MoFEM: An open source, parallel finite element library

25 April 2019

Introduction

`MoFEM` (Mesh-Oriented Finite Element Method) is a C++ library for managing complexities related to the finite element method (FEM). FEM is a widely used numerical approach for solving partial differential equations (PDEs) arising in various physical problems and engineering applications. `MoFEM` is developed to provide free and open source finite element codes, incorporating modern approximation approaches and data structures, for engineers, students and academics. It was primarily designed to solve crack propagation in nuclear graphite bricks (radiated and oxidised) used in Advanced Gas-cooled reactors (see Fig. 1).

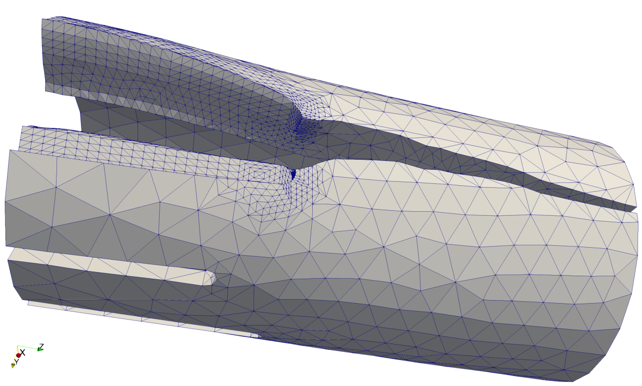

Figure 1: Fractured brick made from nuclear graphite.

The need for solutions to increasingly complex problems demands control over numerical errors; otherwise, we will be unable to distinguish discretisation artefacts from the real physical phenomena. A brute force approach based on a pure *h-adaptivity* leads to a low polynomial convergence rate and relies on the machines computing power. Since we like to solve bigger and more complex problems, no matter how big computer we will have, it will be never enough. Therefore, it is insufficient to have total control over numerical errors. A more sophisticated approach was paved by Ivo Babuska et al. (Babuška and Guo 1992), who showed that if one could increase at the same time the polynomial order and the mesh density, i.e. employ *hp-adaptivity*, the exponential convergence is achievable. This has been seen as the 'Holy Grail' of the numerical methods.

However, raising the order of approximation comes with a cost of the algebraic solver time and the matrix assembly time are increased. Those two issues can be tackled independently. Use of multi-grid solvers can reduce algebraic solver time exploiting a hierarchical approximation base (Ainsworth and Coyle 2003, @fuentes2015orientation), available in MoFEM. This approach is ideal for elliptic problems such as solid elasticity, or with synergy with block solvers. However, for some problems efficiency, the bottleneck is assembly time, e.g. acoustic wave propagation. For that case, different approximation bases, e.g. Bernstein-Bézier base (Ainsworth, Andriamaro, and Davydov 2011), allowing for fast numerical integration, could be an optimal solution. MoFEM is designed to provide tools that users can tackle such efficiency tradeoff and choose the optimal solution for a given problem.

The control of numerical errors is possible if we can estimate the error to drive the *hp-adaptivity* algorithm. This error estimator needs to be as much efficient as possible, and one possible solution is to use mixed finite element formulations, where error estimators become a part of the formulation, see e.g. (Carstensen 1997). However, the stability of such elements is an issue, which can be addressed by the appropriate use of a combination of H^1, **H-curl**, **H-div** and L^2 spaces. Mixed formulations have other advantages including reduced regularity of approximation, or the resulting sparse system of equations, that can be exploited by problem-tailored solution algorithms. In Fig **??** is shown p-adaptivity on hierarchical approximation base, with multi-grid solver applied to Scordelis-Lo perforated roof problem (Kaczmarczyk, Ullah, and Pearce 2016).

`MoFEM` is designed to provide all discussed above solutions for *hp-adaptivity*, enabling rapid implementation of the finite element method for solving complex multi-domain, multi-scale and multi-physics engineering problems. Moreover, it releases users from programming complexities related to the bookkeeping of degrees of freedom (DOFs), finite elements, matrix assembly, etc.

Design

Modern finite element software is an 'ecosystem' managing various complexities related to mesh and topology, sparse algebra and approximation, integration and dense tensor algebra at the integration point level. `MoFEM` has not developed and will not develop all these capabilities from scratch. Instead, `MoFEM` integrates

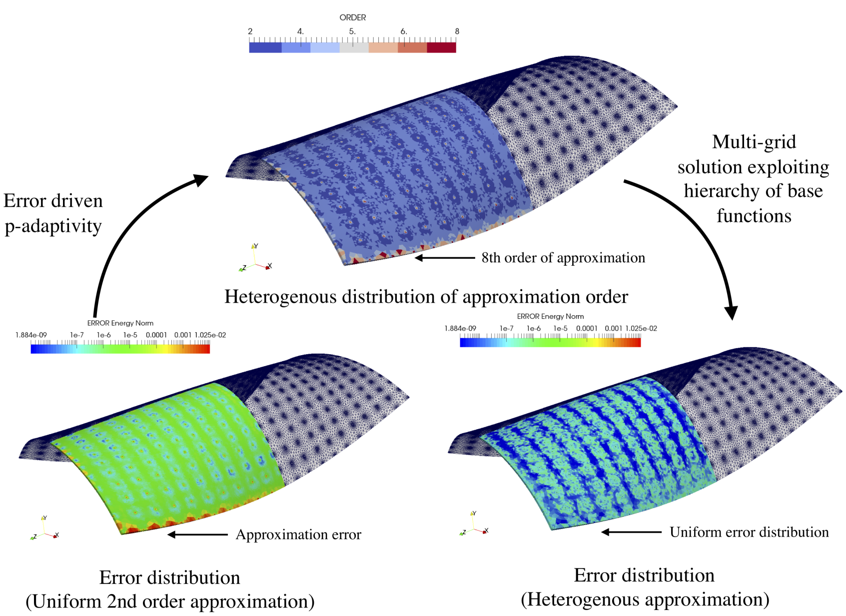

Figure 2: Example of p-adaptivity for hierarchical and heterogenous approximation with multi-grid solver applied for solid-shell element.

advanced scientific computing tools for sparse algebra from PETSc (Portable, Extensible Toolkit for Scientific Computation) (Balay et al. 2015), components for handling mesh and topology from MOAB (Mesh-Oriented Database) (Tautges et al. 2004) and data structures from Boost libraries ("Boost Web Page" 2019). An illustration of how these packages are utilised in `MoFEM` is shown in Fig. 3. Finally, `MoFEM` core library is developed to manage complexities directly related to the finite element method. Therefore, each part of this ecosystem has its own design objectives and appropriate programming tools from a spectrum of solutions can be selected. Resilience of `MoFEM` ecosystem is ensured since the underpinning components have sustainable fundings, dynamic and established groups of developers and significant user base. Fig. 4 shows different components that are employed in the ecosystem including popular pre- and post processing software.

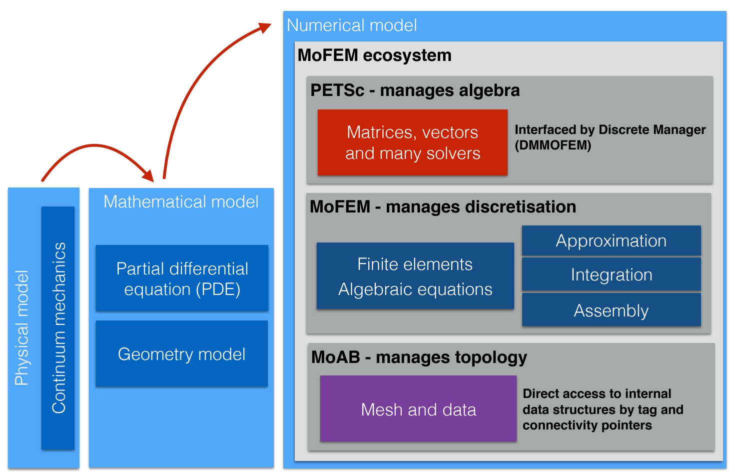

Figure 3: Basic design of `MoFEM` (Adopted from `MoFEM` webpage ("MoFEM Web Page" 2019)).

Traditional finite element codes are element-centric meaning the type of an element defines the approximation space and base. Therefore, they are not able to fully exploit the potential of emerging approximation methods. On the contrary, the design of data structures for approximation of field variables in `MoFEM` is independent of the specific finite element, e.g. Lagrangian, Nedelec, Rivart-Thomas, since finite element is constructed by a set of lower dimension entities on which the approximation fields are defined. Consequently, different approximation spaces (H^1, **H-curl**, **H-div**, L^2) can be arbitrarily mixed in a finite element to create new capabilities for solving complex problems efficiently.

`MoFEM` data structures enable easy enrichment of approximation fields and modifi-

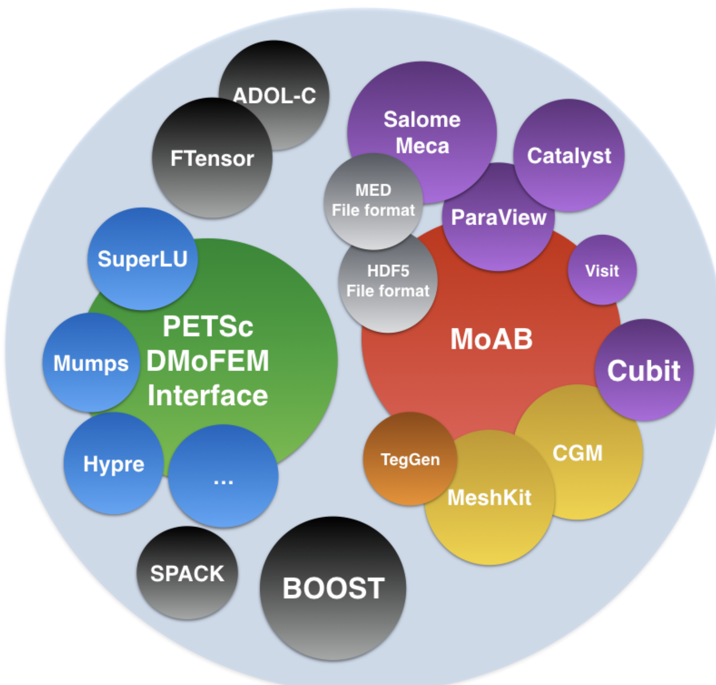

Figure 4: 'Ecosystem' of `MoFEM`. Adopted from MoFEM webpage ("MoFEM Web Page" 2019).

cation of base functions, for example, in case of resolving singularity at the crack front. Applying this technology, it is effortless to construct transition elements between domains with different problem formulation and physics, e.g. from two-field mixed formulation to single-field formulation, or elements with anisotropic approximation order, e.g. with arbitrary high order on surface and arbitrary low order through thickness of solid shells). This approach also sets the benchmark in terms of how finite element codes are implemented, introducing a concept of user-defined data operators acting on fields that are associated with entities (vertices, edges, faces and volumes) rather on the finite element directly. Such an approach simplifies code writing, testing and validation, making the code resilient to bugs.

Furthermore, `MoFEM` core library provides functionality for developing user modules where applications for particular problems can be implemented. This toolkit-like structure allows for independent development of modules with different repositories, owners and licences, being suitable for both open-access academic research and private industrial sensitive projects.

`MoFEM` is licensed under the [GNU Lesser General Public License] (https://www.gnu.org/licenses/lgpl.html), can be deployed and developed using the package manager Spack, see MoFEM installation instructions for more details.

Examples

`MoFEM` was initially created with the financial support of the Royal Academy of Engineering and EDF Energy to solve the problem of crack propagation in the nuclear graphite (Kaczmarczyk, Nezhad, and Pearce 2014),(Kaczmarczyk, Ullah, and Pearce 2017). Over time, the domain of applications expanded to include computational homogenisation (DURACOMP EPSRC Project EP/K026925/1), (Ullah et al. 2019, @zhou2017stochastic),(Ullah et al. 2017) bone remodelling and fracture (Kelvin Smith Scholarship), modelling of the gel rheology and acoustics problems. Moreover, `MoFEM` includes an extensive library of example applications such as soap film, solid shell, topology optimisation, phase field fracture, Navier-Stokes flow, cell traction microscopy, bone remodelling, configurational fracture, plasticity, mortar contact, magnetostatics and acoustic wave propagation as shown in Fig. 5.

Acknowledgements

`MoFEM` development has been supported by EDF Energy Nuclear Generation Ltd. (grant no. 4840360333), The Royal Academy of Engineering (grant no.

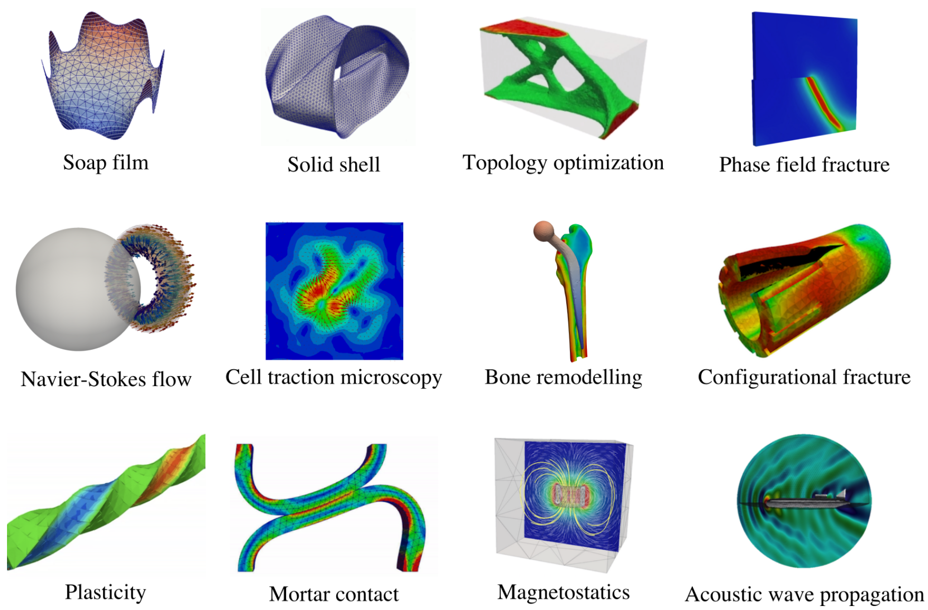

Figure 5: Examples of user modules implemented using `MoFEM`.

RCSRF1516\2\18), DURACOMP EPSRC Project (EP/K026925/1), and Kelvin Smith Scholarship programme at University of Glasgow.

References

Ainsworth, Mark, Gaelle Andriamaro, and Oleg Davydov. 2011. "Bernstein-Bézier Finite Elements of Arbitrary Order and Optimal Assembly Procedures." *SIAM Journal on Scientific Computing* 33 (6): 3087–3109.

Ainsworth, Mark, and Joe Coyle. 2003. "Hierarchic Finite Element Bases on Unstructured Tetrahedral Meshes." *International Journal for Numerical Methods in Engineering* 58 (14): 2103–30.

Babuška, I., and B. Q. Guo. 1992. "The H, P and H-P Version of the Finite Element Method; Basis Theory and Applications." *Advances in Engineering Software* 15: 159–74.

Balay, Satish, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, et al. 2015. "PETSc Web Page." http://www.mcs.anl.gov/petsc.

"Boost Web Page." 2019. https://www.boost.org.

Carstensen, Carsten. 1997. "A Posteriori Error Estimate for the Mixed Finite

Element Method." *Mathematics of Computation of the American Mathematical Society* 66 (218): 465–76.

Fuentes, Federico, Brendan Keith, Leszek Demkowicz, and Sriram Nagaraj. 2015. "Orientation Embedded High Order Shape Functions for the Exact Sequence Elements of All Shapes." *Computers & Mathematics with Applications* 70 (4): 353–458.

Kaczmarczyk, Lukasz, Zahur Ullah, and Chris Pearce. 2016. "Prism Solid-Shell with Heterogonous and Hierarchical Approximation Basis." *UKACM Cardiff, UK*. https://doi.org/10.5281/zenodo.789521.

Kaczmarczyk, Łukasz, Mohaddeseh Mousavi Nezhad, and Chris Pearce. 2014. "Three-Dimensional Brittle Fracture: Configurational-Force-Driven Crack Propagation." *International Journal for Numerical Methods in Engineering* 97 (7): 531–50.

Kaczmarczyk, Łukasz, Zahur Ullah, and Chris J Pearce. 2017. "Energy Consistent Framework for Continuously Evolving 3D Crack Propagation." *Computer Methods in Applied Mechanics and Engineering* 324: 54–73.

"MoFEM Web Page." 2019. http://mofem.eng.gla.ac.uk.

Tautges, T. J., R. Meyers, K. Merkley, C. Stimpson, and C. Ernst. 2004. "MOAB: A Mesh-Oriented Database." SAND2004-1592. Sandia National Laboratories.

Ullah, Zahur, SA Grammatikos, MC Evernden, CJ Pearce, and others. 2017. "Multi-Scale Computational Homogenisation to Predict the Long-Term Durability of Composite Structures." *Computers & Structures* 181: 21–31.

Ullah, Z, X-Y Zhou, L Kaczmarczyk, E Archer, A McIlhagger, and E Harkin-Jones. 2019. "A Unified Framework for the Multi-Scale Computational Homogenisation 3D-Textile Composites." *Composites Part B: Engineering*.

Zhou, X-Y, PD Gosling, Z Ullah, L Kaczmarczyk, and CJ Pearce. 2017. "Stochastic Multi-Scale Finite Element Based Reliability Analysis for Laminated Composite Structures." *Applied Mathematical Modelling* 45: 457–73.