Technical note: Open-source finite element software – from the early to the present

The dawn of the finite element method (FEM) can be dated back to the aeronautical and civil structural mechanics of the mid- and late-fifties (Turner, Clough, Martin, Topp, Argyris), with roots in the earlier discretization methods of Hrennikoff (lattice analogy, 1941) and Courant (triangular subregions, 1943) or even in the variational methods of Galerkin and Ritz (early 1900s) as well as Rayleigh (late 1800s) [1, 2, 3]. The early developments of the finite element software, in turn, have walked hand in hand with the development of computers: from analog to digital; from room-sized to desktop and laptop; from central to personal. As a link between the abstract mathematical method and concrete computers, the developments of operating systems and graphical user interfaces (especially the change from textual to visual has been essential) and programming languages (especially the first version of Fortran [4] from 1957 and the first standard from 1966 have been crucial) have eventually connected a wide range of highly educated professionals to the FEM: mathematicians, physicists, computer scientists and engineers from several disciplines.

The first pieces of finite element software were either academic or industrial in-house codes for specific purposes. In the mid-sixties, the National Aeronautics and Space Administration (NASA) decided to order from MacNeal–Schwendler Corporation (MSC) a general-purpose FEM-software NASTRAN (NASA STRucture Analysis; developed under the name GPSA, General Purpose Structural Analysis) which was released to NASA in 1968 and to the public in 1971 [1, 2]. From this endeavor, MSC published in 1971 the first commercial FEM-software MSC/Nastran [5]. Other commercial versions of NASTRAN are currently available from several vendors [6]. Another FEM-software of the time was SAP (Structural Analysis Program) which was published in 1970 by Wilson (University of California, Berkeley) and has then formed the core of many commercial FEM-software [1]. Today, even after multiple acquisitions (by the larger players) during the past decades, there are still tens (if not hundreds) of special or generic commercial finite element software suites available. Beside the commercial market, there are several free – even open-source – FEM-packages (under different license types), typically developed by university research groups or combines. Analogously to the commercial counterparts, some of these packages are more general than others, including capabilities for (coupled) multi-physics. In the present note, only three different examples are addressed – more can be found via the related (incomplete) list of a Wikipedia article [7], for instance.

Elmer [8, 9] (from 1995) is a free and open-source (from 2005) FEM-software for multi-physics, mainly developed by CSC – IT Center for Science [8]. Noteworthy, the software package includes a graphical user interface (ElmerGUI). The Elmer Models Manual [9] lists a laudable collection of

application fields, each including several problem types: Models of fluid mechanics and transport phenomena; Models of solid mechanics; Models of acoustics; Models of electromagnetism; Other physical models; Free surfaces, phase change and particle dynamics.

FEniCS [10, 11, 12] (from 2003) is a free and open-source software platform for solving problems based of partial differential equations, with a high-level programming interface (Python or C++) for describing the problem set-up. FEniCS has been organized as a collection of components: DOLFIN (a C++/Python finite element library); FFC (the FEniCS form compiler); FErari (an optimizing compiler for variational forms); FIAT (numerical construction of finite element basis functions); Instant (just-in-time compilation of C/C++ in Python); SyFi and SFC (symbolic finite elements and form compilation); UFC (a finite element code generation interface); UFL (a finite element form language), Unicorn (a unified continuum mechanics solver). The documentation of FEniCS includes a set of books (e.g., [10, 11]), documented demonstration programs and reference manuals [12].

Code_Aster [13] (from 1989) is a free (from 2001) and open-source FEM-software for thermo-mechanical analysis of solids and structures, developed mainly by the Advanced Mechanical Analysis department of Électricité de France (EDF) research and development department (ERMES, ElectRo-Mechanics Studies). Salome-Meca, instead, is an open-source software environment combining Salome, a generic pre- and post-processing platform, and Code_Aster. A related example of fatigue analysis has been published in Journal of Structural Mechanics earlier this year [14].

References

[1] R. W. Clough and E. L. Wilson. Early finite element research at Berkeley, Proceedings of the Fifth U.S. National Conference on Computational Mechanics, August 4–6, 1999 (1999).
[2] W. K. Liu, S. Li and H. S. Park. Eighty years of the Finite Element Method: birth, evolution, and future, *Arxiv* (2021). arXiv:2107.04960
[3] Finite element method, Wikipedia (2021) https://en.wikipedia.org/wiki/Finite_element_method
[4] Fortran, Wikipedia (2021) https://en.wikipedia.org/wiki/Fortran#FORTRAN_66
[5] Analysis Origins – MSC and NASTRAN, NAFEMS https://www.nafems.org/blog/posts/analysis-origins-msc-and-nastran/
[6] Nastran, Wikipedia (2021) https://en.wikipedia.org/wiki/Nastran
[7] List of finite element software packages, Wikipedia (2021) https://en.wikipedia.org/wiki/List_of_finite_element_software_packages
[8] Elmer – CSC Company Site https://www.csc.fi/web/elmer
[9] Elmer – Community Site http://www.elmerfem.org
[10] A. Logg, K.-A. Mardal, G. N. Wells et al. *Automated Solution of Differential Equations by the Finite Element Method*, *The FEniCS Book,* Lecture Notes in Computational Science and Engineering, Springer (2012). https://doi.org/10.1007/978-3-642-23099-8
[11] H. P. Langtangen and A. Logg. *Solving PDEs in Python*, *The FEniCS Tutorial I,* Springer (2017). https://doi.org/10.1007/978-3-319-52462-7
[12] FEniCSx https://fenicsproject.org/
[13] Code_Aster https://www.code-aster.org/
[14] I. Valkonen. Utilizing an open-source code in fatigue analysis, Journal of Structural Mechanics, 54, 80–94 (2021). https://doi.org/10.23998/rm.103377

Jarkko Niiranen
Department of Civil Engineering, Aalto University
PO Box 12100, 00076 AALTO, Finland
jarkko.niiranen@aalto.fi