List of Figures

Figure 2.1: Dividing a complex domain into a triangle and quadrilaterals…………...….8

Figure 2.2: A pre-ordered structured triangular grid………...………………………...8

Figure 2.3: A pre-ordered structured quadrilateral grid...……………………………...9

Figure 2.4: Unstructured triangular mesh………………...……………………………..10

Figure 2.5: Unstructured quadrilateral mesh…...………………………………………10

Figure 2.6: Quadtree decomposition of a simple 2D domain……...……………………11

Figure 2.7: Delaunay criterion illustration……….………...………………………..13

Figure 2.8: Advancing front illustration…...……………………………………………14

Figure 2.9: Paving method applied to a square domain…..……………………………15

Figure 2.10: Splitting triangle method…….……………………………………….....16

Figure 2.11: Combining triangle method, no triangle left in the domain…..…………..17

Figure 2.12: Combining triangle method, two triangles left in the domain………….....17

Figure 3.1: A simple 3D light frame wooden house……………………………………24

Figure 3.2: Showing detailed view of section M-M…………………………………….24

Figure 3.3: Typical FEM model generated in SAP 2000. The stud wall
is modeled as roller supports…………………………………………………..27

Figure 3.4: Typical FEM model generated in SAP 2000. The stud wall
is modeled as spring supports…………………………………………………28

Figure 3.5: Free body diagram for maximum tie down force calculation (length
units are in inches)……………………………………………………………..29

Figure 3.6: Stud wall modeled as roller support and tie down spring
stiffness = 5,000 kips/in…………………...……………………………..31

Figure 3.7: Stud wall modeled as roller support and tie down spring
stiffness = 50,000 kips/in………………...……………………………….....32

Figure 3.8: Stud wall modeled as roller support and tie down spring
stiffness = 500,000 kips/in……………...………………………………..33

Figure 3.9: Stud wall modeled as spring support with stiffness = 76 kips/in.
and tie down spring stiffness = 5,000 kips/in…….………………...……..34

Figure 3.10: Stud wall modeled as spring support with stiffness = 76 kips/in. and tie down spring stiffness = 50,000 kips/in……………..……………...35
Figure 3.11: Stud wall modeled as spring support with stiffness = 76 kips/in. and tie down spring stiffness = 500,000 kips/in…………………………..36
Figure 3.12: Stud wall modeled as spring support with stiffness = 19 kips/in. and tie down spring stiffness = 5,000 kips/in……………………………..37
Figure 3.13: Stud wall modeled as spring support with stiffness = 19 kips/in. and tie down spring stiffness = 50,000 kips/in……………………………38
Figure 3.14: Stud wall modeled as spring support with stiffness = 19 kips/in. and tie down spring stiffness = 500,000 kips/in…………………………..39
Figure 4.1: Interior domain lying on the left of initial generation front……………….43
Figure 4.2: Interior domain lying on the right of initial generation front……………...43
Figure 4.3: A node in 3D vector space…………………………………………………44
Figure 4.4: Line segment AB…………………………………………………………..44
Figure 4.5: Shortest distance between node N and line segment AB…………………..44
Figure 4.6: Generation front……………………………………………………………45
Figure 4.7: Area operator……………………………………………………………….45
Figure 4.8: Intersection operator……………………………………………………….46
Figure 4.9: Valid node set……………………………………………………………...47
Figure 4.10: α quality of some example triangles………………………………………48
Figure 4.11: Pseudo code for triangular element generation process…………………..50
Figure 4.12: Formula for NSF at any point P(x,y)……………………………………...52
Figure 4.13: Domain with multiple openings and multiple constraint lines……………53
Figure 4.14: Square shaped domain in xy-plane………………………………………..54
Figure 4.15: L-shaped domain in xy-plane……………………………………………..54
Figure 4.16: Square-shaped domain with an opening in xy-plane……………………...55
Figure 4.17: Square-shaped domain with a constraint line in xy-plane…………………55
Figure 4.18: Square shaped domain with two openings and two constraint lines in xy-plane………..…………………………………………………………..56
Figure 4.19: Virginia Tech logo in xy-plane…………………………………...……..56

Figure 4.20: A convex domain with an inside point lying to the left of all boundary segments…………………………..……………………………………57
Figure 4.21: A non-convex domain with an inside point lying to the left of all the boundary segments except for the one shown with dash style…..………..58
Figure 4.22: Showing ray tracing method……………………………………………58
Figure 4.23: Rays intersecting vertices of the domain……………………………….59
Figure 4.24: Rays overlapping edges of the domain…………………………………59
Figure 4.25: Showing solution to the problem of ray intersecting the vertices of boundary edges………………………………………………………….60
Figure 4.26: Showing solution to the problem of ray being collinear with the boundary edges……………………………………………………………….60
Figure 4.27: Showing line segments AB and CD…………………………………….61
Figure 4.28: Showing orientation of AB with respect to normal N..…………………....62
Figure 4.29: Showing distance of point from a line segment…………………………...63
Figure 4.30: A simple quadrilateral domain…………………………………………..63
Figure 4.31: Boundary nodes named in continuous order……………………………64
Figure 4.32: Showing subdivision of shorter segments from opposite edges…………..64
Figure 4.33: Dividing opposite edges into same number of subdivisions as the opposite shorter edges…………………………………………………...65
Figure 4.34: Showing quadrilateral element generation process……………………..66
Figure 4.36: Showing a skewed quadrilateral domain………………………………..67
Figure 5.1: Class xyz's attributes and methods (C++ interface)………....……………69
Figure 5.2: Inherited class uvw's attributes and methods. All public and protected attribute and methods of class xyz also belongs to class uvw (C++ interface)……………………………………………………………70
Figure 5.3: Showing polymorphism (operator overloading, C++ interface)……………71
Figure 5.4: Basic architecture of the WoodFrameMesh program…………………...….74
Figure 5.5: Interface of KMDomainBldr class…………………………………………75
Figure 5.6: Interface of KMTxtFileReader class……………………………………..76
Figure 5.7: Interface of the KMDomain class………………………………………..77
Figure 5.8: Interface for the KMFEModelBldr class…………………………………78

Figure 5.9: Interface for KMFEMeshGenerator class…………………………………..79
Figure 5.10: Interface for the KMAFT class……………………………………………80
Figure 5.11: Interface for the KMFEModel class………………………………………81
Figure 5.12: Interface for the KMS2KFileWriter class…………………………………81
Figure 5.13: Interface for the KMGeometricObject class………………………………82
Figure 5.14: Interface for the KMSurfaceObject class…………………………………83
Figure 5.15: Interface for the KMRestraintLineObject class…………………………...83
Figure 5.16: Interface for the KMSpringLineObject class……………………………...84
Figure 5.17: Interface of the KMFrameLineObject class………………………………84
Figure 5.18: Interface for the KMNonLinearLinkLineObject class……………………84
Figure 5.19: Interface for the KMLoadLineObject class………………………………85
Figure 5.20: Interface for the KMSpaceVector class…………………………………...86
Figure 5.21: Interface for the KMNode class…………………………………………...86
Figure 5.22: Interface for the KMLineSegment class…………………………………...87
Figure 5.23: Interface for the KMRestraint class………………………………………88
Figure 5.24: Interface for the KMSpring class…………………………………………88
Figure 5.25: Interface for the KMShell class……………………………………………..89
Figure 5.26: Interface for the KMFrame class…………………………………………89
Figure 5.27: Interface for the KMNonLinearLink class………………………………..90
Figure 5.28: Interface for the KMNodalLoad class……………………………………90
Figure 5.29: Interface for the KMMaterial class………………………………………..91
Figure 5.30: Interface for the KMSection class…………………………………………91
Figure 5.31: Interface for the KMFrameSection class…………………………………..92
Figure 5.32: Interface for the KMShellSection class……………………………………92
Figure 5.33: Interface for the KMNonLinearLinkSection class………………………...93
Figure 6.1: KSModelBuilder acting as interface between KSEcho and KSModel……..98
Figure 6.2: Interface of a new derived element class…………………………………103
Figure 6.3: Interface for a new constraint derived class………………………………104
Figure 6.4: Interface of new section class……………………………………………..105
Figure 6.5: Interface of new solver class………………………………………………106
Figure 6.6: Interface of new numberer class…………………………………………..107

Figure 6.7: Example showing addition of new code (shown in bold) in the main program if a new numberer child class is added……………………..108