Chapter 7

Summary and Future Development

The main aim of the thesis was the development of an object oriented automatic mesh generator in C++ and the verification of models developed by it using the WoodFrameSolver program. The finite element analysis engine was developed by the author and other WoodFrameSolver team members as a separate research project. This chapter presents the summary and scope for the future development of these programs.

7.1 Summary

The mesh generation problem involves numerous geometric computations, but generating automatic meshes is no longer a challenge. Structured mesh generation is easy to automate as compared to unstructured meshing techniques. Several automatic mesh generation algorithms are available in the literature and they have been successfully implemented. Automatic generation of consistent, reproducible, high-quality meshes without user intervention has made the power of finite element analysis accessible to those who are not expert in mesh generation. Also, automation of mesh generation is an important prerequisite for the complete integration of the finite element method with

design processes in computer aided engineering. In this framework, unstructured grid technology has been proven as a promising approach for the generation of good finite element meshes. The use of object oriented design techniques and C++ implementation in developing mesh generation software is a budding concept.

As a part of this thesis, an automatic finite element mesh generator program for wooden houses is developed in C++. Currently, the program is capable of generating complete finite element models of wooden houses incorporating frames, non-linear links, springs, restraints and nodal loads at the desired locations. The finite element mesh generated by the program can be of triangular or quadrilateral type. The triangular mesh may be generated over any arbitrary domain with multiple openings and line constraints. The program implements a popular method of advancing front for triangulation as discussed by Lee and Hobbs (1999). The difference is made by implementing the algorithm using object oriented concepts. Quadrilateral mesh generation is limited to a simple quadrilateral domain with no openings and constraint lines. The amount of time spent in manual generation of complete finite element model of wooden houses has been considerably reduced by automating the whole process. Overall, the use of object oriented design has facilitated the code development and provides a platform for further additions. The program relies on the use of STL as it provides dynamic data structures, algorithms for storage, searching, sorting etc. Efficiency of the program gets improved by the use of the in-built features in STL instead of developing new code. The discussion of the program architecture was presented in chapter 5.

The WoodFrameSolver program was developed by WoodFrameSolver team as a separate research project. The discussion of the program architecture was presented in chapter 6. The use of object oriented design in the development of finite element analysis programs is a popular concept these days and a huge amount of literature and implementation is available on it. WoodFrameSolver is based on this design philosophy. The program is capable of analyzing models generated by WoodFrameMesh program and models generated from SAP 2000. Under the current capability the program can perform linear static analysis on structures having frame, 3 and 4 node shell, non-linear link, spring, and equal constraint elements. The load can be of nodal, frame point, frame distributed or shell pressure type, applied in global directions. The program can also

handle different load combinations. The program generates nodal displacements, frame element forces, non-linear link forces, spring forces and shell stresses inside an output file. Extensive testing of the WoodFrameSolver program has been performed and the results are found to be similar with SAP 2000.

7.2 Future Development

This section presents recommendations for future development of the WoodFrameMesh and WoodFrameSolver programs. As both the programs are developed in the same environment, integration of these two programs is quite possible. In fact the step involving generation of text input file by the pre-processor may be discarded once the performance of these two programs is found satisfactory. Further recommendations include:

7.2.1 WoodFrameMesh

1.) Providing a quadrilateral mesh generation class for any arbitrary domain with multiple openings and multiple constraint lines.
2.) Generation of automatic shell pressure loading, frame point or distributed loading etc.
3.) Adding a new mesh generation algorithm for new types of element generation, for example, 8-node brick element or tetrahedral element.
4.) Parallel computation of meshes for different objects.
5.) Providing graphic features to view the generated models.

7.2.2 WoodFrameSolver

1.) Adding new solvers, i.e., sparse matrix solver, eigen value solver, etc.
2.) Adding new elements, i.e., plane element, 8-node brick element, etc.
3.) Adding new numberer, i.e., PFM renumberer, etc.
4.) Adding new analysis class, i.e., dynamic analysis, non-linear analysis, etc.
5.) Adding new materials, i.e., non-linear material, etc.

6.) Adding new constraints, i.e., rigid diaphragm, plate constraint, rod constraint, etc.
7.) Providing graphic features to see the deformed shape of elements, force diagrams, etc.