CHAPTER 7

Finite Element models validation methods

7.1 Introduction

In the previous Chapters we saw the various aspects of a finite element calculation: from result interpretation to the way of avoiding mistakes that could lead to erroneous conclusions, up to the modelling of particular structural conditions. Nevertheless all these speeches are useless if we do not have a good confidence that the model behaves, at least within a certain range of tolerance, in the same way that the actual structure would do in service conditions. Therefore, assuming that the structural engineer has used all the instruments at his disposal in order to realize a "perfect" model, it is still necessary to evaluate somehow its reliability. A part from the checks that the user has to perform before "running" the solution some other instruments exist that can give further information on the model quality after the solver has generated the results. And this is usually done through the messages that the calculation code gives the user by printing them in an ASCII file; we will classify this information under the nomenclature "numerical validation".

Generally, a part from a big confidence based on a great experience acquired on similar structures, before starting the building of a structure a prototype of it is built in order to execute some structural tests; we can think of a simple test using the nominal loads in order to check its resistance and the possible residual deformations after load removal or of the measurement of the strains through strain gauges techniques, photoelasticity or others; we will group these methods under the nomenclature "experimental validation"

7.2 Numerical validation

As we mentioned in Chapter 5, graphical pre-processors generally have some instruments that should avoid the presence of "poor" elements from the numerical point of view. Therefore we will assume that the model has undergone all the required checks and that is then ready to be passed to the solution algorithm.

If the calculation code does not give an error message we can be satisfied because often it is easy to forget a constraint or to write a card in the wrong way (i.e. writing