

A C++ object-oriented programming strategy for the implementation of the finite element sensitivity analysis for a non-linear structural material model

Lluís Gil, Gabriel Bugada*

Department of Strength of Materials and Structures in Engineering, Technical University of Catalonia UPC, Campus Nord UPC, Mòdul C1, C/ Gran Capità s/n, 08034 Barcelona, Spain

Received 6 September 2000; accepted 26 June 2001

Abstract

The Finite Element Method (FEM) has become the most popular numerical method for solving a wide variety of complex engineering problems. However, from the programming point of view, when a FEM program has a lot of computational capabilities it is very difficult to maintain and enlarge the program codes. Recently the object-oriented programming paradigm has become a powerful method for overcoming such difficulties. This paper contains the description of an open environment for the FEM, written in C++, with explanations about how the sensitivity analysis and the non-linear material behaviour (damage models) have been taken into account. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Sensitivity analysis; Non-linear analysis; Object-oriented programming

1. Introduction

The continuous increase of the capabilities of hardware and software tools allows to solve very complex structural problems. Nowadays, the use of new materials, new industrial processes and coupling between different physical phenomena can be considered altogether. This has produced a progressive substitution of the traditional experimental methods by the less expensive numerical analysis.

Since the very beginning, the task of programming the Finite Element Method FEM has been increasing in difficulty over the years. The more that engineers want to compute the more difficult it becomes to implement. The reasons rise in the difficulty of the mathematical boundary problems and in the language and programming strategies.

In this paper the authors present an overview of the FEM programming drawbacks and comment their experience with different programming strategies. Finally, an object-oriented approach for the implementation of non-linear material behaviour and its sensitivity analysis is described.

2. FEM from object-oriented programming

The problem of programming FEM with object-oriented techniques has frequently been treated in literature. Forde [1] made the first and notable approach to the problem statement and solution, and it was also followed by contribution [2]. Zimmermann and Dubois-Pèlerin presented the main advances during the last decade with a very active team in this field (see Refs. [3–8]). Finally, some references must also be made to Kong [9] and Mackie [10]. Most of the mentioned authors offered their personal point of view with regards to object design in FEM for problems, which range from linear elastic to non-linear and dynamics. All the contributions have been very valuable and rich, and the scope goes from philosophical discussions linking FEM and objects to the design of a FEM shell as a natural language for problem-solving in engineering. Moreover, the natural approach to the problem depends on personal views, so that the final implementation is not an easy task. Notice that, on one hand, FEM behaves as a procedural method with a clear flow but, on the other, the method contains abstract data like nodes, material properties, etc. In spite of the advantages of object-orientation, this approach emphasises data abstraction and the programmer tends to forget that FEM is a procedure, hence FEM should be programmed balancing both concepts.

* Corresponding author.

E-mail address: bugeda@cimne.upc.es (G. Bugada).

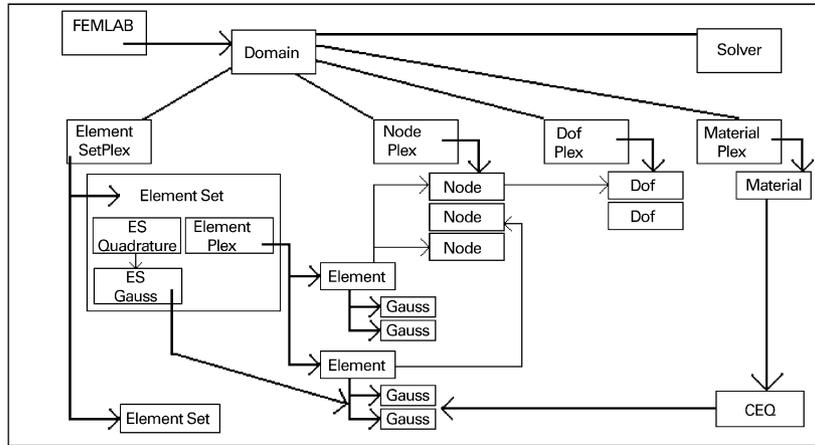


Fig. 1. Relationship between the different FEM classes.

3. The OO methodology in FemLab

FemLab is an object-oriented finite element framework. FemLab was developed [11] in the International Centre for Numerical Methods CIMNE and it becomes a good platform for further developments. FemLab provides a basic general structure through a hierarchy of classes that allows the programmer to build a FEM code adapted to each necessity. The programmer selects the type and the part of the code he needs depending on his mathematical model. He/she chooses the appropriate classes and modifies them conveniently. The encapsulation property provides the security that the new code will not interfere with the rest of existing classes. In addition, the programmer can create completely new classes or can derive some others using the inheritance properties and, finally, he/she can redefine some functions using the polymorphism property and help to maintain a coherence in the lecturing of the code. Therefore, it can be said that the FemLab code was written with the objective of exploiting as much as possible the possibilities of OO and C++.

The original version of FemLab allowed for the static linear analysis in one, two and three dimensions with a big variety of elements. Inside of FemLab there are complex programming concepts like: the Sets of data, Plexes or

linked lists of objects, libraries of vectors and matrices, a symbolic language for the matrix operations and, specially, the hierarchy of classes that represents the knowledge of the FEM.

The FemLab context provides a code with a hierarchy of classes where the main relations are established in Fig. 1.

There are three main classes:

- (i) FemLab class that performs the analysis of the problem. See Fig.2.
- (ii) Domain class that contains the problem data.
- (iii) Solver class that solves a system of equations.

The Domain class contains lists of elements, nodes, degrees of freedom, materials and the solver. Each list contains a different type of object. The use of an object-oriented paradigm makes it possible to create a structure that is logic and consistent with the finite element theory that has been established at the moment of defining the relations among: sets of elements, elements, integration points, constitutive equations and materials, as well as between nodes and elements.

The list of degrees of freedom, DofPlex, contains all the existing degrees of freedom in the finite element mesh. For example, in a plane strain elasticity problem each node has

```

class FemLab // Master class of the Finite Element Laboratory
{
    Domain* Mesh; // The mesh data

    void ConstructMesh      (); // read mesh data
    void StartNewInterval   (); // following and advance and correction procedure
    void AdvanceAndPredict  ();
    void UpdateDofs         ();
    void ApplyBoundaryConditions ();
    void StartNewIteration  ();
    void AssembleMatrixes  ();
    void ComputeUnknowns   ();
    int HasConverged       ();
    void OutputResults     (); // the end
};

```

Fig. 2. Class FemLab contains the whole Finite Elements problem.

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات