Chapter 2

Improvement of Continuity Between Industrial Software and Research One by Object–Oriented Finite Element Formulation of Shell and Plate Element

Sabah Moussaoui
Setif 1 University, Algeria

Mourad Belgasmia
Setif 1 University, Algeria

ABSTRACT

This chapter shows, through the example of the addition of a plate and shell element to freeware FEM-object, an object-oriented (C++) finite element program, how object-oriented approaches, as opposed to procedural approaches, make finite element codes more compact, more modular, and versatile but mainly more easily expandable, in order to improve the continuity and the compatibility between software of research and industrial software. The fundamental traits of object-oriented programming are first briefly reviewed, and it is shown how such an approach simplifies the coding process. Then, the isotropic shell and orthotropic plate formulations used are given and the discretized equations developed. Finally, the necessary additions to the FEM-object code are reviewed. Numerical examples using the newly created plate membrane plate element are shown.

DOI: 10.4018/978-1-5225-7059-2.ch002
INTRODUCTION

The authors can divide all the software of modelization into 4 large families

- **General Search Software**: Which serves as a basis for the testing of innovative models or methods, must be flexible and capable of undergoing substantial changes. They should be conceived as constructs of “modules” independent and easily modifiable.

- **Specialized Research Software**: Models a particular type of phenomena and uses very specific methods. Like general cousins, specialized search software has a rough user interface and the data that the researcher manipulates is of very low level (matrix and vector in common); the commands are Fortran statements.

- **General Industrial Software**: Encompasses a maximum number of models and algorithms. They are equipped with a comfortable user interface and can communicate with many mesh and post-processors. The user here manipulates high-level data (physical structures, geometric data) and the commands are launched to the mouse. Because of their size these software are not very scalable.

- **Specialized Industrial Software, or “Business” Software**: Are often ex-specialized research software that have been made more reliable, cleaned and documented. Laboratories strive to transform their novelties into a reliable product in order to gain fame. Similarly, industrialists naturally seek to quickly transfer the results of research to their code. One should therefore improve the continuity and the compatibility between software of research and industrial software that is to say to seek to professionalize the research software too often “artisanal”.

In the last decades, the finite element method has gained a large acceptance as a general tool for modeling and simulating physical systems. It has become, in fact, the most widely used technique applicable in fields such as solid mechanics, fluid mechanics, electromagnetics etc.... In recent years, easy maintenance, extensibility and flexibility of the finite element code, encouraged many researchers to change from procedural to object oriented programming in order to make the finite element software more flexible.

The Object-Oriented programming approach appeared in the seventies and found its first convivial concretization with the SmallTalk language at the beginning of the eighties. Applications in the finite element field appeared around 1990. (Dubois, Zimmermann & Bomme, 1992 and Dubois & Zimmermann, 1993), and since then its use has spread. Object-Oriented approaches yield more modular programs than the standard procedural approach.

In this Chapter, an Object-Oriented approach of the finite element method for the study of bending of isotropic and orthotropic plates in dynamic and shell element in statics is presented.

The authors, shown how Object-Oriented programming features, through the concept of Object, Class, Inheritance and Polymorphism, make it possible to overcome those limits, and deals with plate and shell theory. Displacement and strain approximations are described, the equations of motion developed, the basics behind Mindlin-Reissner plate and shell theory are recalled, and stiffness matrices for this formulation are developed. A section is concerned with the addition of the aforementioned plate and shell element into the FEMObject code described elsewhere, (Commend & Zimmermann, 2001). Object-Oriented programming is shown to easily accomodate the addition of a plate and shell classes, thus demonstrating code extendibility. Finally, the authors present numerical results for isotropic and orthotropic thin plates and shells with conclusion at the end.