TOWARDS THE MODELING OF FLUID-STRUCTURE INTERACTIVE LOST CORE DEFORMATION IN HIGH-PRESSURE DIE CASTING

SEBASTIAN KOHLSTÄDT[1], MICHAEL VYNNYCKY[2], JAN JÄCKEL[3], LUDGER LOHRE[4]
[1]*KTH Royal Institute of Technology, Stockholm, Sweden, skoh@kth.se*
[2]*KTH Royal Institute of Technology, Stockholm, Sweden, michaelv@kth.se*
[3]*Volkswagen AG - Division of components manufacturing, Baunatal, Germany, jan.jaeckel@volkswagen.de*
[4]*Volkswagen AG - Division of components manufacturing, Baunatal, Germany, ludger.lohre@volkswagen.de*

Keywords: *compressible two-phase flow, fluid-structure interaction, high-pressure die casting, lost salt cores, solver development, experimental validation*

1 Introduction

High-pressure die casting (HPDC) is an important process for manufacturing high volume and low cost automotive components, such as automatic transmission housings and gear box components [1, 2]. Liquid metal, generally aluminium or magnesium, is injected through complex gate and runner systems and into the die at high speed, typically between 50 and 100 ms^{-1}, and under very high pressures up to 100 MPa. However, it has up to date proven difficult to employ lost salt cores within the process [3]. The basic idea of using salt cores is to create undercuts or hollow sections with them, which may then later act as cooling or oil-flow channels [4, 5, 6]. Given this process constraint in design freedom for the CAD-engineer, the idea of using salt as the material for lost cores has been put forward by machine manufacturers, as well as automotive companies [7, 8]. One way to determine whether this is indeed viable is to employ numerical simulation [9].

2 Model equations

2.1 Fluid side

We model the two-phase flow of molten metal and air in high pressure die casting by using the VOF method [10], wherein a transport equation for the VOF function, γ, of each phase is solved simultaneously with a single set of continuity and Navier-Stokes equations for the whole flow field; note also that γ, which is advected by the fluids, can thus be interpreted as the liquid fraction. Considering the molten melt and the air as Newtonian, compressible and immiscible fluids, the governing equations can be written as [11, 12]

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla p$$
$$+ \nabla \cdot \left\{ (\mu + \mu_{tur}) \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right) \right\} + \rho \mathbf{g} + \mathbf{F}_s, \tag{2}$$

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma \mathbf{U}) + \nabla \cdot (\gamma (1-\gamma) \mathbf{U}_r) =$$
$$- \frac{\gamma}{\rho_g} \left(\frac{\partial \rho_g}{\partial t} + \mathbf{U} \cdot \nabla \rho_g \right), \tag{3}$$

where t is the time, $\mathbf{U}$ the mean fluid velocity, p the pressure, $\mathbf{g}$ the gravity vector, $\mathbf{F}_s$ the volumetric representation of the surface tension force and $^{\mathrm{T}}$ denotes the transpose. In particular, $\mathbf{F}_s$ is modelled as a volumetric force by the Continuum Surface Force (CSF) method [13]. It is only active in the interfacial region and formulated as $\mathbf{F}_s = \sigma \kappa \nabla \gamma$, where σ is the interfacial tension and $\kappa = \nabla \cdot (\nabla \gamma / |\nabla \gamma|)$ is the curvature of the interface. The term $\mathbf{U}_r$ is a supplementary velocity field for compressing the phase-interface introduced by the solving scheme for the γ-field (MULES) [11, 14]. The material properties ρ and μ are the density and the dynamic viscosity, respectively, and are given by

$$\rho = \gamma \rho_l + (1-\gamma)\rho_g, \tag{4}$$
$$\mu = \gamma \mu_l + (1-\gamma)\mu_g, \tag{5}$$

where the subscripts g and l denote the gas and liquid phases, respectively. We take ρ_l, μ_g and μ_l to be constant, but use the approach to assume air as a barotropic fluid, i.e. its density changes linearly with pressure and the process to be isothermal and hence the equation of state for our model reads

$$\rho_g = \frac{p}{R_s T} = p\psi. \tag{6}$$

where R_s is the specific gas constant and T is the temperature. Since $\frac{1}{R_s T}$ is taken as constant, we introduce the constant compressibility factor ψ for the gaseous phase. Furthermore, μ_{tur} in equation (2) denotes the turbulent eddy viscosity, which will be calculated via the Menter k–ω-SST model [15]. The implementation of the Menter k–ω-SST model inside the OpenFOAM framework has shown to be robust and also in excellent agreement with experimental data [16].

2.2 Solid side

On the solid side only the stress equation is evaluated for calculating the displacement of the salt core in space. This process is physically governed by the following equation

$$\rho_s \frac{\partial^2 \mathbf{D}}{\partial t^2} - \nabla \cdot [(2\mu_s + \lambda_s)\nabla \mathbf{D}] = \nabla \cdot \mathbf{q}. \tag{7}$$

In equation (7) **D** represents the displacement vector, ρ_s denotes the density of the solid, λ_s and μ_s denote the solid's first and second Lamé parameters, respectively and **q** is the load per unit unit area. **q** consists of the pressure and viscous forces from the fluid side. There are no body forces taken into account for the selected approach.

3 Solver development and testing

Given the complexity of the physics involved and also the niche nature of the application, the C++ toolbox OpenFOAM [17, 18] was used to implement the model as it is freely available, rather suitable for being extended by the user and very well scalable for industrial application as extra cores in parallelisation do not require additional licenses. The solver that was used for the simulations in this paper does not yet exist in neither the OpenFOAM foundation release nor the foam-extend project. However, the foam-extend release offers a toolbox called the fsi library [19]. Inside the fsi library several solvers are available depending on the intensity of the coupling between the fluid and the solid. Here, as we expect to see rather large deformations, the solver *fsiFoam* was used. It follows a process chain depicted in figure 1. The names of the steps closely resemble the names given to the methods that are being called while the code is running. It may thus be easier to bridge between the code and the chart. One sees clearly that at first the displacement of the flow domain is evaluated, then the flow field according to the displacement and boundary conditions is evolved and then its eventual impact on the stress field in the solid is calculated. The blue loop indicates that this entire process is repeated until the specified residual is reached. After that the solver moves on to the next time step.

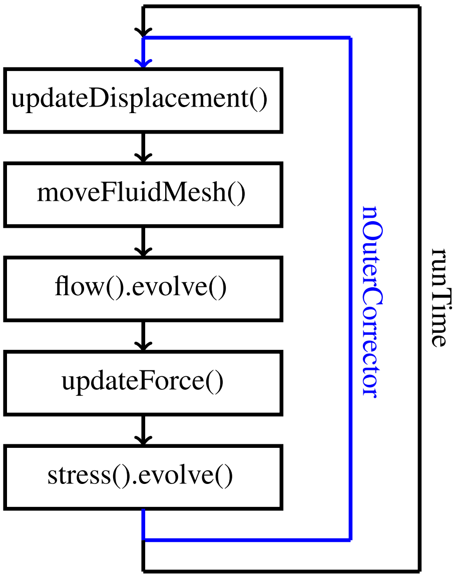
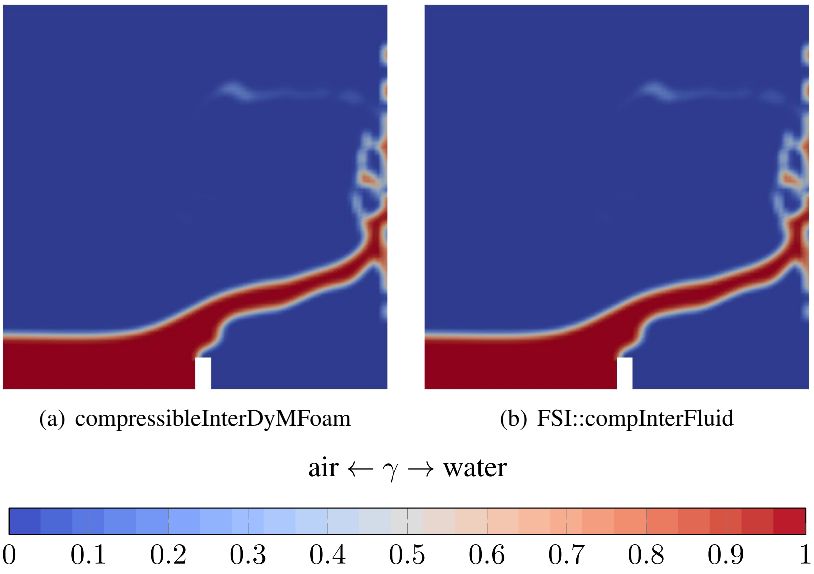


Figure 1: Solving process scheme of fsiFoam Solver

As outlined in the previous section, already the fluid side is a rather complex fluid flow problem. There was thus previously no solver class available that could solve such a problem within the framework of the fsi library. It however is available as a top level solver inside the normal OpenFOAM solver toolbox. The first step was therefore to implement the same methodology of the solver *compressibleInterDyMFoam* into a class that fits the requirements of an fsi fluid solver class. The standard pimple algorithm was for this purpose modified and an additional step at the beginning was added that solved the transport equation (equation 3) at the beginning of the process chain, i.e. before the momentum predictor.

This newly coded solver class inside the fsi library was benchmarked against the standard version of the solver *compressibleInterDyMFoam* with the simple standard *damBreak* tutorial case (figure 2) with the coupling between the fluid and the solid as well as the mesh motion switched off. Figure 2 shows the result of this benchmark study. As the reader can easily see the results for the phase field are entirely identical in the two pictures indicating that both the standard *compressibleInterDyMFoam* as well as the newly developed solver class *FSI::compInterFluid* produce the same flow field result.


(a) compressibleInterDyMFoam

(b) FSI::compInterFluid

Figure 2: γ-field comparsion of compressibleInterDyMFoam and FSI::compInterFluid at 0,4 s

In addition to this graphical comparison the solver has also been tested for mass conservation, volume conservation and conservation of γ in the process on simple geometries and benchmarked with analytical models.

4 Results and validation

The ultimate goal is to simulate the deformation of lost cores during the high-pressure die casting process. A simple structured mesh was constructed for this purpose that resembles the setup of a three-point bending test within a casting mould. Figure 3 shows a picture of the produced casting without the core in the upper part and also a picture of the deformation of the core in the lower part. The result is written out for the time step of 0.1 s shortly after the first drops of melt hit the core. Note that the geometry does not have an outlet. The die is initially filled with air and melt is flowing in at a constant velocity from the inlet.

It can be observed in figure 3 that the salt core undergoes a deformation in the range of 2.5 mm as soon as the melt hits the core and hence the momentum of the melt gets redirected due to the core blocking the initial channel flow. The core vibrates initially due to an observable spike in the force when the interface hits the core and is then displaced constantly by the bypassing flow, when the flow pattern around it becomes stable.

For the sake of validating the solver an experimental die was constructed and manufactured that resembles the geometry of the meshed body shown in figure 3. The die was then tested on a high-pressure die casting machine with the same process parameters that had previously been used for the simulation. One result of the so cast prototypes is shown in its cross-section in figure 4. One sees in this picture that analogously to the shape that we saw in figure 3 the core also gets symmetrically bent by the inflowing melt. The displacement is also in the order of magnitude of more than 2 mm and thus produces a comparable result as previously seen in the CFD-simulation (figure 3). Previously so far unpublished research by the authors also shows that the highest expectable impact on the core also occurs at the transition from the air phase to the melt phase at the core interface.

References

[1] B. Nogowizin, *Theorie und Praxis des Druckgusses*, 1. Edition ed. Berlin: Schiele und Schoen, 2010.

[2] E. Brunnhuber, *Praxis der Druckgussfertigung*. Berlin: Schiele und Schoen, 1991.

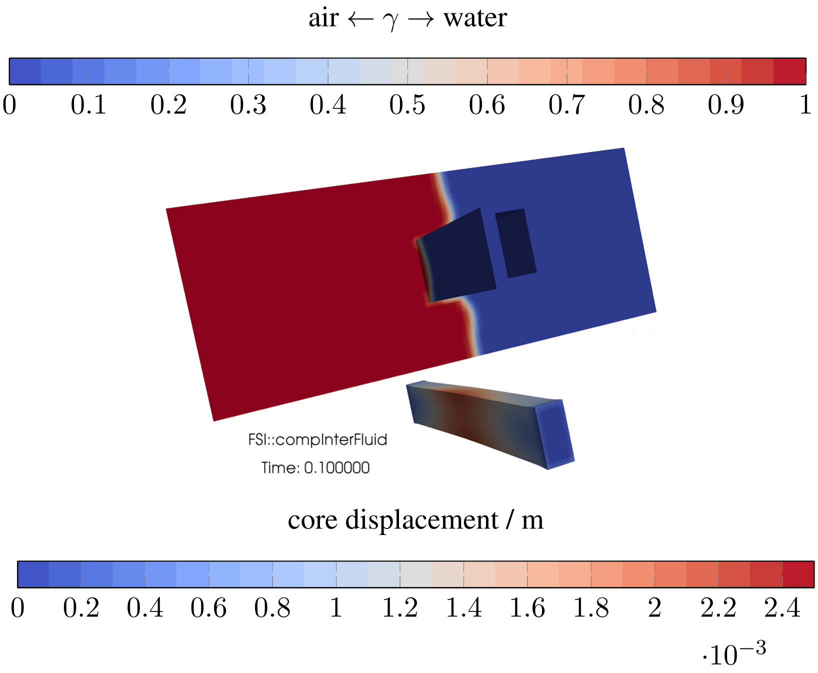


Figure 3: The deformation of the salt core as result of the CFD simulation.

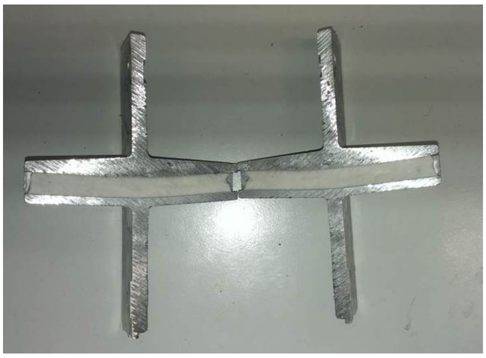


Figure 4: The deformation of the salt core in a casting experiment.

[3] B. Fuchs, H. Eibisch, and C. Körner, “Core viability simulation for salt core technology in high-pressure die casting,” *Int. J. Metalcasting*, vol. 7, pp. 39–45, 2013.

[4] E. Graf and G. Soell, “Vorrichtung zur Herstellung eines Druckgussbauteils mit einem Kern und einem Einlegeteil,” 2003, dE Patent DE2,001,145,876.

[5] E. D. I. Graf and G. D. I. Söll, “Verfahren zur Herstellung eines Zylinderkurbelgehäuses,” 2008, dE Patent DE200,710,023,060.

[6] S. Kohlstädt, U. Rabus, S. Goeke, and S. Kuckenburg, “Verfahren zur Herstellung eines metallischen Druckgussbauteils unter Verwendung eines Salzkerns mit integrierter Stützstruktur und hiermit hergestelltes Druckgussbauteil,” 2016, dE Patent DE201,410,221,359.

[7] P. Jelínek and E. Adámková, “Lost cores for high-pressure die casting,” *Archives of Foundry Engineering*, vol. 14, no. 2, pp. 101–104, 2014.

[8] T. Schneider, S. Kohlstädt, and U. Rabus, “Gehäuse mit Druckgussbauteil zur Anordnung eines elektrischen Fahrmotors in einem Kraftfahrzeug und Verfahren zur Herstellung eines Druckgussbauteils,” 2016, dE Patent DE201,410,221,358.

[9] B. Fuchs and C. Körner, “Mesh resolution consideration for the viability prediction of lost salt cores in the high pressure die casting process,” *Prog. Comp. Fluid Dyn.*, vol. 14, pp. 24–30, 2014.

[10] C. W. Hirt and B. D. Nichols, “Volume of fluid (VOF) method for the dynamics of free boundaries,” *J. Comp. Phys.*, vol. 39, pp. 201–225, 1981.

[11] P. M. Ferrer, D. Causon, L. Qian, C. Mingham, and Z. Ma, “A multi-region coupling scheme for compressible and incompressible flow solvers for two-phase flow in a numerical wave tank,” *Computers & Fluids*, vol. 125, pp. 116–129, 2016.

[12] R. Mayon, Z. Sabeur, M.-Y. Tan, and K. Djidjeli, “Free surface flow and wave impact at complex solid structures,” in *12th International Conference on Hydrodynamics , Egmond aan Zee, NL, 18 - 23 Sep 2016*, 2016, p. 10pp.

[13] J. Brackbill, D. B. Kothe, and C. Zemach, “A continuum method for modeling surface tension,” *Journal of computational physics*, vol. 100, no. 2, pp. 335–354, 1992.

[14] H. Rusche, “Computational fluid dynamics of dispersed two-phase flows at high phase fractions,” Ph.D. dissertation, Imperial College London (University of London), 2003.

[15] F. R. Menter, “2-equation eddy-viscosity turbulence models for engineering applications,” *AIAA J.*, vol. 32, pp. 1598–1605, 1994.

[16] E. Robertson, V. Choudhury, S. Bhushan, and D. Walters, “Validation of OpenFOAM numerical methods and turbulence models for incompressible bluff body flows,” *Computers & Fluids*, vol. 123, pp. 122–145, 2015.

[17] H. Jasak, A. Jemcov, and Z. Tukovic, “OpenFOAM: A C++ Library for Complex Physics Simulations,” in *International Workshop on Coupled Methods in Numerical Dynamics IUC*, 2007.

[18] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques,” *Computers in Physics*, vol. 12, no. 6, pp. 620–631, 1998.

[19] P. Cardiff, Ž. Tuković, H. Jasak, and A. Ivanković, “A block-coupled finite volume methodology for linear elasticity and unstructured meshes,” *Computers & structures*, vol. 175, pp. 100–122, 2016.