6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
1115 June 2018, Glasgow, UK

OOFEM.ORG - PROJECT STATUS, CHALLENGES AND NEEDS

BOŘEK PATZÁK[1], MIKAEL ÖHMAN[2], MARTIN HORÁK[1], VÍT ŠMILAUER[1], MILAN JIRÁSEK[1], FILIP KOLAŘÍK[1] AND EDITA DVOŘÁKOVÁ[1]

[1] Czech Technical University in Prague, Faculty of Civil Engineering, Department of Mechanics
Thákurova 7, 166 29 Prague 6, Czech Republic
borek.patzak@fsv.cvut.cz

[2] Chalmers University of Technology, Department of Physics
Fysikgården 2B, 41258 Göteborg, Sweden
mikael.ohman@chalmers.se

Key words: Finite element solver, numerical modelling, open-source, interoperability

Abstract. The paper describes the object-oriented design of OOFEM, a general, open-source finite element solver. The design combines several design approaches, that contribute to highly modular and extensible structure and that allowed sustainable development over the last two decades. The current capabilities are briefly presented. The aim of the paper is also to discuss some challenges and needs not only for the future development of the code, but also for the open-source modelling community

1 INTRODUCTION

OOFEM is multi-physics, parallel finite element solver with object oriented architecture [1]. Its development started in 1993 at Czech Technical University as a part of Ph.D. thesis of the first author on crack propagation in concrete structures. In 2001 the project has been released under open source license. Since then the the project evolved from initially solid mechanics solver to multi-physics solver with many unique capabilities. The code has been used to solve industrial problems in many fields including material design, civil, petroleum, and bio-medical engineering. At present, the project is being developed by international community, with major contributions from research teams from Czech Technical University and Chalmers University.

The current multi-physics capabilities cover solid mechanics, transport and selected fluid-mechanics problems, many can be solved in parallel on systems with distributed as well as shared memory architectures, supporting static as well as dynamic load balancing for optimal scalability to account for changing processor loads during problem solution. The solver interfaces to several linear sparse libraries including PETSc [2], SuperLU [4], and Pardisso [3]. The general features include also the full restart capability, possibility to run individual problems in a staggered scheme allowing to pass solution fields and variables along complex workflows. For selected problems, automatic h-adaptive refinement

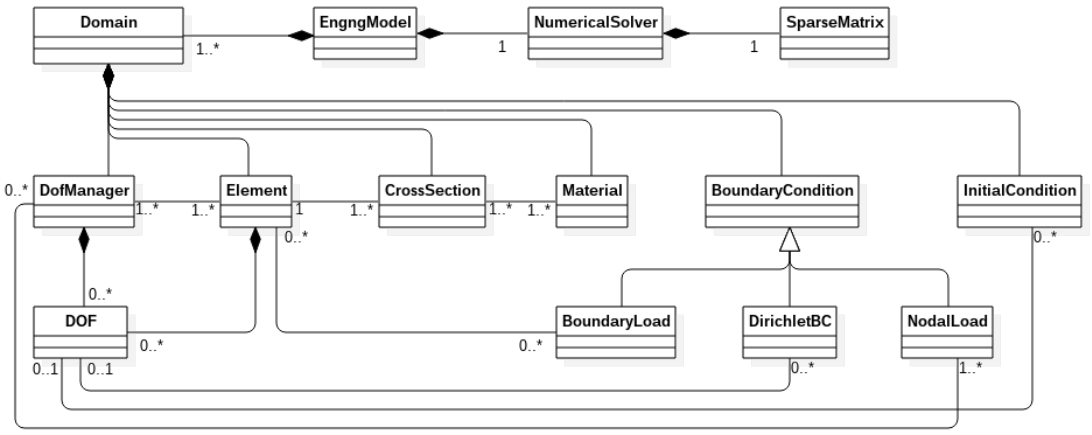

Figure 1: Top level class structure of OOFEM

and variable transfer is available. Solver supports eXtended and Isogeometric element formulations.

The solid mechanics module covers linear and non-linear static and dynamic solvers, element library covering all types of finite elements, rich material library including viscous, plasticity and damage based constitutive models (with particular strong set of models focused on modeling of concrete). The thermal module allows to solve stationary and transient thermal and coupled heat & mass transport problems. Fluid dynamic module allows to solve stationary and transient incompressible problems for single as well as inmiscible fluids with free surface tracking.

OOFEM is implemented in C++ programming language. At present, the code consists of approximately 240k lines of source code. The solver also provides Python interface, allowing to set-up and execute simulations.

2 DESIGN

The design of OOFEM follows the object-oriented approach. The hierarchy of abstract classes is established to represent key components, such as problems, linear solvers, sparse storage formats, finite elements, constitutive models, boundary and initial conditions, or solution fields, for example. The role of abstract classes consists in defining a common interface that has to be implemented by particular component, allowing to manipulate all components of the same type regardless their implementation details.

The problem to be solved is represented by class derived from abstract *EngineringModel* class, which is responsible for assembling the governing equation usually by assembling the individual contributions from elements and boundary conditions. The governing equation is then solved with the help of suitable instance of *NumericalSolver* class. The different sparse storage schemes for system matrices are provided, represented by a hierarchy of classes derived from *SparseMatrix* class. The above mentioned classes allow for mutually independent problem formulation, numerical solution and storage scheme selection. The

problem can consist of one or more discretized spatial domains, represented by the *Domain* class, which keep the list of elements, nodes (defining the element geometry), cross sections, materials and boundary and initial conditions. Most of the relations can be conveniently expressed using sets, allowing to apply individual properties (such as materials, boundary and initial conditions, etc) to a group of nodes or elements.

The individual elements, derived from parent *Element* class, manage its nodes, links to corresponding cross section (representing cross section, keeping the link to related constitutive model) and list of integration rules, containing individual integration points, keeping the loading history via set of internal variables defined and updated by corresponding material model. The element implementation is facilitated by interpolation classes providing methods to evaluate the interpolation functions and their derivatives. This object oriented design relying on abstract interfaces defined by top level classes essentially leads to highly modular structure and allows to implement new components without the need of detailed knowledge of the rest of the system.

3 CHALLENGES

The aim of the paper is also to discuss some challenges and needs not only for the future development of the code, but also for the open-source modeling community. As for any open source project, the active user base is essential. Having in mind the size of potential community on a global scale, we have to actively disseminate our work and bring more people to using open-source simulation tools. In many areas this is a challenge, especially when approaching or collaborating with industrial partners.

In a long-term perspective, we believe it is essential to focus on interoperability between individual simulation tools. Each software has its strong capabilities and it makes a lot of sense to combine different simulation tools to solve complex problems. The open source modeling community can be the driving force to establish an open interoperability standard, allowing exchange of the simulation data and steering of individual modeling tools to create complex simulation workflows. There is a number of projects focusing on development of open simulation platforms. The importance of interoperability in modeling has been recognized by funding agencies, most notably there is EU supported European Material Modeling Council, which activities also involve interoperability in material modeling [5]. One of the objectives is to formulate European Materials Modeling Ontology (EMMO) that should enable semantic interoperability for models and data.

The community should also look for ways how to convince the funding agencies and get a public funding for open source modeling tools. In most of the cases, the projects are supported indirectly, by various research projects. While this allows for development in specific direction, it typically does not allow to support fundamental development which is essential.

4 EXAMPLES

The OOFEM has been used to a wide class of problems. Here we show several representative examples, illustrating the use of the solver for solving engineering problems

from different domains. More application examples can be found on project website [1]

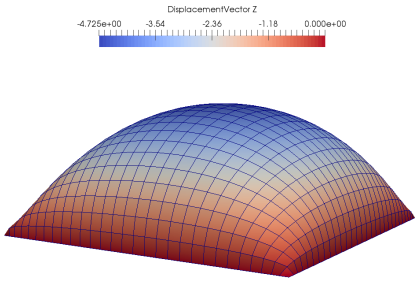

Figure 2: Simulation of a square rubber membrane loaded by the internal pressure, see [6, 7] for the application of the axisymmetric version of the membrane to circular optical liquid lenses

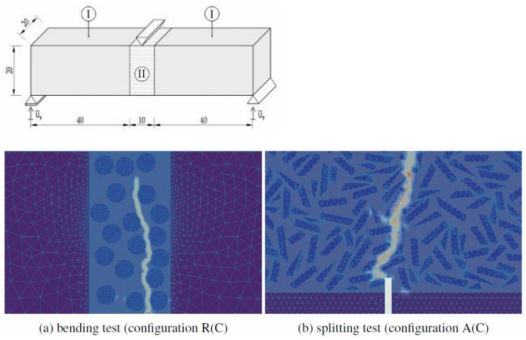

Figure 3: 2D Finite Element Analysis of Aggregate Influence On Mechanical Properties Of Mortar Samples

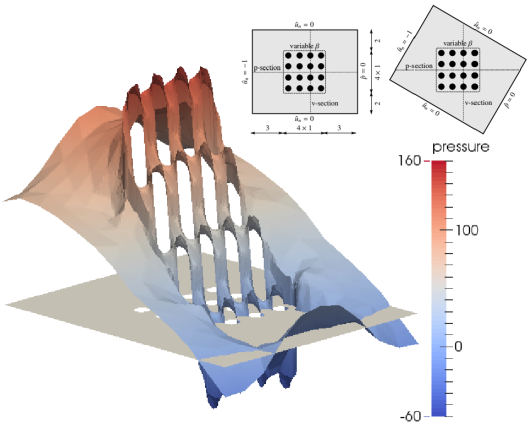

Figure 4: Illustration of pressure profiles of an unidirectional flow of concrete around reinforcing bars which have a significant influence on casting process obtained by multicsale flow analysis [6]

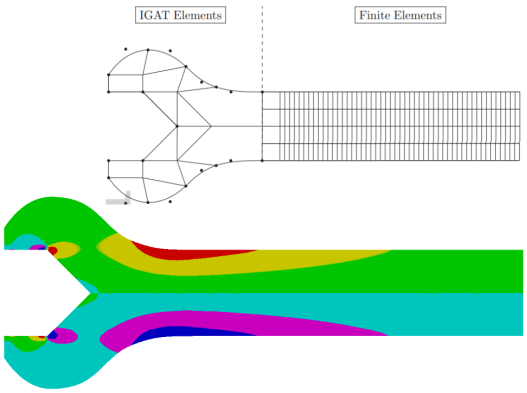

Figure 5: Stress distribution in a spanner simulated using novel IGAT technique based on Hybrid-Trefftz element approximation inside the element domains and Isogeometric analysis on the boundaries [9].

5 CONCLUSIONS

The paper presents an overview of OOFEM project, providing a modular and extensible object oriented environment environment for finite element modeling. The code is disptributed under open sourse LGPL license and has active international development team.

ACKNOWLEDGEMENTS

This work was supported by the Grant Agency of the Czech Republic, under Project No.: 16-20008S.

REFERENCES

[1] Patzák,B. and et.al: *OOFEM project*, www.oofem.org, 2018.

[2] Balay,S. and Abhyankar, S. and Adams, M.F. and Brown, J. and Brune, P. and et al. *PETSc Web page*, http://www.mcs.anl.gov/petsc, 2015.

[3] PARDISO Solver project, https://www.pardiso-project.org/, 2018.

[4] Li, X.S.: *An Overview of SuperLU: Algorithms, Implementation, and User Interface*, ACM Transactions on Mathematical Software (TOMS), vol. 31, no. 3, 2005.

[5] The European Materials Modelling Council, https://emmc.info, 2018.

[6] Pokorný, P. and et.al: *Calculation of nonlinearly deformed membrane shape of liquid lens caused by uniform pressure*, Applied optics, volume 56, number 21, pages 5939–5947, 2017.

[7] Pokorný, P. and et.al: Deformation of prestressed liquid lens membrane, Applied optics, volume 56, number 34, pages 9368–9376, 2017.

[8] Kolařík,F. and Patzák,B. and Zeman, J. *Computational homogenization of fresh concrete flow around reinforcing bars*, Computers & Structures, 2017

[9] Horák, M. and Patzák,B. and Novák, J. *An Isogeometric Extension of Trefftz Method for Elastostatics in Two Dimensions*, International Journal for Numerical Methods in Engineering, accepted