Advances in Engineering Software 48 (2012) 70–88

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Object-oriented design to automate a high order non-linear solver based on asymptotic numerical method

Arnaud Lejeune [a,*], Fabien Béchet [b], Hakim Boudaoud [c], Norman Mathieu [d], Michel Potier-Ferry [d]

[a] Institut Femto-ST, UMR 6174, Département Méc'Appli, 24 rue de l'épitaphe, 25000 Besançon, France
[b] TEMPO, EA 4542 Université de Valenciennes et du Hainaut Cambrésis, 59313 Valenciennes, France
[c] Equipe de Recherche sur les Processus Innovatifs, Ecole Nationale Supérieure en Génie des Systèmes Industriels, 8, rue Bastien Lepage, BP 90647, 54010 Nancy, France
[d] LEM3, Laboratoire d'Etude des Microstructures et de Mécanique des Matériaux, UMR 7239, Ile du Saulcy, 57045 Metz, France

ARTICLE INFO

Article history:
Received 3 November 2011
Received in revised form 23 January 2012
Accepted 18 February 2012
Available online 29 March 2012

Keywords:
Asymptotic numerical method
Object Oriented Programming
Non-linear solver
Generic programing
Automatic computation
C++

ABSTRACT

The Manitoo library is devoted to the resolution of analytical non-linear problems using a high order method called asymptotic numerical method. We describe here the Object Oriented design of this library and especially the choices made to obtain a quite generic and flexible numerical solver.

Through classical examples, we present a comparison with some existing tools implemented in Matlab and Fortran 77.

1. Introduction

For about twenty years, many works on the resolution of non-linear problems using Asymptotic Numerical Method (ANM) have been proposed [1] and applied to a wide range of problems in fluid and solid mechanics. For instance, various applications concerning the design of marine structures [2], non-linear vibrations [3], sheet metal forming [4], biomechanics [5] or multi-scale instabilities [6] have been implemented in many research informatics tools using Fortran 77, Fortran 90 [7,8], Matlab [9,3] and also in an industrial code [10]. Within ANM, a solution branch is computed using series expansion. The step length of this branch is then defined a posteriori and it yields naturally automatic path following techniques. This automation of the non-linear computation is very important in many cases and especially for the numerical computation of physical problems involving instabilities [11–16]. The robustness of the algorithm is assessed for instance in [2], where thousand thin shell computations have been performed in order to predict thin shell buckling with random imperfections. In Computational Fluid Dynamics, the discretized problem often involves millions of degrees of freedom so parallel implementations are needed. The robustness and efficiency of ANM have been established to solve such large scale problems [17].

ANM needs high order derivatives that can be obtained by recurrence formulae. These recurrence formulae computation is easy for algebraically simple equations, as Navier–Stokes equations, but it becomes more intricate for many other physical problems [18]. A first answer to this question is the MANLAB software [9] which permits to solve generic problems with few unknowns, providing the problem written in a quadratic form. Another approach, called DIAMANT and based on Automatic Differentiation (AD) by operator overloading, has been recently proposed [19,20] and applied to small size academic problems in the Matlab and Fortran contexts [21,22]. Due to Object Oriented limitation of these languages [23], it seems then difficult to obtain a really generic, reusable and efficient ANM library. Moreover, as for most of informatics libraries, and based on a 15 years old experience, we know that the ANM library mainly requires maintainability, a wide extensibility and portability. Indeed, as mechanical models based on finite elements (or other approximation methods) often need a large number of degrees of freedom, we have to consider the use of parallel computing.

Since the 90's, Object Oriented Programming has been commonly used to design complex scientific applications. As mentioned in the context of finite element method by [24], object oriented programming permits to develop numerical tools with portability on different computer architectures such as clusters, which was not possible with the prior sequential Fortran codes. Moreover,

* Corresponding author. Tel.: +33 381666024; fax: +33 381666700.
E-mail addresses: arnaud.lejeune@univ-fcomte.fr (A. Lejeune), fabien.bechet@univ-valenciennes.fr (F. Béchet), hakim.boudaoud@ensgsi.inpl-nancy.fr (H. Boud aoud), Norman.mathieu@univ-metz.fr (N. Mathieu), michel.potierferry@univ-metz.fr (M. Potier-Ferry).

0965-9978/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2012.02.012

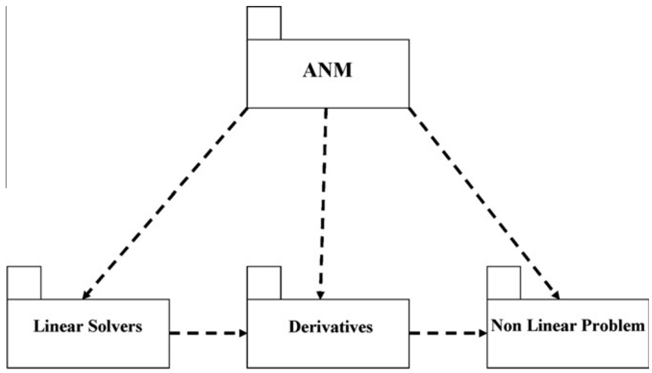

Fig. 1. Analysis and model separation.

inheritance and polymorphism are important characteristics to obtain a quite generic solver. One could argue for the loss of performance due to the overhead of oriented object implementation. Such a loss is not so obvious and Veldhuizhen demonstrates that C++ could surperform Fortran [25,26].

Despite of the propagation of object oriented and C++ programming in the field of numerical simulation of engineering problems, the ANM community has not developed an up-to-date numerical tool.

Considering all these remarks, we have been developing a new C++ library, called MANITOO which is devoted to the resolution of non-linear problems with ANM, since 2008. The main goal of this tool is to reduce development costs without losing computational performance compared to the former library developed in Fortran 77. Here we propose to deal with the object oriented design of Manitoo which has never been published and is about to be mature.

In the second part, we make a description of the ANM and present the corresponding algorithms while the third part is devoted to the Object Oriented design of the library. The fourth part shows some applications and comparisons with existing tools. We conclude with some remarks on future developments.

2. General structure

ANM consists in solving an analytical non-linear problem with a path-following (or continuation) method associated with a high order perturbation technique. First, unknowns are expanded in Taylor series with respect to a scalar path parameter. Then the non-linear problem, as a problem depending on unknowns, is also expanded in Taylor series leading to a system of linear equations at each order. Expressing high order equations with respect to Taylor coefficients requires a hard programming work. Coupling ANM with AD (namely the DIAMANT approach) allows to automate the computation of Taylor series terms in a peculiar direction using well-known recurrence formulae.

As in [27] and to obtain a generic library, the model describing the non-linear problem is distinguished from the analysis. Moreover, we split the analysis in ANM solver and linear solver.

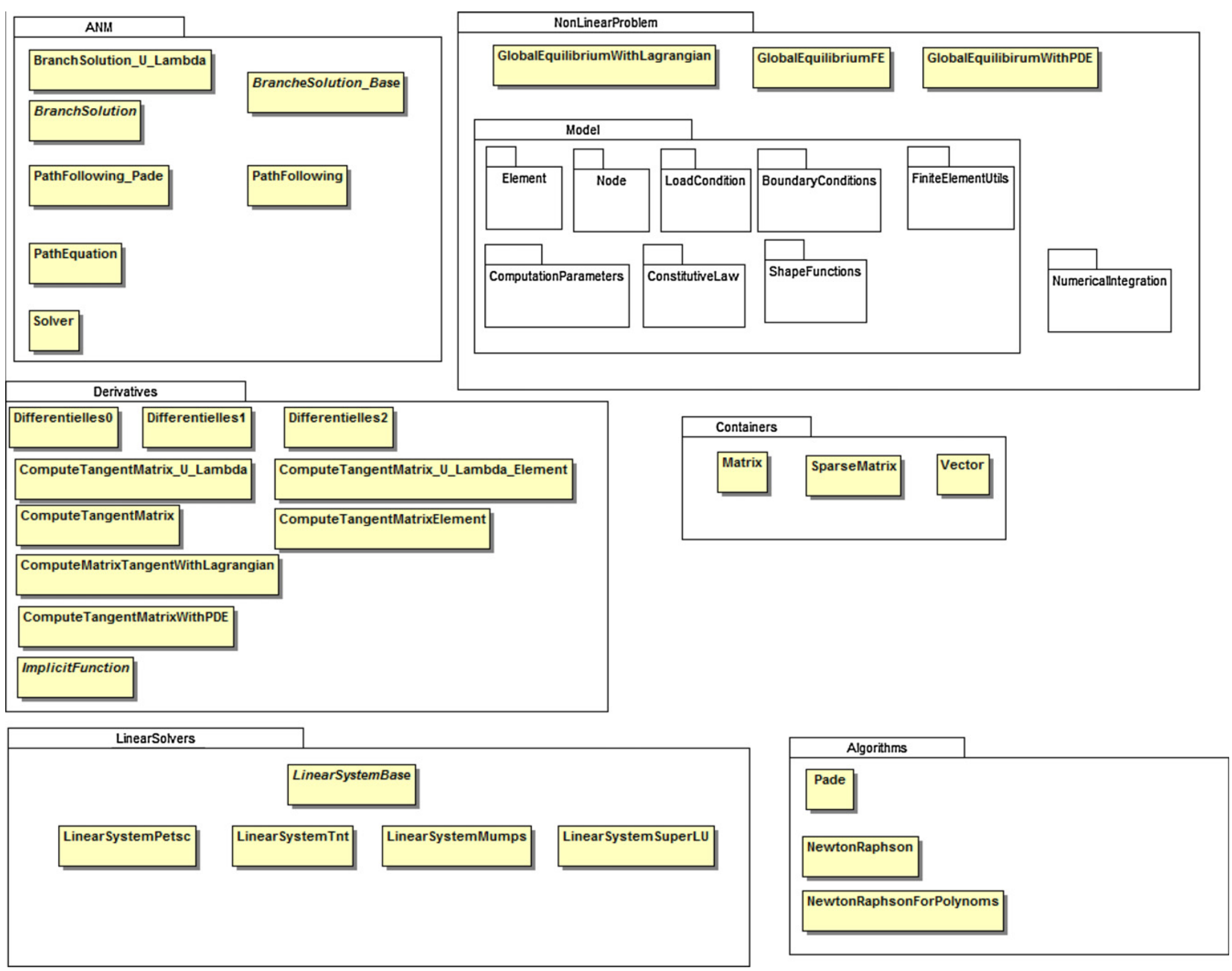

Fig. 2. Some packages and classes of the MANITOO Library.