

Coupling Materials for Fluid-Structure interactions in OpenFOAM

T. B. J. Di Giusto¹, A. J. Gil¹, C. Hean Lee², A. Huerta³ and M. Giacomini³

¹ Zienkiewicz Centre for Computational Engineering, Swansea University,
t.b.j.digiusto@swansea.ac.uk, a.j.gil@swansea.ac.uk

² Glasgow Computational Engineering Centre, University of Glasgow,
chunhean.lee@glasgow.ac.uk

³ Laboratori de Càlcul Numèric (LaCàN), Universitat Politècnica de Catalunya,
antonio.huerta@upc.edu, matteo.giacomini@upc.edu

ABSTRACT

Numerical simulations of nonlinear and multi-material large strain solid dynamics problems better account for highly complex applications in academia and industry. In particular, fluid-structure interactions problems (FSI) constitute a broad range of complex solid deformations, intricate fluid motion and a specific care about the numerical methods. In [1], Farhat uses Arbitrary Lagrangian Eulerian equations in an aeroelasticity context (interactions between a flow and an elastic body) where a particular care is put on the time integrator, and presents a method to exchange flux data across discrete interfaces and with a procedure to solve a coupled problem. The choice of relevant method and formulation determines the overall accuracy. In classical finite elements, strains and stresses converge slowly and numerical instabilities arise, especially around shocks, which can lead to locking. Besides, Lagrangian nodes follow the path of their associated particles which leads to a bad representation of large distortions, whereas Eulerian algorithms handle satisfyingly large strains but are more costly and more difficult to set up. The kinematical description of the continuum is crucial because it relates the numerical finite grid description with natural physical quantities. Arbitrary Lagrangian-Eulerian (ALE) aims to combine advantages of both kinematic descriptions to achieve higher robustness and accuracy. This approach allows the nodes to move independently based on a separate set of conservation laws, and is of great interest for partitioned FSI coupling. Indeed, explicit Fast Solid Dynamics may experience extreme deformation affecting dramatically the CFL number. A fast solid dynamics toolkit has been developed by Haider [2] to overcome numerical drawbacks of classical commercial frameworks and showed excellent results in parallel applications of complex problems. It is based on an in-house mixed formulation, uses a two-stage Runge-Kutta time integrator and an upwinding Riemann acoustic solver for reconstructing fluxes. The framework used for this work is based on this toolkit coded on the OpenFOAM platform, an open source collection of CFD solvers and utilities aiming to tackle a wide range of complex problems faced by researchers and industrials. It features parallel computing, pre and post-processing tools and original numerical methods such as dynamic mesh motion. The latter was implemented in OpenFOAM by Jasak and Tukovic [3] following the principles of the ALE method.

A vertex-centred FVM solid dynamics framework using a novel ALE mixed formulation will be coupled with OpenFOAM CFD solvers to perform complex fluid-structure interactions problems occurring in industry, The objectives are to overcome numerical limitations of a Lagrangian implementation (mesh entanglement [4]) and propose FSI with a pure OpenFOAM setup.

REFERENCES

- [1] Farhat, C., *CFD-Based Nonlinear Computational Aeroelasticity*, Encyclopedia of Computational Mechanics Second Edition, 2017, pp. 1-21

- [2] Haider, J. and Hean Lee, C. and Gil, A. J. and Bonet, J. and Huerta, A., *Development of a high performance parallel computing toolkit for fast solid dynamics in OpenFOAM*, 2018
- [3] Jasak, H. and Tukovic, Z., *Dynamic mesh handling in OpenFOAM applied to fluid-structure interaction simulations*, European Conference on Computational Fluid Dynamics, 2010, pp. 1-19
- [4] Donea, J. and Huerta, A. and Ponthot, J. and Rodriguez-Ferran, A., *Arbitrary Lagrangian-Eulerian Methods*, Encyclopedia of Computational Mechanics Second Edition, 2017, pp. 1-23