.51E-02	1.41E-01	0.71%	17.53%	1.56%	9.23%	1.51%

Table 6.30 Example 6.9: details of the reference mesh



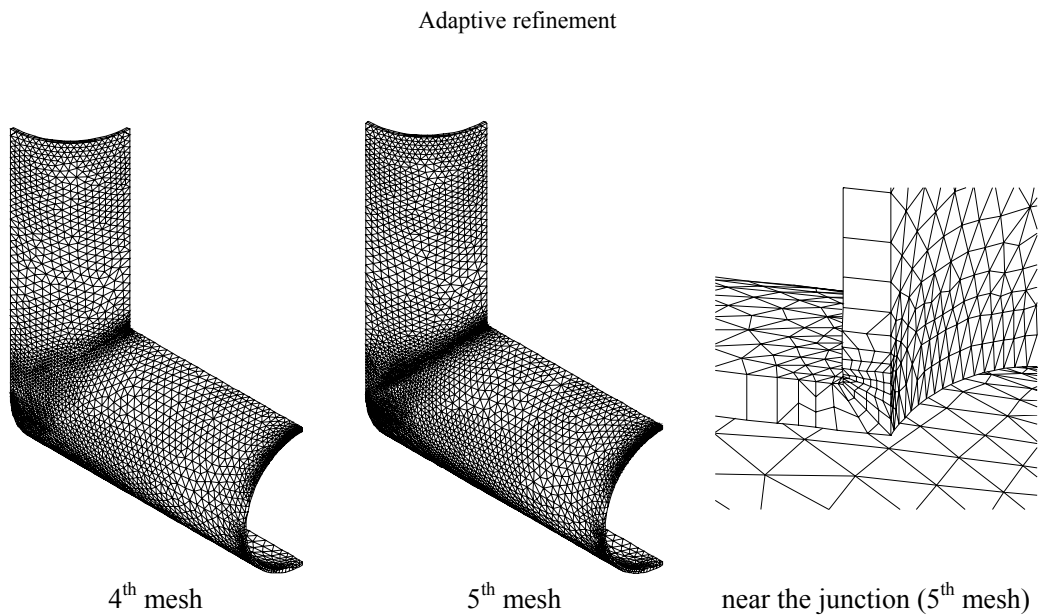


Figure 6.52 Example 6.9: adaptive refinement

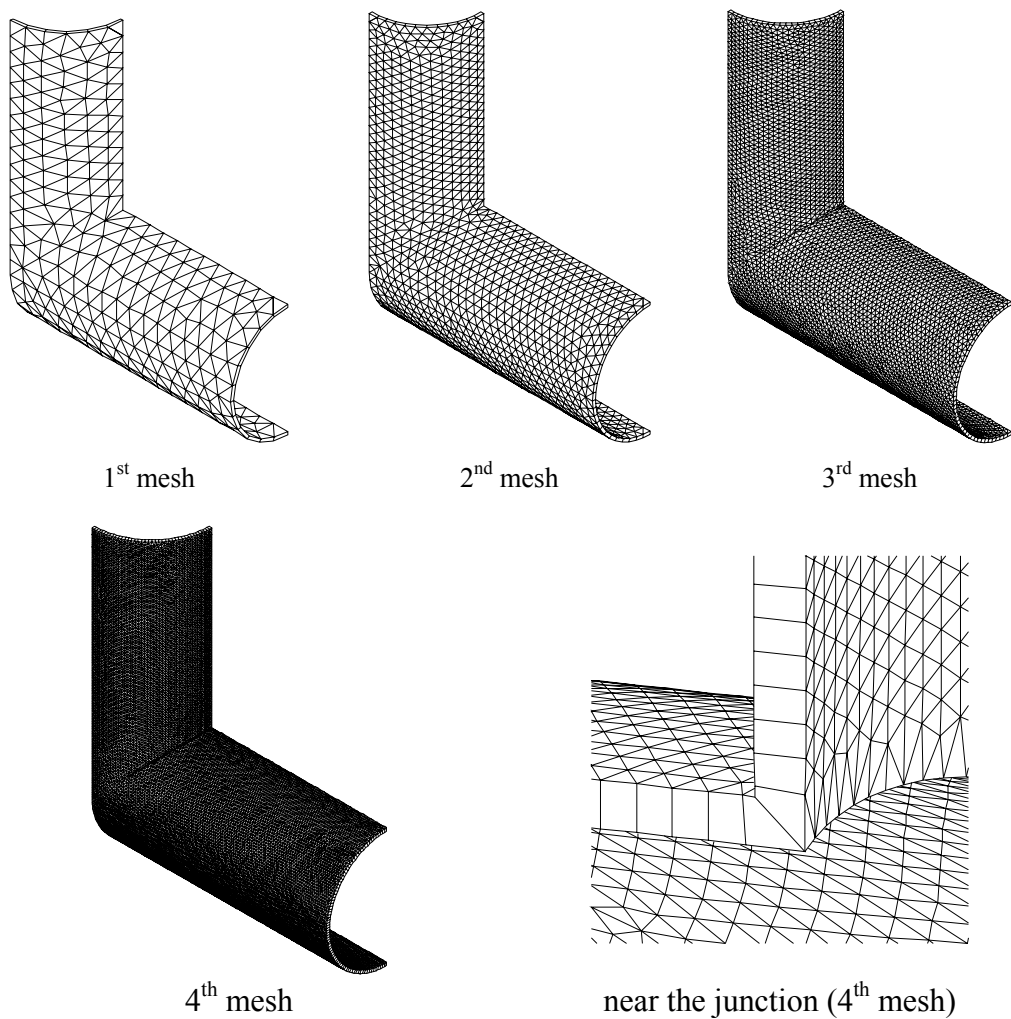


Figure 6.53 Example 6.9: uniform refinement

Adaptive refinement

Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	overall	surf	normal	shell	junction	overall
A1	7467	2.29E-02	1.37E-02	2.66E-02	2.37E-02	3.57E-02	16.37%	188.87%	19.05%	157.48%	25.36%
A2	24813	7.27E-03	5.56E-03	9.15E-03	1.17E-02	1.48E-02	5.21%	76.69%	6.54%	77.43%	10.54%
A3	87555	3.12E-03	2.83E-03	4.22E-03	7.66E-03	8.74E-03	2.24%	39.06%	3.01%	50.85%	6.22%
A4	115920	2.42E-03	2.30E-03	3.34E-03	6.05E-03	6.91E-03	1.73%	31.70%	2.39%	40.14%	4.91%
A5	153357	1.97E-03	1.98E-03	2.79E-03	3.61E-03	4.56E-03	1.41%	27.35%	2.00%	23.94%	3.24%

Table 6.31 Example 6.9: results of the adaptive refinement scheme

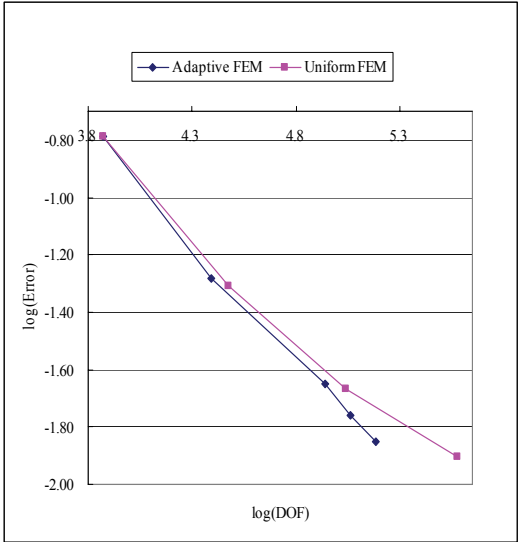
Mesh	NTD	$\ e\ $					η				
		surf	normal	shell	junction	all	surf	normal	shell	junction	all
U1	7467	2.29E-02	1.37E-02	2.66E-02	2.37E-02	3.57E-02	16.37%	188.87%	19.05%	157.48%	25.36%
U2	29976	6.85E-03	5.19E-03	8.59E-03	1.19E-02	1.46E-02	4.90%	71.55%	6.14%	78.69%	10.41%
U3	109665	3.00E-03	2.78E-03	4.09E-03	7.74E-03	8.75E-03	2.15%	38.33%	2.93%	51.39%	6.22%
U4	379194	1.75E-03	2.15E-03	2.77E-03	5.50E-03	6.16E-03	1.25%	29.67%	1.98%	36.54%	4.38%

Table 6.32 Example 6.9: results of the uniform refinement scheme

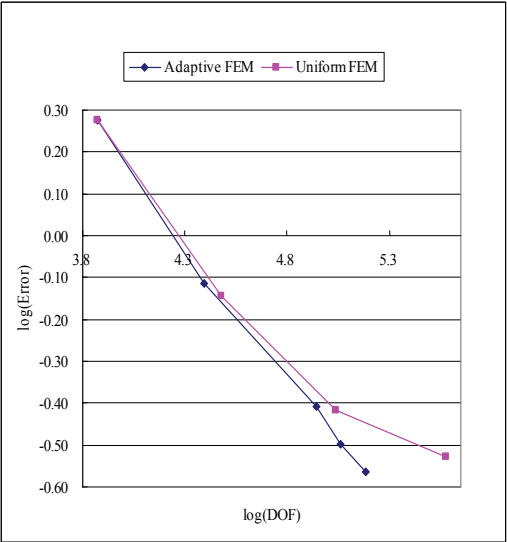
Mesh	Mesh Generation	FE Analysis	Stress Recovery	Error Estimation	Adaptive Scheme Design	Total
A1	1	1	7	18	1	28
A2	1	8	25	71	1	106
A3	4	37	103	292	2	438
A4	5	54	139	396	3	597
A5	9	84	188	507	4	792
U1	1	1	7	18	/	27
U2	1	7	35	95	/	138
U3	5	27	132	301	/	465
U4	20	129	457	1254	/	1860

Table 6.33 Example 6.9: comparison of computational time (time unit: second)

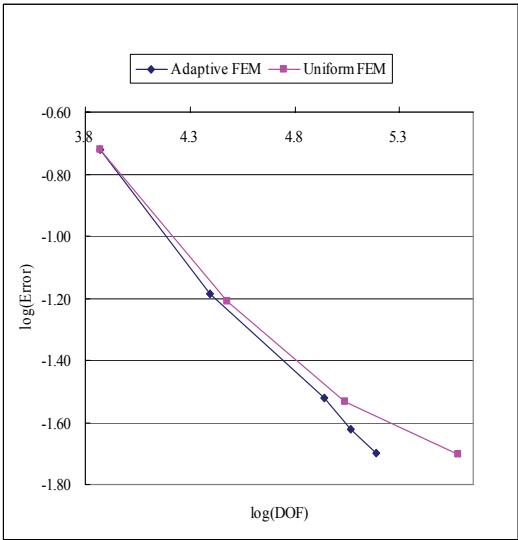
Adaptive refinement



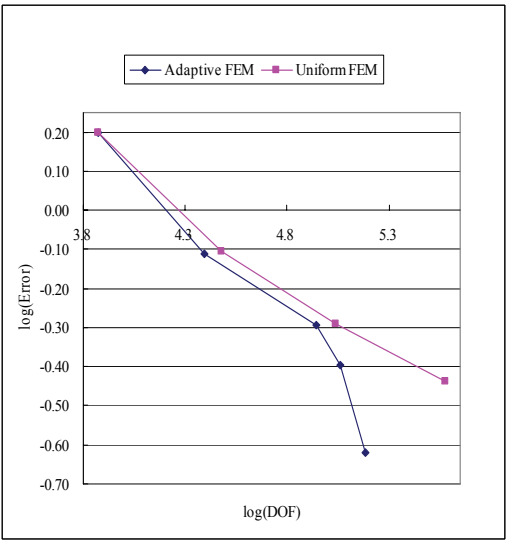
(a) shell zone: surface direction



(b) shell zone: normal direction

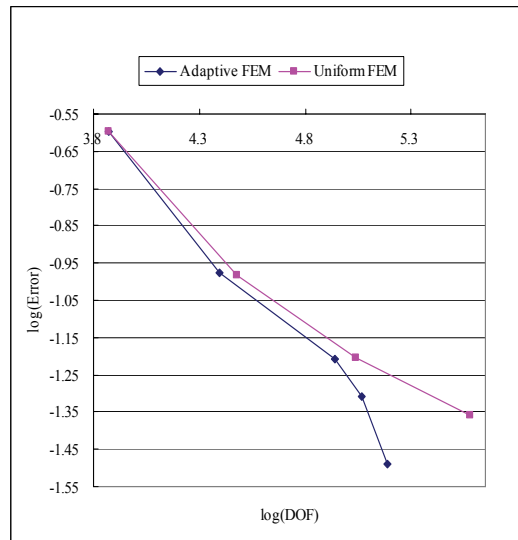


(c) shell zone



(b) junction zone

Adaptive refinement



(e) whole mesh

Figure 6.54 Example 6.9: comparison of convergence rate

From the results shown in Table 6.31 and Table 6.32 and Figure 6.54, it can be seen in the initial several steps, the convergence rate of ARS is only slightly higher than that of URS. After the third mesh, since the surface direction error of the shell zone is already small, the increase of DOFs is mainly devoted to the junction part. Hence both the junction error and the total error reduce fast. By comparing the last rows in Table 6.31 and Table 6.32, it can be found that for the relative error of the junction part, ARS can achieve a better results than URS while its DOFs is only less than a half of the DOFs of URS. Therefore, the adaptive refinement scheme is able to get a good refinement effect for the structure with a tubular joint.

6.5 Closure

In this chapter, focus is devoted to the design of the adaptive refinement scheme for TWSs. Different refinement schemes are presented for different kind of structures. For the structures without junction, the refinement is carried out along the mesh surface direction and the mesh thickness direction separately; for the structures with junctions, the local mesh in the junction zone will be refined as a

general 3D solid mesh. In order to control the errors in different directions or in different zones (i.e. shell zone or junction zone), the target error distribution algorithms are designed for the case that the user defined targets errors do not consist of all the part of errors. To get a better refinement effect and also in order to save the DOF, a mesh speed controlling technique is proposed to avoid over refinement for coarse initial meshes. Finally, special element size adjustment technique is used to avoid bad-shaped elements.

For the problems with no available exact solutions, in order to examine the performance of the adaptive refinement schemes, reference meshes are especially generated. The recovered stresses from the reference mesh can be approximately regarded as the exact stresses. Methods for retrieving the stresses from the reference mesh are discussed and the related algorithms are developed.

At the end of this chapter, examples are provided to show the efficiency of the proposed adaptive refinement schemes. The examples include some representative problems, from 2D problems to 3D problems, from plate structures to curved structures, and from single surface structures to multi-surface structures with junctions. By contrast, for each example, uniform refinement results are also generated. All the examples show that for TWSs, the adaptive refinement scheme proposed in this study surpasses the uniform refinement scheme obviously. By comparing the cost of DOFs and computational time from ARS and URS, it is obvious that the developed ARS can greatly save the computational cost and hence, it is more efficient than URS.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions for Current Work

In this thesis a new h-version adaptive refinement scheme has been developed for the analysis of TWS in elastostatic range. The basic concepts of adaptivity for the finite element method have been discussed and a brief review of the related theories and techniques has also been presented. Focus of this thesis is devoted to three essential ingredients for the whole adaptive refinement scheme: automatic mesh generation, error estimation on the FE solution and adaptive refinement scheme design.

7.1.1 Conclusion of the automatic mesh generation step

The originality of this part is the development a new 3D mesh generator for the TWS. Different from the general 3D mesh generator, in this study, the formation of the 3D mesh is divided into 2 steps. In the first step, a surface mesh is created for the middle surface of the TWS by any existing surface mesh generator. In the second step, a special designed algorithm is employed to convert the surface mesh to solid mesh. The new mesh generator has several characteristics which are listed as follows:

Conclusion and future work

1) Specially designed for TWSs

In the process of conversion from surface mesh to solid mesh, the main technique involved is surface extrusion (Chapter 3). This technique is specially designed for the generation of 3D mesh for TWSs. If the thickness of the structure is not much smaller than the size in surface direction, then the extrusion technique is not suitable. For the TWSs which are considered in this study, it is assumed that the extrusion will not cause any overlapping of elements for a single surface.

2) Fast mesh generation speed

The 3D elements are simply generated by extruding the surface element along the surface normal directions. The thicknesses of the structure are simply defined at every surface node and the element sizes in normal direction are easily controlled by the nodal number of element layers (NEL). The corresponding computational time on solid mesh generation is much fewer than that of the conventional 3D mesh generators.

3) No restriction on the surface element types

Both triangular elements and quadrilateral elements can be used in the input surface mesh. Furthermore, a mixed mesh (having both triangular elements and quadrilateral elements) can also be used. For a mixed input surface mesh, the solid element types in the corresponding solid mesh will consist of tetrahedron, pyramid, prism and hexahedron.

4) Surface normal sensitivity

The extrusion technique is very sensitive to the surface normal directions. For the structure with intersecting surfaces, some normal directions must be modified to avoid element overlapping. The modified normal directions will affect the quality of the volume mesh and the smoothing work is implemented by adjusting the modified

normal directions.

7.1.2 Conclusion of the error estimation step

In this procedure, the *a priori* error estimate and the *a posteriori* error estimate for TWSs have been made. The stresses are recovered based on the SPR technique and the FE error is estimated by Z-Z error estimator. The contribution of the part is the design of the modified superconvergent patch recovery technique. Different from the traditional SPR technique, the modified SPR technique will separate the stress in two parts: the surface part components and normal part components, and then recover these two parts of stress components individually. The characteristics of this new error estimator are summarized as following:

1) Specially designed for TWSs

The modified SPR technique is designed based on the separation of stress components according to surface direction and normal direction. The concepts of surface direction and normal direction are only applicable for TWSs, hence the modified SPR is suitable for the analysis of TWSs.

2) Focus on the surface direction

Because the stress components in surface direction are predominant in TWSs, in the stress recovery process, focus is given to the stress components in the surface direction only. A linear distribution of stress variation along the normal direction is assumed so that less polynomial terms are used for the stress components along the normal direction. By this treatment, the computational cost can be greatly saved while the overall stress recovery effect will not be affected much.

3) Structural junction is treated separately

For the TWS with junctions, because the stress variation near the junction is

complicated and no distinct surface direction or normal direction can be identified, the elements near the junction are considered as general 3D elements and no stress separation is applied. Hence, the traditional SPR technique is used for these elements.

4) Boundary patch enhancement technique is used

The traditional SPR technique is unable to achieve good recovery effect at the mesh boundaries and sometimes even fail due to the bad alignment of elements. In this study, boundary patch enhancement technique has been adopted to enhance the stability of numerical calculation and improve the recovered results for boundary patches.

7.1.3 Conclusion of the adaptive refinement scheme design step

The originality of this part is the development of a new adaptive refinement scheme for TWSs. Because the error of the FE solution is estimated in surface direction and normal direction separately, the mesh refinement process is also implemented in two separate directions: surface direction and normal direction. The adaptive refinement scheme developed in this study has the following characteristics:

1) Separation of the refinement process

Different from the traditional mesh refinement process, which is carried out in only one mesh generator procedure, in this thesis, the mesh is refined by two mesh generation procedures. Firstly, the surface direction element sizes are adjusted by surface mesh refinement which is carried out by an existing surface mesh generator. Secondly, the normal direction element sizes are adjusted by the volume mesh refinement which is carried out by the solid mesh converter designed in this study.

2) Controlled by different target errors

For TWSs, there are a group of target errors: target error of the whole mesh, target error of the junction zone, target error of the shell zone, target error of the surface direction of the shell zone and target error of the normal direction of the shell zone. The mesh refinement procedure is controlled by all these target errors. In case that not all the target errors are defined by users, the developed adaptive refinement scheme can determine all the unspecified target errors by itself.

3) Focus on the surface mesh refinement

Since the stress components in surface direction are predominant in TWSs, the refinement focus is also given to the surface direction. Therefore, compared to the target error for surface direction, a relatively large target error is assigned to the normal direction. This treatment can save the total DOFs and hence reduce the computation time for mesh generation and FE analysis.

4) Control of mesh refinement speed

In order to avoid over refinement in some local regions caused by inaccurate error estimation during the first several meshes, immediate target errors are designed to control the mesh refinement speed. Although this treatment will take more refinement steps to reach the final mesh, it can save the total DOFs spent in the final mesh. Normally, computational time spent on mesh generation is much less than the one spent on FE analysis, hence this kind of treatment is worthy and necessary.

7.2 Plans for Further Research Work

With regard to the further work of this study, several issues remained to be investigated in the future are as the following:

Conclusion and future work

1) Improvement for complicated surface intersections

The solid mesh generator developed in this study is mainly designed for the TWS formed with simple surface intersections. For the structures where complex junctions (Figure 7.1) are involved, the current algorithms are unable to handle them yet. Therefore, in the future work, more complicated cases of surface intersections should be considered and the corresponding algorithms should be developed.

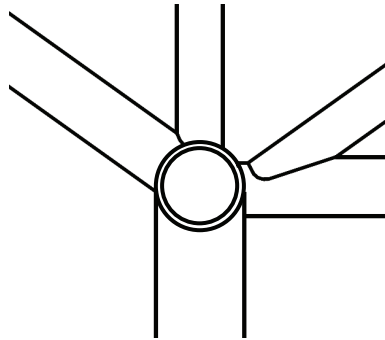


Figure 7.1 Example of TWS with complex surface intersection

2) Improvement of refinement in the thickness direction

In the volume mesh refinement part, it is required in this study that the difference of the element number of layers between two neighboring nodal points cannot be greater than 2. This constraint could be too strict for some cases when the difference of NELs between two nodal points is big, for example, in the region near the boundary between junction zone and shell zone. For two neighboring nodes whose NELs are denoted as NEL_1 and NEL_2 , when the difference of NEL_1 and NEL_2 is quite big such as

$$NEL_1 - NEL_2 > 6 \quad (7.1)$$

the NEL_2 has to be increased to $NEL_1 - 2$ by the current NEL redefinition algorithm. It should be noted that the redefined NEL_2 is much bigger than the original value so that the final element layers at this node will differ too much from the designed value. This problem will be more serious as the difference of NELs increases. An example is shown in Figure 7.2, the initial NEL values at two neighboring nodal

points A and B are 8 and 1, respectively. If the current NEL redefinition algorithm is used, then the NEL value of node B will be modified to 6 (Figure 7.2a). The new NEL at B will further affect the NELs of its neighboring nodes and so on. If the initial NEL at A is very big, then the NELs of a large number of nodes nearby have to be increased. A better way is to divide the normal sides at A and B into 3 parts first as shown in Figure 7.2b and each part is treated individually, then the NEL value of node B will only be increased to 3, which is only a half of the previous determined NEL value (6). Such kind of adjustment can effectively limit the propagation of a large NEL value.

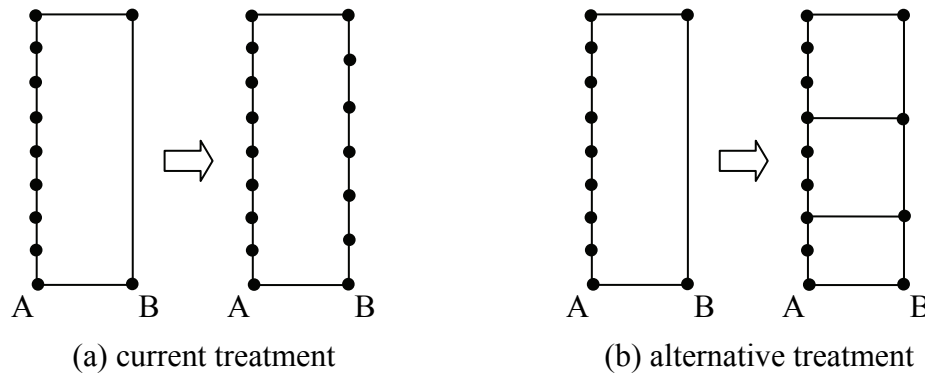


Figure 7.2 Adjust the algorithm on NEL redefinition

3) Use of advance solid elements

In the current study, standard second order solid elements are used for the FE analysis. Although some currently available non-conforming elements do not show much advantage than the standard elements in the analysis of TWSs, with the development of the high performance elements, more accurate solid element types will be available and they can be applied to the FE analysis of TWSs.

4) Selection of normal direction

In this study, the stress separation is carried out based on the surface direction and normal direction. Here the normal direction is not a strict geometric normal

Conclusion and future work

direction but only used for the assumption that the stress variation along this direction is linear. If the normal direction is not well defined, then the above assumption will be less accurate and the recovery results will be affected. Therefore, how to get a better definition of the normal direction is important and the current algorithm involved needs to be improved.

5) Improved schemes for the definition of immediate target errors

Immediate target errors are used in the current study to control the refinement speed and they are defined as the half of the current errors. Other than the current definition, some other methods to determine the immediate target errors can be considered. For example, an available different definition of the immediate target errors is to always use the half of the current total error as the immediate total target error, and all the other target errors are determined from the immediate total target error by the target error distribution algorithm described in Section 6.2.2. Different definitions of immediate target errors should be tested and compared and the one which can provide the best performance will be selected.

6) Local mesh regeneration for the junction zone

From all the numerical examples shown in Chapters 3 and 4, it can be seen that when the angle between two intersected surfaces is small (e.g. less than 45°), the solid elements near the junction are not well shaped, therefore, the FEA results of these elements are not as accurate as the elements far away from the junction. While for the structures with joints, the stresses around the joint are more complicated and important than the stresses at the position elsewhere. Hence, it will be very useful to enhance the FEA performance by improving the shape qualities of the elements near the surface junctions. To achieve this aim, a part of future work will be devoted to the design of a local mesh regeneration scheme for the junction zone. In order to ensure the compatibility of the nodal connection with the elements from the shell

Conclusion and future work

zone, the facet elements which are obtained by projecting the 3D elements onto the boundary of junction zone will be kept unchanged. The 3D elements within the junction zone are then removed and a new local mesh is generated using Delaunay Triangulation Algorithm (Section 2.2.2.3).

7) Improvement on the computer program

Currently, all the designed algorithms have been applied to a computer program which is written by the author. It must be mentioned that in this study, the programming work is only concentrated in the implementation of theory rather than to develop an optimized commercial software. The current computer program is not optimized yet in speed performance and further improvement in CPU time is still possible. More detailed checking work is also needed to enhance the stability of the program and to make it robust.

8) Analysis of TWSs made of inhomogeneous materials

In so far the structures studied in this thesis are all made of homogeneous and isotropic materials. In the future work, the structures made of anisotropic materials and even inhomogeneous materials should also be considered. A particular problem which is often met in application is how to deal with the welding area around the surface junctions. Since the welding material usually has the different properties from the material of the shell structures, special techniques need to be designed near the welding area.

9) Analysis of nonlinear problem for TWSs

In this study the adaptive refinement scheme is only designed for the problems limited within the linear elastic range. In the future work, efforts should be invested into the analysis of nonlinear problem, which is known as one of the most difficult areas in FEM. A robust adaptive refinement procedure would be very useful for the

Conclusion and future work

solution of this kind of problem since it can dramatically cut down the huge computational cost needed and permit the users to have a more detailed understanding of the complex solutions. There are two types of nonlinearity in nonlinear problem, namely, geometrical nonlinearity and material nonlinearity. For the plans of future study of TWSs, the problems involved in geometrical nonlinearity and material nonlinearity mainly refer to the large deformation problems and the elastoplastic problems.

REFERENCE

1. D. Bowness and M. M. K. Lee (1995), "The development of an accurate model for the fatigue assessment of doubly curved cracks in tubular joints", *International Journal of Fracture*, Vol. 73, No. 2, pp: 129-147.
2. M. M. K. Lee (1999), "Strength, stress and fracture analyses of offshore tubular joints using finite elements", *Journal of Constructional Steel Research*, Vol. 51, No. 3, pp: 265-286.
3. C. K. Lee (2000), "Automatic metric advancing front triangulation over curved surfaces", *Engineering Computations*, Vol. 17, No. 1, pp: 48-74.
4. Y. K. Lee and C. K. Lee (2003), "A new indirect anisotropic quadrilateral mesh generation scheme with enhanced local mesh smoothing procedures", *International Journal for Numerical Methods in Engineering*, Vol. 58, No. 2, pp: 277-300.
5. C. K. Lee, S. M. Wong and S. T. Lie (2001), "On increasing the order and density of 3d finite element meshes", *Communications in Numerical Methods in Engineering*, Vol. 17, No. 1, pp: 55-68.
6. K. Y. Sze and H. Fan (1996), "An economical assumed stress brick element and its implementation", *Finite Elements in Analysis and Design*, Vol. 21, No. 3, pp: 179-200.
7. K. Y. Sze and W. K. Chan (2001), "A six-node pentagonal assumed natural strain solid-shell element", *Finite Elements in Analysis and Design*, Vol. 37, No. 8, pp: 639-655.
8. S. H. Lo and C. Ling (2000), "Improvement on the 10-node tetrahedral element for three-dimensional problems", *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, No. 3, pp: 961-974.
9. O. C. Zienkiewicz and J. Z. Zhu (1992), "The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity",

International Journal for Numerical Methods in Engineering, Vol. 33, No. 7, pp: 1365-1382.

10. O. C. Zienkiewicz and J. Z. Zhu (1992), "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique", International Journal for Numerical Methods in Engineering, Vol. 33, No. 7, pp: 1331-1364.
11. C. K. Lee (1999), "Automatic adaptive mesh generation using metric advancing front approach", Engineering Computations, Vol. 16, No. 2-3, pp: 230-263.
12. R. Courant (1943), "Variational methods for the solution of problems of equilibrium and vibrations", Bulletin of the American Mathematical Society, Vol. 49, pp: 1-23.
13. M. J. Turner, R. W. Clough, H. C. Martin and L. J. Topp (1956), "Stiffness and deflection analysis of complex structures", Journal of the Aeronautical Sciences, Vol. 23, No. 9, pp: 805-823.
14. Clough (1960), "The finite element method in plane stress analysis", 2nd Conference on Electronic Computation Proceedings, ASCE, Pittsburgh, Pennsylvania.
15. O. C. Zienkiewicz and Y. K. Cheung (1965), "Finite elements in the solution of field problems", The Engineer, Vol. 220, pp: 507-510.
16. O. C. Zienkiewicz and D. V. Phillips (1971), "An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co-ordinates", International Journal for Numerical Methods in Engineering, Vol. 3, No. 4, pp: 519-528.
17. S. C. Wu and J. F. Abel (1979), "Representation and discretization of arbitrary surfaces for finite element shell analysis", International Journal for Numerical Methods in Engineering, Vol. 14, No. 6, pp: 813-836.
18. F. Ghassemi (1982), "Automatic mesh generation scheme for a two- or three-dimensional triangular curved surface", Computers and Structures, Vol. 15, No. 6, pp: 613-626.
19. T. S. Lau and S. H. Lo (1996), "Finite element mesh generation over analytical curved surfaces", Computers and Structures, Vol. 59, No. 2, pp: 301-309.

20. R. Lohner (1996), "Regriidding surface triangulations", Journal of Computational Physics, Vol. 126, No. 1, pp: 1-10.
21. R. T. Farouki (1997), "Optimal parameterizations", Computer-Aided Geometric Design, Vol. 14, No. 2, pp: 153-168.
22. D. O. Potyondy, P. A. Wawrzynek and A. R. Ingraffea (1995), "An algorithm to generate quadrilateral or triangular element surface meshes in arbitrary domains with applications to crack propagation", International Journal for Numerical Methods in Engineering, Vol. 38, No. 16, pp: 2677-2701.
23. L. Piegl and W. Tiller (1987), "Curve and surface constructions using rational b-splines", Computer Aided Design, Vol. 19, No. 9, pp: 485-498.
24. H. Borouchaki, P. L. George and B. Mohammadi (1997), "Delaunay mesh generation governed by metric specifications .2. Applications", Finite Elements in Analysis and Design, Vol. 25, No. 1-2, pp: 85-109.
25. H. Borouchaki, P. L. George, F. Hecht, P. Laug and E. Saltel (1997), "Delaunay mesh generation governed by metric specifications .1. Algorithms", Finite Elements in Analysis and Design, Vol. 25, No. 1-2, pp: 61-83.
26. R. J. Cass, S. E. Benzley, R. J. Meyers and T. D. Blacker (1996), "Generalized 3-d paving: An automated quadrilateral surface mesh generation algorithm", International Journal for Numerical Methods in Engineering, Vol. 39, No. 9, pp: 1475-1489.
27. B. Delaunay (1934), "Sur la sphere vide", Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk, Vol. 7, pp: 793-800.
28. C. L. Lawson (1977), "Software for c1 surface interpolation", Mathematical Software III, pp: 161-194.
29. D. F. Watson (1981), "Computing the n-dimensional delaunay tessellation with application to voronoi polytopes", Vol. 24, No. 2, pp: 167-172.
30. T. J. Baker (1989), "Automatic mesh generation for complex three-dimensional regions using a constrained delaunay triangulation", Engineering with Computers (New York), Vol. 5, No. 3-4, pp: 161-175.
31. N. P. Weatherill and O. Hassan (1994), "Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints", International Journal for Numerical Methods in Engineering, Vol. 37, No. 12,

pp: 2005-2039.

32. P. L. George, F. Hecht and E. Saltel (1991), "Automatic mesh generator with specified boundary", *Computer Methods in Applied Mechanics and Engineering*, Vol. 92, No. 3, pp: 269-288.
33. S. H. Lo (1985), "A new mesh generation scheme for arbitrary planar domains", *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 8, pp: 1403-1426.
34. R. Lohner and Paresh Parikh (1988), "Generation of three-dimensional unstructured grids by the advancing-front method", *International Journal for Numerical Methods in Fluids*, Vol. 8, No. 10, pp: 1135-1149.
35. P. J. Frey, H. Borouchaki and P. L. George (1996), "Delaunay tetrahedralization using an advancing-front approach", *5th International Meshing Roundtable Proceeding*, Vol. 31-46,
36. P. Fleischmann and S. Selberherr (1997), "Fully unstructured delaunay mesh generation using a modified advancing front approach for applications in technology cad", *IEEE Transactions on Semiconductor Technology Modeling and Simulation*.
37. P. J. Frey, H. Borouchaki and P.-L. George (1998), "3d delaunay mesh generation coupled with an advancing-front approach", *Computer Methods in Applied Mechanics and Engineering*, Vol. 157, No. 1-2, pp: 115-131.
38. M. S. Renato Perucchio, Ajay Kela, (1989), "Automatic mesh generation from solid models based on recursive spatial decompositions", *International Journal for Numerical Methods in Engineering*, Vol. 28, No. 11, pp: 2469-2501.
39. S. T. Tan, M. M. F. Yuen, K. Y. Hung, (1991), "A hierarchical approach to automatic finite element mesh generation", *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 3, pp: 501-525.
40. M. S. Shephard and M. K. Georges (1991), "Automatic three-dimensional mesh generation by the finite octree technique", *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 4, pp: 709-749.
41. A. G. L. Borthwick and D. M. Greaves (1999), "Hierarchical tree-based finite element mesh generation", *International Journal for Numerical Methods in Engineering*, Vol. 45, No. 4, pp: 447-471.

42. D. A. Field (1988), "Laplacian smoothing and delaunay triangulations", Communications in Applied Numerical Methods, Vol. 4, No. 6, pp: 709-712.
43. J. U. Brackbill and J. S. Saltzman (1982), "Adaptive zoning for singular problems in two dimensions", Journal of Computational Physics, Vol. 46, pp: 342-368.
44. S. A. Canann, J. R. Tristano and M. L. Staten (1998), "An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes", 7th International Meshing Roundtable Proceedings, pp: 479-494.
45. S. Hyun and L.-E. Lindgren (2001), "Smoothing and adaptive remeshing schemes for graded element", Communications in Numerical Methods in Engineering, Vol. 17, No. 1, pp: 1-17.
46. L. Freitag, M. Jones and P. Plassmann (1995), "An efficient parallel algorithm for mesh smoothing", 4th International Meshing Roundtable Proceedings, pp: 47-58.
47. S. A. Canann, M. B. Stephenson and T. Blacker (1993), "Optismoothing: An optimization-driven approach to mesh smoothing", Finite Elements in Analysis and Design, Vol. 13, No. 2-3, pp: 185-190.
48. J. B. Frank and P. S. Heckbert (1996), "A pliant method for anisotropic mesh generation", 5th International Meshing Roundtable Proceeding, pp: 63-76.
49. Kenji Shimada, Atsushi Yamada and T. Itoh (1997), "Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles", 6th Internal Meshing Roundtable Proceedings, pp: 375-390.
50. P. Kinney (1997), "Cleanup: Improving quadrilateral finite element meshes", 6th International Meshing Roundtable Proceedings, pp: 437-447.
51. M. L. Staten and S. A. Canann (1997), "Post refinement element shape improvement for quadrilateral meshes", American Society of Mechanical Engineers, Applied Mechanics Division, AMD, Vol. 220, pp: 9-16.
52. B. Joe (1995), "Construction of three-dimensional improved-quality triangulations using local transformations", SIAM Journal on Scientific Computing, Vol. 16, No. 6, pp: 1292-1307.
53. S. A. Mitchell and T. J. Tautges (1995), "Pillowing doublets: Refining a mesh

to enhance that faces share at most one edge", 4th International Meshing Roundtable Proceedings, pp: 231-240.

54. M. C. Rivara (1984), "Algorithms for refining triangular grids suitable for adaptive and multigrid techniques", International Journal for Numerical Methods in Engineering, Vol. 20, No. 4, pp: 745-756.
55. D. J. Estep, S. M. Verduyn Lunel and R. D. Williams (1995), "Error estimation for numerical differential equations", IEEE Antennas and Propagation Magazine, Vol. 38, No. 2, pp: 71-76.
56. S. Prudhomme, P. F. J. T. J.T. Oden and A. Romkes (2000), "Review of a priori error estimation for discontinuous galerkin methods", TICAM REPORT, Texas Institute for Computational and Applied Mathematics, The University of Texas at Austin, pp: 00-27.
57. S. R. Wu (2003), "A priori error estimates for explicit finite element for linear elasto-dynamics by galerkin method and central difference method", Computer Methods in Applied Mechanics and Engineering, Vol. 192, No. 51-52, pp: 5329-5353.
58. M. G. Larson (2000), "A posteriori and a priori error analysis for finite element approximations of self-adjoint elliptic eigenvalue problems", SIAM Journal on Numerical Analysis, Vol. 38, No. 2, pp: 608-625.
59. J. R. Cho and J. T. Oden (1996), "Priori error estimations of hp-finite element approximations for hierarchical models of plate- and shell-like structures", Computer Methods in Applied Mechanics and Engineering, Vol. 132, No. 1-2, pp: 135-177.
60. J. R. Stewart and T. J. R. Hughes (1998), "A tutorial in elementary finite element error analysis: A systematic presentation of a priori and a posteriori error estimates", Computer Methods in Applied Mechanics and Engineering, Vol. 158, No. 1-2, pp: 1-22.
61. I. Babuska and W. C. Rheinboldt (1978), "Error estimates for adaptive finite element computations", SIAM Journal on Numerical Analysis, Vol. 15, No. 4, pp: 736-754.
62. M. Ainsworth and J. T. Oden (1993), "A unified approach to a posteriori error estimation using element residual methods", Numerische Mathematik, Vol. 65, No. 1, pp: 23-50.

63. M. Ainsworth and J. T. Oden (1993), "A posteriori error estimators for second order elliptic systems. 1. Theoretical foundations and a posteriori error analysis", *Computers & Mathematics with Applications*, Vol. 25, No. 2, pp: 101-113.
64. M. Ainsworth and J. T. Oden (1993), "A posteriori error estimators for second order elliptic systems. 2. An optimal order process for calculating self-equilibrating fluxes", *Computers & Mathematics with Applications*, Vol. 26, No. 9, pp: 75-87.
65. M. Ainsworth and J. Tinsley Oden (1997), "A posteriori error estimation in finite element analysis", *Computer Methods in Applied Mechanics and Engineering*, Vol. 142, No. 1-2, pp: 1-88.
66. K. Eriksson and C. Johnson (1987), "Error estimates and automatic time step control for non-linear parabolic problems", *SIAM Journal on Numerical Analysis*, Vol. 24, pp: 19-23.
67. K. Eriksson and C. Johnson (1988), "An adaptive finite element method for linear elliptic problems", *Mathematics of Computation*, Vol. 50, No. 182, pp: 361-383.
68. C. Johnson and P. Hansbo (1992), "Adaptive finite element methods in computational mechanics", *Computer Methods in Applied Mechanics and Engineering*, Vol. 101, No. 1-3, pp: 143-181.
69. R. Mucke and J. R. Whiteman (1995), "A posteriori error estimates and adaptivity for finite element solutions in finite elasticity", *International Journal for Numerical Methods in Engineering*, Vol. 38, No. 5, pp: 775-795.
70. I. Babuska and A. Miller (1984), "Post-processing approach in the finite element method - part 1: Calculation of displacements, stresses and other higher derivatives of the displacements", Vol. 20, No. 6, pp: 1085-1109.
71. I. Babuska and A. Miller (1984), "Post-processing approach in the finite element method - part 2: The calculation of stress intensity factors", Vol. 20, No. 6, pp: 1111-1129.
72. I. Babuska and A. Miller (1984), "Post-processing approach in the finite element method - part 3: A posteriori error estimates and adaptive mesh selection", *International Journal for Numerical Methods in Engineering*, Vol. 20, No. 12, pp: 2311-2324.

73. O. C. Zienkiewicz and J. Z. Zhu (1987), "Simple error estimator and adaptive procedure for practical engineering analysis", *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 2, pp: 337-357.
74. J. Z. Zhu and O. C. Zienkiewicz (1990), "Superconvergence recovery technique and a posteriori error estimators", *International Journal for Numerical Methods in Engineering*, Vol. 30, No. 7, pp: 1321-1339.
75. B. Boroomand and O. C. Zienkiewicz (1997), "Recovery by equilibrium in patches (rep)", *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 1, pp: 137-164.
76. T. Lee, H. C. Park and S. W. Lee (1997), "A superconvergent stress recovery technique with equilibrium constraint", *International Journal for Numerical Methods in Engineering*, Vol. 40, No. 6, pp: 1139-1160.
77. J. Z. Zhu and Z. Zhang (1999), "The relationship of some a posteriori estimators", *Computer Methods in Applied Mechanics and Engineering*, Vol. 176, No. 1-4, pp: 463-475.
78. C.-K. Choi and Y.-M. Park (1992), "Adaptive h-refinement using transition element for plate bending problems", *International Journal for Numerical Methods in Engineering*, Vol. 35, No. 1, pp: 145-163.
79. J. K. Wilson and B. H. V. Topping (1998), "A new element bisection algorithm for unstructured adaptive tetrahedral mesh generation", *Engineering Computations*, Vol. 15, No. 5, pp: 588-615.
80. E. Hinton, M. Oezakca and N. V. R. Rao (1991), "Adaptive analysis of thin shells using facet elements", *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 6, pp: 1283-1301.
81. P. Fernandes, P. Girdinio and G. Molinari (1994), "Techniques of h-refinement in adaptive meshing algorithms", *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 13, pp: 329-334.
82. I. Babuska, B. A. Szabo and I. N. Katz (1981), "The p-version of the finite element method", *SIAM Journal on Numerical Analysis*, Vol. 18, No. 3, pp: 515-545.
83. B. A. Szabo (1985), "Mesh design for the p-version of the finite element method", *Computer Methods in Applied Mechanics and Engineering*, Vol. 55,

No. 1-2, pp: 181-197.

84. W. Gui and I. Babuska (1986), "The h, p and h-p version of the finite method in i dimension. Part 1: The error analysis of the p-version", *Numerische Mathematik*, Vol. 48, pp: 557-612.
85. M. Papadrakakis, G. Babilis and P. Braouzi (1997), "Efficiency of refinement procedures for the p-version of the adaptive finite element method", *Engineering Computations*, Vol. 14, No. 1, pp: 98-118.
86. A. Duter, H. Broker and E. Rank (2001), "The p-version of the finite element method for three-dimensional curved thin walled structures", *International Journal for Numerical Methods in Engineering*, Vol. 52, No. 7, pp: 673-703.
87. I. Babuska and M. R. Dorr (1981), "Error estimates for the combined h and p versions of the finite element method", *Numerische Mathematik*, Vol. 37, No. 2, pp: 257-257.
88. I. Babuska and B. Guo (1986), "The h-p version of the finite element method. Part 2: General results and applications", *Computational Mechanics*, Vol. 1, No. 3, pp: 203.
89. I. Babuska and B. Guo (1986), "The h-p version of the finite element method. Part 1: The basic approximation results", *Computational Mechanics*, Vol. 1, No. 1, pp: 21-41.
90. W. Gui and I. Babuska (1986), "The h, p and h-p version of the finite method in i dimension. Part 3: The adaptive h-p version", *Numerische Mathematik*, Vol. 48, pp: 658-683.
91. O. C. Zienkiewicz, J. Z. Zhu and N. G. Gong (1989), "Effective and practical h-p-version adaptive analysis procedures for the finite element method", *International Journal for Numerical Methods in Engineering*, Vol. 28, No. 4, pp: 879-891.
92. P. K. Jimack (1996), "Optimal eigenvalue and asymptotic large-time approximations using the moving finite-element method", *IMA Journal of Numerical Analysis*, Vol. 16, No. 3, pp: 381-398.
93. P. K. Jimack (1996), "A best approximation property of the moving finite element method", *SIAM Journal on Numerical Analysis*, Vol. 33, No. 6, pp: 2286-2302.

94. P. K. Jimack and A. J. Wathen (1991), "Temporal derivatives in the finite-element method on continuously deforming grids", *SIAM Journal on Numerical Analysis*, Vol. 28, No. 4, pp: 990-1003.
95. A. K. Noor and I. Babuska (1987), "Quality assessment and control of finite element solutions", *Finite Elements in Analysis and Design*, Vol. 3, No. 1, pp: 1-26.
96. B. P. Johnston and J. M. J. Sullivan (1992), "Fully automatic two dimensional mesh generation using normal offsetting", *International Journal for Numerical Methods in Engineering*, Vol. 33, No. 2, pp: 425-442.
97. B. P. Johnston and J. M. J. Sullivan (1993), "Normal offsetting technique for automatic mesh generation in three dimensions", *International Journal for Numerical Methods in Engineering*, Vol. 36, No. 10, pp: 1717-1734.
98. D. Sharov, H. Luo, J. D. Baum and R. Lohner (2003), "Unstructured navier-stokes grid generation at corners and ridges", *International Journal for Numerical Methods in Fluids*, Vol. 43, No. 6-7, pp: 717-728.
99. D. L. Marcum (2001), "Efficient generation of high-quality unstructured surface and volume grids", *Engineering with Computers*, Vol. 17, No. 3, pp: 211-233.
100. A. N. Athanasiadis and H. Deconinck (2004), "Object-oriented three-dimensional hybrid grid generation", *International Journal for Numerical Methods in Engineering*, Vol. 58, No. 2, pp: 301-318.
101. W. R. Quadros and K. Shimada (2002), "Hex-layer: Layered all-hex mesh generation on thin section solid via chordal surface transformation", *Proceedings of the 11th International Meshing Roundtable*, pp: 169-180.
102. G. Bedrosian (1992), "Shape functions and integration formulas for three-dimensional finite element analysis", *International Journal for Numerical Methods in Engineering*, Vol. 35, No. 1, pp: 95-108.
103. R. P. Bhatia and K. L. Lawrence (1990), "Two-dimensional finite element mesh generation based on stripwise automatic triangulation", *Computers and Structures*, Vol. 36, No. 2, pp: 309-319.
104. S. H. Lo (1989), "Generating quadrilateral elements on plane and over curved surfaces", *Computers and Structures*, Vol. 31, No. 3, pp: 421-426.

105. P. J. Frey and H. Borouchaki (1999), "Surface mesh quality evaluation", International Journal for Numerical Methods in Engineering, Vol. 45, No. 1, pp: 101-118.
106. J. Robinson (1994), "Quadrilateral and hexahedron shape parameters", Finite Elements in Analysis and Design, Vol. 16, No. 1, pp: 43-52.
107. D. A. Field (2000), "Qualitative measures for initial meshes", International Journal for Numerical Methods in Engineering, Vol. 47, No. 4, pp: 887-906.
108. S. H. Lo and W. J. Chen (2004), "High-performance 3d solid pyramid and hexahedral elements", Private communication.
109. S. H. Lo (2004), "High-performance 3d solid 13-node pyramid elements", Proceedings of the Sixth World Congress in Computational Mechanics, Beijing, China, September.,
110. R. L. Taylor, P. J. Beresford and E. L. Wilson (1976), "A non-conforming element for stress analysis", International Journal for Numerical Methods in Engineering, Vol. 10, No. 6, pp: 1211-1219.
111. C. K. Lee and R. E. Hobbs (1997), "Automatic adaptive refinement for shell analysis using nine-node assumed strain element", International Journal for Numerical Methods in Engineering, Vol. 40, No. 19, pp: 3601-3638.
112. O. C. Zienkiewicz and J. Z. Zhu (1992), "The superconvergent patch recovery (spr) and adaptive finite element refinement", Computer Methods in Applied Mechanics and Engineering, Vol. 101, No. 1-3, pp: 207-224.
113. O. C. Zienkiewicz and J. Z. Zhu (1995), "Superconvergence and the superconvergent patch recovery", Finite Elements in Analysis and Design, Vol. 19, No. 1-2, pp: 11-23.
114. O. C. Zienkiewicz, J. Z. Zhu and J. Wu (1993), "Superconvergent patch recovery techniques - some further tests", Communications in Numerical Methods in Engineering, Vol. 9, No. 3, pp: 251-258.
115. A. G. P. Labbé(1995), "A robust implementation of zienkiewicz and zhu's local patch recovery method", Communications in Numerical Methods in Engineering, Vol. 11, No. 5, pp: 427-434.
116. N. Wiberg and F. Abdulwahab (1992), "An efficient postprocessing technique for stress problems based on superconvergent derivatives and equilibrium",

Numerical Methods in Engineering '92, Elsevier Science Publishers, Vol. 8, pp: 25-32.

117. T. Belytschko and T. D. Blacker (1994), "Enhanced derivative recovery through least square residual penalty", *Applied Numerical Mathematics*, Vol. 14, No. 1-3, pp: 55-68.
118. S. H. Lo and C. K. Lee (1998), "On using different recovery procedures for the construction of smoothed stress in finite element method", *International Journal for Numerical Methods in Engineering*, Vol. 43, No. 7, pp: 1223-1252.
119. Juan J. Ródenas, F. J. Funemayor, A. Roda and A. Pedrosa (2003), "Modification of the spr technique to ensure the fulfillment of imposed tractions in boundary nodes", *International Conference on Adaptive Modeling and Simulation*.
120. I. Szabo and I. Babuska (1991), "Finite Element Analysis", Chap 7, Wiley, New York.
121. T. Blacker and T. Belytschko (1994), "Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements", *International Journal for Numerical Methods in Engineering*, Vol. 37, No. 3, pp: 517-536.
122. S. P. Timoshenko and J. N. Goodier (1970), "Theory of elasticity", Chap. 4, McGraw-Hill, New York, 3rd edition.
123. O. C. Zienkiewicz and J. Z. Zhu (1989), "Error estimates and adaptive refinement for plate bending problems", *International Journal for Numerical Methods in Engineering*, Vol. 28, No. 12, pp: 2839-2853.
124. R. Verfurth (1994), "Posteriori error estimation and adaptive mesh-refinement techniques", *Journal of Computational and Applied Mathematics*, Vol. 50, No. 1-3, pp: 67-83.
125. C. K. Lee and C. E. Zhou (2004), "On error estimation and adaptive refinement for element free galerkin method: Part ii: Adaptive refinement", *Computers and Structures*, Vol. 82, No. 4-5, pp: 429-443.
126. C. K. Lee and R. E. Hobbs (1997), "On using different finite elements with an automatic adaptive refinement procedure for the solution of 2-d stress analysis problems", *International Journal for Numerical Methods in Engineering*, Vol. 40, pp: 4547-4576.

127. I. Babuska and L. Li (1992), "The problem of plate modeling: Theoretical and computational results", *Computer Methods in Applied Mechanics and Engineering*, Vol. 100, No. 2, pp: 249-273.
128. C. K. Lee and R. E. Hobbs (1998), "Automatic adaptive refinement for plate bending problems using reissner-mindlin plate bending elements", *International Journal for Numerical Methods in Engineering*, Vol. 41, No. 1, pp: 1-63.

APPENDIX A

FORMATION OF THE HIGH-PERFORMANCE ELEMENT

In Chapter 5, it was mentioned that some high-performance elements haven been tested in this study. In this appendix, detailed descriptions will be given for the formation of such kind of high-performance elements.

A.1 Weaker Continuity Condition

It is well known that convergence FE results can be ensured in conforming elements; however the rate of convergence is sometimes rather slow. Moreover, locking will often occur when conforming elements are employed in the analysis of some special problems, such as, slender and thin structures or nearly incompressible materials. Conforming elements can be improved by using reduced integration, by adding interior nonconforming function, and by using the enhanced strain method etc. The refined nonconforming element method can be derived from the weaker continuity condition between elements, which is given by,

$$\int_{V_e} u_{i,j}^* dv = \int_{\partial V_e} \tilde{u}_i n_j ds \quad \text{A.1)}$$

where $u_i^*(i, j=1,2,3)$ are the displacement functions of the element V_e , $\tilde{u}_i(i=1,2,3)$ are the displacement functions on the boundary between two elements, and n_j are the direction cosines normal to the boundary of the element ∂V_e .

According to Green-Gauss theorem, it can be obtained that

Formation of the high-performance element

$$\int_{V_e} u_{i,j}^* dv = \int_{\partial V_e} u_i^* n_j ds \quad (A.2)$$

So

$$\int_{\partial V_e} (u_i^* - \tilde{u}_i) n_j ds = 0 \quad (A.3)$$

The above equation implies that the inter-element continuity condition of the displacement u_i^* , which is related to $\tilde{u}_i$, can be satisfied in an average sense.

Therefore, Equation A.1) is the so-called weaker continuity condition. It should be noted that this method can be used to improve both nonconforming and conforming elements.

A.2 Refined Nonconforming Displacement function

In general the nonconforming displacement $\mathbf{u}$ can be separated into two parts

$$\mathbf{u} = \mathbf{u}_b + \mathbf{u}_\lambda \quad (A.4)$$

where $\mathbf{u}_b$ is conforming displacement and $\mathbf{u}_\lambda$ is the nonconforming displacement.

The above formula can be written in the form of vectors

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u_b \\ v_b \\ w_b \end{Bmatrix} + \begin{Bmatrix} u_\lambda \\ v_\lambda \\ w_\lambda \end{Bmatrix} \quad (A.5)$$

Based on refined nonconforming element method, new element function can be enhanced as follows,

$$\begin{Bmatrix} u^* \\ v^* \\ w^* \end{Bmatrix} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} + \mathbf{P}\boldsymbol{\alpha} \quad (A.6)$$

where

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} 2x & 0 & 0 & y & 0 & z \\ 0 & 2y & 0 & x & z & 0 \\ 0 & 0 & 2z & 0 & y & x \end{bmatrix} \text{ and } \boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \quad \alpha_5 \quad \alpha_6] \quad (A.7)$$

In Equation (A.6), $\alpha_i (i=1,2,...6)$ are undetermined parameters. Substituting

Formation of the high-performance element

Equations (A.5) and (A.7) into Equation (A.6), it can be obtained that

$$\begin{Bmatrix} u^* \\ v^* \\ w^* \end{Bmatrix} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} + \mathbf{P}\mathbf{a} = \begin{Bmatrix} u_b \\ v_b \\ w_b \end{Bmatrix} + \begin{Bmatrix} u_\lambda \\ v_\lambda \\ w_\lambda \end{Bmatrix} + \begin{Bmatrix} \alpha_1 x + \frac{1}{2}\alpha_4 y + \frac{1}{2}\alpha_6 z \\ \alpha_2 y + \frac{1}{2}\alpha_4 x + \frac{1}{2}\alpha_5 z \\ \alpha_3 z + \frac{1}{2}\alpha_5 y + \frac{1}{2}\alpha_6 x \end{Bmatrix} \quad (\text{A.8})$$

The element functions shown in Equation (A.8) are nonconforming for arbitrary $\alpha_i (i=1,2,\dots,6)$. In order to ensure convergence, the new element functions should satisfy the weaker continuity condition, which can be written as follows (using displacement u^* for example)

$$\begin{aligned} \int_{V_e} \frac{\partial u^*}{\partial x} dv &= \int_{\partial V_e} \tilde{u} l ds \\ \int_{V_e} \frac{\partial u^*}{\partial y} dv &= \int_{\partial V_e} \tilde{u} m ds \\ \int_{V_e} \frac{\partial u^*}{\partial z} dv &= \int_{\partial V_e} \tilde{u} n ds \end{aligned} \quad (\text{A.9})$$

where l, m, n are the direction cosines normal to element boundary. For conforming elements, boundary displacements $\tilde{u}, \tilde{v}, \tilde{w}$ of the element are equal to the values of conforming element displacements on the boundary

$$\begin{aligned} \int_{\partial V_e} \tilde{u} l ds &= \int_{V_e} \frac{\partial u_b}{\partial x} dv \\ \int_{\partial V_e} \tilde{u} m ds &= \int_{V_e} \frac{\partial u_b}{\partial y} dv \\ \int_{\partial V_e} \tilde{u} n ds &= \int_{V_e} \frac{\partial u_b}{\partial z} dv \end{aligned} \quad (\text{A.10})$$

Substituting Equation (A.9) into Equation (A.10), it can be obtained

Formation of the high-performance element

$$\begin{aligned}
 \int_{V_e} \frac{\partial u^*}{\partial x} dv &= \int_{V_e} \frac{\partial u_b}{\partial x} dv \\
 \int_{V_e} \frac{\partial u^*}{\partial y} dv &= \int_{V_e} \frac{\partial u_b}{\partial y} dv \\
 \int_{V_e} \frac{\partial u^*}{\partial z} dv &= \int_{V_e} \frac{\partial u_b}{\partial z} dv
 \end{aligned} \tag{A.11}$$

Substituting Equation (A.8) into Equation (A.11), it can be obtained

$$\begin{aligned}
 \alpha_1 V &= - \int_{V_e} \frac{\partial u_\lambda}{\partial x} dv \\
 \frac{\alpha_4}{2} V &= - \int_{V_e} \frac{\partial u_\lambda}{\partial y} dv \\
 \frac{\alpha_6}{2} V &= - \int_{V_e} \frac{\partial u_\lambda}{\partial z} dv
 \end{aligned} \tag{A.12}$$

where $V = \int_{V_e} dv$. As for the displacements v^*, w^* , similar relations can be obtained

$$\begin{aligned}
 \alpha_2 V &= - \int_{V_e} \frac{\partial v_\lambda}{\partial y} dv & \alpha_3 V &= - \int_{V_e} \frac{\partial w_\lambda}{\partial z} dv \\
 \frac{\alpha_4}{2} V &= - \int_{V_e} \frac{\partial v_\lambda}{\partial x} dv & \text{and } \frac{\alpha_5}{2} V &= - \int_{V_e} \frac{\partial w_\lambda}{\partial y} dv \\
 \frac{\alpha_5}{2} V &= - \int_{V_e} \frac{\partial v_\lambda}{\partial z} dv & \frac{\alpha_6}{2} V &= - \int_{V_e} \frac{\partial w_\lambda}{\partial x} dv
 \end{aligned} \tag{A.13}$$

From Equations (A.12) and (A.13), it can be obtained

$$\left\{ \begin{aligned} \alpha_1 V &= - \int_{V_e} \frac{\partial u_\lambda}{\partial x} dv \\ \alpha_2 V &= - \int_{V_e} \frac{\partial v_\lambda}{\partial y} dv \\ \alpha_3 V &= - \int_{V_e} \frac{\partial w_\lambda}{\partial z} dv \end{aligned} \right. \text{ and } \left\{ \begin{aligned} \alpha_4 V &= - \int_{V_e} \left(\frac{\partial u_\lambda}{\partial y} + \frac{\partial v_\lambda}{\partial x} \right) dv \\ \alpha_5 V &= - \int_{V_e} \left(\frac{\partial v_\lambda}{\partial z} + \frac{\partial w_\lambda}{\partial y} \right) dv \\ \alpha_6 V &= - \int_{V_e} \left(\frac{\partial u_\lambda}{\partial z} + \frac{\partial w_\lambda}{\partial x} \right) dv \end{aligned} \right. \tag{A.14}$$

The above relations can be written as

Formation of the high-performance element

$$\mathbf{a} = -\frac{1}{V} \int_{V_e} \mathbf{L} \begin{Bmatrix} u_\lambda \\ v_\lambda \\ w_\lambda \end{Bmatrix} dv = -\mathbf{B}_{\lambda 0} \boldsymbol{\lambda} \quad (\text{A.15})$$

where

$$\mathbf{B}_{\lambda 0} \boldsymbol{\lambda} = \frac{1}{V} \int_{V_e} \mathbf{L}^T \begin{Bmatrix} u_\lambda \\ v_\lambda \\ w_\lambda \end{Bmatrix} dv, \quad \mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^T \quad (\text{A.16})$$

and $\boldsymbol{\lambda}$ is the vector for all the nonconforming displacements.

Substituting Equation (A.6) into Equation (A.15), it can be obtained

$$\begin{Bmatrix} u^* \\ v^* \\ w^* \end{Bmatrix} = \begin{Bmatrix} u_b \\ v_b \\ w_b \end{Bmatrix} + \begin{Bmatrix} u_\lambda \\ v_\lambda \\ w_\lambda \end{Bmatrix} - \mathbf{P} \mathbf{B}_{\lambda 0} \boldsymbol{\lambda} \quad (\text{A.17})$$

According to Equation (A.6), the vector for displacements can be written as

$$\mathbf{q}^* = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} \quad (\text{A.18})$$

where $\mathbf{q}$ is the vector correspond to all the conforming displacements of the element nodes and $\boldsymbol{\lambda}$ is the vector correspond to the appendix nonconforming displacements.

From Equation (A.17), the following deduction can be made

$$\begin{aligned} \begin{Bmatrix} u^* \\ v^* \\ w^* \end{Bmatrix} &= \begin{Bmatrix} u_b + u_\lambda \\ v_b + v_\lambda \\ w_b + w_\lambda \end{Bmatrix} - \mathbf{P} [0 \mid \mathbf{B}_{\lambda 0}] \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} \\ \Rightarrow \mathbf{L} \begin{Bmatrix} u^* \\ v^* \\ w^* \end{Bmatrix} &= \mathbf{L} \begin{Bmatrix} u_b + u_\lambda \\ v_b + v_\lambda \\ w_b + w_\lambda \end{Bmatrix} - \mathbf{L} \mathbf{P} [0 \mid \mathbf{B}_{\lambda 0}] \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} \end{aligned}$$

Formation of the high-performance element

$$\begin{aligned} \Rightarrow \mathbf{B}^* \begin{bmatrix} \mathbf{q} \\ \lambda \end{bmatrix} &= [\mathbf{B}_b \mid \mathbf{B}_\lambda] \begin{bmatrix} \mathbf{q} \\ \lambda \end{bmatrix} - \mathbf{E}[0 \mid \mathbf{B}_{\lambda 0}] \begin{bmatrix} \mathbf{q} \\ \lambda \end{bmatrix} \\ \Rightarrow \mathbf{B}^* \mathbf{q}^* &= [\mathbf{B}_b \mid \mathbf{B}_\lambda] \mathbf{q}^* - [0 \mid \mathbf{B}_{\lambda 0}] \mathbf{q}^* \\ \Rightarrow \mathbf{B}^* &= [\mathbf{B}_b \mid \mathbf{B}_\lambda] - [0 \mid \mathbf{B}_{\lambda 0}] = [\mathbf{B}_b \mid (\mathbf{B}_\lambda - \mathbf{B}_{\lambda 0})] \end{aligned}$$

Let $\mathbf{B}_t = \mathbf{B}_\lambda - \mathbf{B}_{\lambda 0}$, then $\mathbf{B}^* = [\mathbf{B}_b \mid \mathbf{B}_t]$, so the stiffness matrix is

$$\mathbf{K}^* = \int \mathbf{B}^{*T} \mathbf{D} \mathbf{B}^* dV = \int [\mathbf{B}_b \mid \mathbf{B}_t]^T \mathbf{D} [\mathbf{B}_b \mid \mathbf{B}_t] dV = \int \begin{bmatrix} \mathbf{B}_b^T \mathbf{D} \mathbf{B}_b & \mathbf{B}_b^T \mathbf{D} \mathbf{B}_t \\ \mathbf{B}_t^T \mathbf{D} \mathbf{B}_b & \mathbf{B}_t^T \mathbf{D} \mathbf{B}_t \end{bmatrix} dV \quad (\text{A.19})$$

Let the equations $\mathbf{K}^* \mathbf{q}^* = \mathbf{r}$ represent a portion of the entire structure. Since $\mathbf{q}^*$ has been partitioned such that $\mathbf{q}^* = [\mathbf{q} \ \lambda]^T$ and $\mathbf{r}$ also can be partitioned as $\mathbf{r} = [\mathbf{r}_q \ 0]^T$, thus, $\mathbf{K}^* \mathbf{q}^* = \mathbf{r}$ becomes

$$\begin{bmatrix} \mathbf{K}_{qq} & \mathbf{K}_{q\lambda} \\ \mathbf{K}_{\lambda q} & \mathbf{K}_{\lambda\lambda} \end{bmatrix} \begin{Bmatrix} \mathbf{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_q \\ 0 \end{Bmatrix} \quad (\text{A.20})$$

where

$$\begin{cases} \mathbf{K}_{qq} = \int \mathbf{B}_b^T \mathbf{D} \mathbf{B}_b dV \\ \mathbf{K}_{q\lambda} = \int \mathbf{B}_b^T \mathbf{D} \mathbf{B}_t dV \\ \mathbf{K}_{\lambda q} = \int \mathbf{B}_t^T \mathbf{D} \mathbf{B}_b dV \\ \mathbf{K}_{\lambda\lambda} = \int \mathbf{B}_t^T \mathbf{D} \mathbf{B}_t dV \end{cases} \quad (\text{A.21})$$

The final stiffness matrix which will be used for the element can be obtained by condensation. The lower partition is solved for λ :

$$\lambda = -\mathbf{K}_{\lambda\lambda}^{-1} \mathbf{K}_{\lambda q} \mathbf{q} \quad (\text{A.22})$$

Then λ is back substituted into the upper partition of equation

$$(\mathbf{K}_{qq} - \mathbf{K}_{q\lambda} \mathbf{K}_{\lambda\lambda}^{-1} \mathbf{K}_{\lambda q}) \mathbf{q} = \mathbf{r}_q \quad (\text{A.23})$$

So the condensed final stiffness matrix is

$$\mathbf{K} = \mathbf{K}_{qq} - \mathbf{K}_{q\lambda} \mathbf{K}_{\lambda\lambda}^{-1} \mathbf{K}_{\lambda q} \quad (\text{A.24})$$

Formation of the high-performance element

The element can now be treated in standard fashion. That is, the condensed $\mathbf{K}$ and $\mathbf{r}_q$ will be assembled into the structure. Boundary conditions are imposed and the nodal displacements are resolved. After that, the strains can be calculated as

$$\boldsymbol{\varepsilon} = \mathbf{B}^* \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} = [\mathbf{B}_b \mid \mathbf{B}_\lambda] \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} - [0 \mid \mathbf{B}_{\lambda 0}] \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} = \mathbf{B}_b \mathbf{q} + \mathbf{B}_\lambda \boldsymbol{\lambda} - \mathbf{B}_{\lambda 0} \boldsymbol{\lambda} \quad (\text{A.25})$$

The nonconforming displacements $\mathbf{u}_\lambda$ can be determined by

$$\begin{Bmatrix} u_\lambda \\ v_\lambda \\ w_\lambda \end{Bmatrix} = \mathbf{M} \boldsymbol{\lambda} \quad (\text{A.26})$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & 0 & 0 \\ 0 & \mathbf{M}_1 & 0 \\ 0 & 0 & \mathbf{M}_1 \end{bmatrix} \text{ and } \boldsymbol{\lambda} = [\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4 \quad \lambda_5 \quad \lambda_6 \quad \lambda_7 \quad \lambda_8 \quad \lambda_9] \quad (\text{A.27})$$

Different functions of $\mathbf{M}_1$ should be selected for different element types. In this study, the selection of functions of $\mathbf{M}_1$ is listed in Table A.1.

Element type	M1
T10	$\mathbf{M}_1 = [\xi^2 \eta^2 \quad \eta^2 \zeta^2 \quad \zeta^2 \xi^2]$
P13	$\mathbf{M}_1 = [\xi^2 \eta^2 \quad \xi^2 \quad \eta^2]$
P15	$\mathbf{M}_1 = [\xi^2(1-\zeta^2) \quad \xi\eta(1-\zeta^2) \quad \eta^2(1-\zeta^2)]$
H20	$\mathbf{M}_1 = [\xi^3 \quad \eta^3 \quad \zeta^3]$

Table A.1 Functions of $\mathbf{M}_1$ for different elements

APPENDIX B

SEARCH FOR THE ELEMENT ENCLOSING A GIVEN POINT

In Chapter 6, it was mentioned that in order to obtain the approximate "exact" solution, it is needed to create a reference mesh. The "exact" solution of a given point can be represented by the recovered stress of this point from the reference mesh. In Section 6.3, it has been explained that in order to obtain the recovered stress from the reference mesh, the key point is to determine which element will enclose the given point. Obviously, the simplest algorithm to locate the element that encloses the integration point is to check all the elements within the reference mesh. While for a reference mesh which has a huge amount of elements, this algorithm will be too time-consuming to be adopted in the application. A feasible way is to create a uniform *background grid* for the reference mesh. In every unit, all the elements that intersect with it will be recorded to a set $\mathcal{E}$, which is belonging to this unit. Then if in the current mesh, the integration point P is located in the i th unit, only the elements within the set $\mathcal{E}$ of the i th unit will be checked. With the assistance of the background grid, the computational time can be greatly reduced.

On the other hand, it also must be point out that how to judge whether an element is intersected with a unit block or not is also a troublesome problem. A large amount of calculation on the intersections between lines and faces has to be taken for a general algorithm. In the current study, there is no need to only record the elements exactly intersecting with the unit, all the elements that may have the probability to intersect with the unit can be sent to set $\mathcal{E}$. In this way, although some elements that do not intersect

Search for the element enclosing a given point

with the unit have been saved in $\mathcal{E}$, all the elements that really intersected with this unit are kept in $\mathcal{E}$ for sure. After set $\mathcal{E}$ is established, every element within $\mathcal{E}$ will be checked, until the one enclosing point P is found. The algorithm to get set $\mathcal{E}$ is given in Box B.1 .

- i) Create an $l \times m \times n$ background grid to enclose the whole domain where l, m, n are integer parameters defined by the user. A unit of the background grid is identified by its indexes I, J, K ($1 \leq I \leq l; 1 \leq J \leq m; 1 \leq K \leq n$).
- ii) Loop over the index i from 1 to N, where N is the number of elements in the reference mesh.
- iii) Loop over the index j from 1 to M, where M is the number of the corner nodes of the i th element.
- iv) Determine the grid unit that encloses the j th node, and add the three indexes of this unit to sets II_i, JJ_i and KK_i respectively.
- v) End for loop j
- vi) Determine the maximum and minimum indexes in set II_i, JJ_i and KK_i , denote them as $I_{\min}, I_{\max}, J_{\min}, J_{\max}, K_{\min}$ and $K_{\max}$.
- vii) Group all the units whose three indexes are within the range $[I_{\min}, I_{\max}], [J_{\min}, J_{\max}]$ and $[K_{\min}, K_{\max}]$ respectively and then add them to set $\mathcal{E}_i$.
- viii) End for loop i

Box B.1 Get set $\mathcal{E}$ for the nodes in reference mesh

When judging whether a point is within a solid element, tetrahedron element will be used as the basic element type for checking. It is well-known that the volume of a tetrahedron can be analytically obtained by calculating the scalar triple product of the vectors formed by the sides (Figure B.1), which is expressed as

$$V = \frac{1}{6} \overrightarrow{BD} \cdot (\overrightarrow{BC} \times \overrightarrow{BA}) \quad \text{B.1)}$$

Search for the element enclosing a given point

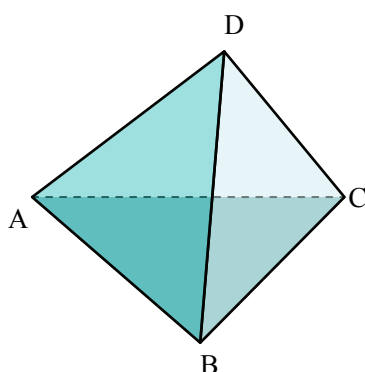


Figure B.1 Volume of the tetrahedron

By using a uniform definition of the sequence of the local nodal labels, it is easy to guarantee that the volume calculated by Equation B.1) for a tetrahedron formed by any element face and an interior node is always positive. While for a node located outside the element, among all the four tetrahedrons formed by it and the element faces, there must be at least one with a negative volume. In this way, it is easy to judge whether a point is within a tetrahedron or not.

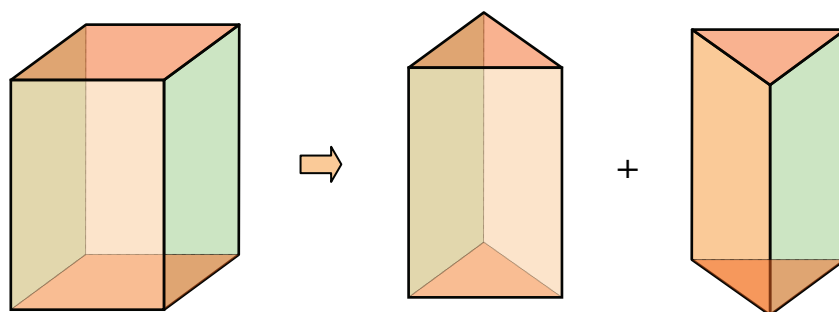


Figure B.2 Split of a hexahedron element

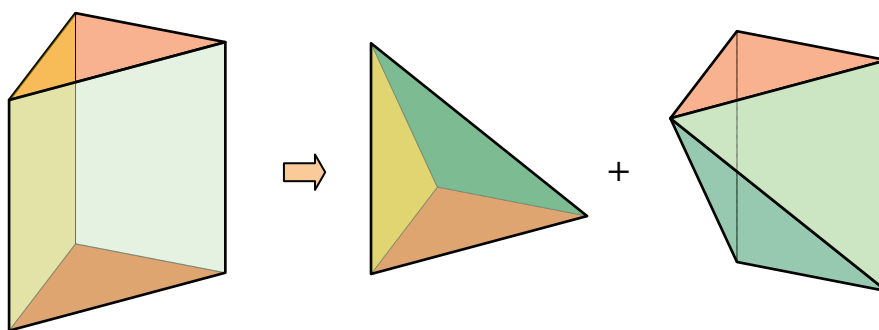


Figure B.3 Spilt of a prism element

Search for the element enclosing a given point

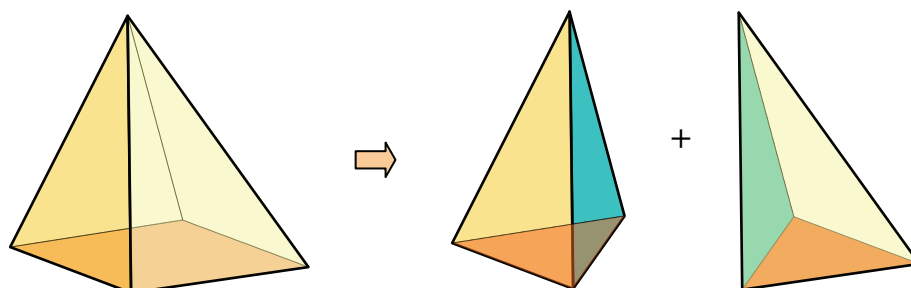


Figure B.4 Split of a pyramid element

For the other solid elements (pyramid, prism and hexahedron) used in the current study, they will be split into several tetrahedrons for checking. A hexahedron can be split to 2 prisms (Figure B.2), a prism can be split to a tetrahedron and a pyramid (Figure B.3), and a pyramid can be split to 2 tetrahedrons (Figure B.4). Thus, all the solid elements used in the current study can be split in to tetrahedrons for checking finally. If a node is within any of the tetrahedron that is created by splitting, then the node is also within the original element definitely.

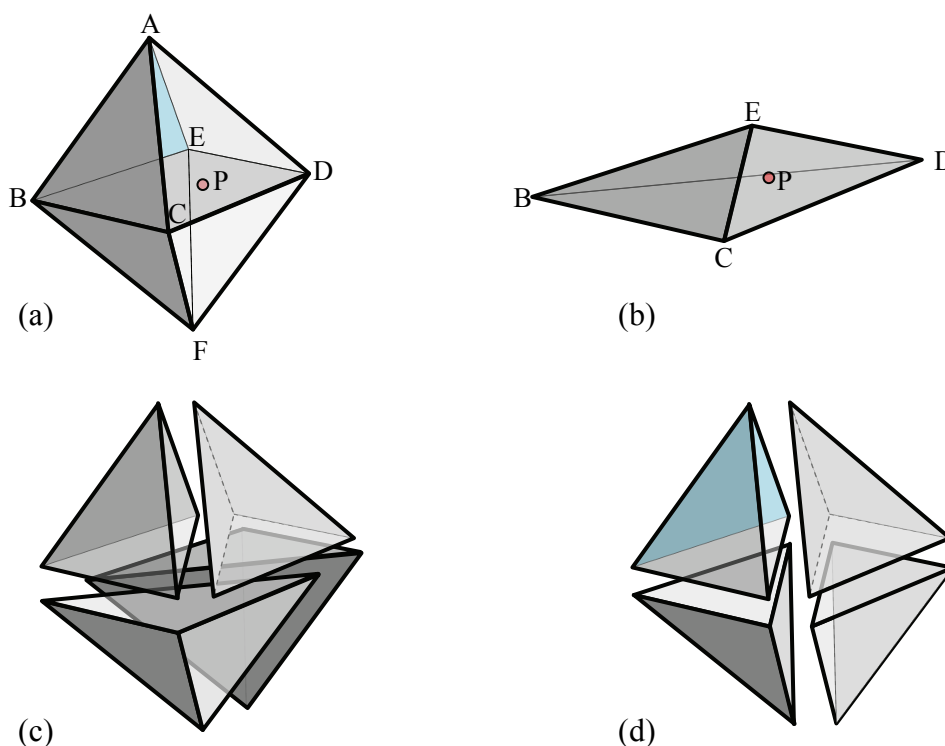


Figure B.5 Split the element that has quadrilateral facet

Search for the element enclosing a given point

It must be cautious in the case when the element has quadrilateral facets. This is because that the 4 nodes of a quadrilateral facet may not be on the same plane. Use pyramid elements for example, in Figure B.5, two pyramid elements are adjacent by a common quadrilateral facet BCDE and P is a node near to this facet (Figure B.5a). If B, C, D, E are not located on the same plane so that they will form a tetrahedron domain (Figure B.5b). If P is just located within this tetrahedron, then the splitting method in Figure B.5c cannot be used because P is not enclosed by any of the four split tetrahedrons, while the splitting method shown in Figure B.5d can guarantee that P is located within one of its generated tetrahedrons.

In order to avoid the case shown in Figure B.5c, for every pyramid element that will be split by dividing its quadrilateral facet, the two splitting ways which are shown in Figure B.5c and Figure B.5d must be both taken.