CSMA 2026
17ème Colloque National en Calcul des Structures
18-22 Mai 2026, Presqu'île de Giens (Var)

Non-intrusive level-set topology optimization for Fluid-Structure Interaction problems

G. Melo Gomes Pereira[1,2], N. Blal[1], A. Gravouil[1], J.-B. Minne[2], J.-F. Desseignes[2], F. Louis[2], D. Bardel[2]

[1] *INSA Lyon, CNRS, LaMCoS, UMR5259, Villeurbanne, {geovanne.melo-gomes-pereira, nawfal.blal, anthony.gravouil}@insa-lyon.fr*
[2] *Framatome, Lyon, {geovanne.melo-gomes-pereira, jean-baptiste.minne, jean-francois.desseignes, ferdinand.louis, didier.bardel,}@framatome.com*

Abstract — A Fluid-Structure Interaction topology optimization method is developed tailored for multiphysics applications in nuclear industry. The Hamilton-Jacobi equation is solved to implicitly track the fluid-structure interface, which is described by a level-set function. The interface velocity is obtained from the shape derivative of a constrained optimization problem considering a one-way partitioned, coupled Fluid-Structure Interaction. The algorithm uses the commercial software Simcenter™ STAR-CCM+™ and Cast3M to solve the governing equations and is validated using a benchmark case.
Keywords — Topology optimization, Fluid-Structure Interaction, level-set method

1 Introduction

Topology Optimization (TO) is a mathematical method used to determine the optimal distribution of material within a defined domain in order to minimize or maximize an objective function, subject to a given set of constraints, boundary conditions, and loads [4]. The development and maturation of Additive Manufacturing (AM) have contributed significantly to the exponential growth of TO applications, primarily because AM enables the fabrication of the resulting optimized shapes, which can often be extremely complex.

Since what is considered the seminal work of the domain [3], a variety of approaches have been developed and applied to different fields, such as solid ([2], [19]) and fluid mechanics ([6], [9]), and, to a lesser extent, multiphysics problems ([8], [11]). Regarding their code implementation, most studies use academic Finite Element Analysis (FEA) codes in MATLAB® ([17]) or open-source Python libraries ([10]), for example. However, it can be essential for stringent industries such as nuclear to rely on well-qualified tools, which often are commercial software.

With the aim of paving the way for applications within the nuclear industry, a multiphysics level-set TO method ([13], [18]) is developed, building on the work presented in [7], and coupling Python with Simcenter™ STAR-CCM+™ and Cast3M [5]. The proposed algorithm is non-intrusive, meaning the optimization operates externally to the solvers, treating them as black boxes without requiring access to or modification of their source code. It is validated using a well-known benchmark from the literature. This work represents the first step towards the development of a comprehensive multiphysics TO framework that couples commercial finite volume and finite element software with Python.

2 Multiphysics TO framework

The proposed method uses a Signed Distance Function (SDF) ϕ for the level-set method. A nodal scalar field is assigned to a structured mesh, where nodes located in the fluid subdomain (Ω_f) take positive values, nodes in the solid subdomain (Ω_s) take negative values, and the zero level set represents the fluid–structure (F-S) interface (Fig. 1). This formulation allows the geometry to be tracked dynamically and implicitly throughout the optimization. The evolution of the level set is governed by the Hamilton–Jacobi (H-J) equation (Eq. 1), where $\underline{V}$ is the interface velocity derived from the shape derivative [1] of a constrained optimization problem.

Figure 2 displays the design domain for an FSI TO problem. The fluid is Newtonian and incompressible. Its boundaries are defined as $\partial\Omega_f = \partial\Omega_f^D \cup \partial\Omega_f^N \cup \Gamma^{FSI}$, where $\partial\Omega_f^D$ denotes the Dirichlet boundary with prescribed velocity, and $\partial\Omega_f^N$ denotes the Neumann boundary with prescribed pressure. Γ^{FSI} is the F-S interface, corresponding to the exact intersection of the fluid and solid boundaries. The solid subdomain consists of a static, linear elastic, isotropic, homogeneous, and continuous structure, whose boundary is defined as $\partial\Omega_s = \partial\Omega_s^D \cup \partial\Omega_s^N \cup \Gamma^{FSI}$, where $\partial\Omega_s^D$ is the Dirichlet boundary with prescribed displacement, and $\partial\Omega_s^N$ is the Neumann boundary with prescribed load.

A one-way partitioned FSI coupling is considered. Under the assumption of small strains, the fluid influences the solid, but not vice versa. In practice, the fluid flow problem is solved first to obtain the stresses along the F-S interface, which are then applied as Neumann boundary conditions for the structural analysis.

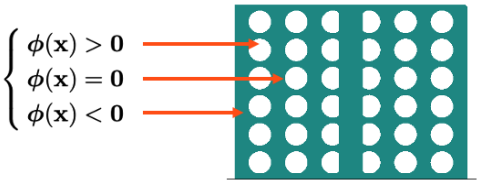


Figure 1: SDF scheme

$$\frac{\partial \underline{\underline{\phi}}}{\partial t} + \underline{V} \|\underline{\underline{\nabla \phi}}\| = \underline{0} \tag{1}$$

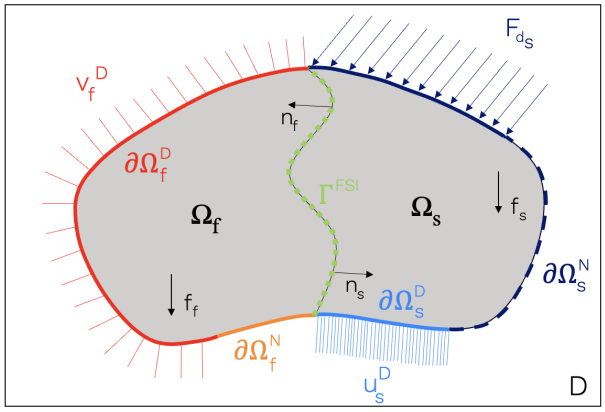


Figure 2: Design domain for FSI topology optimization

In this work, the TO problem given in Equation 2 consists of finding the optimal F-S interface Γ^{FSI} that minimizes J, a multi-objective function composed of the structural compliance and the structural mass. A weighting parameter α is introduced to control the relative importance of each contribution in the minimization of J, thereby allowing the algorithm to prioritize either compliance minimization or mass reduction.

$$\begin{aligned}
&\min_{\Gamma^{FSI}} \; J(\Omega) = \int_{\partial\Omega_s} \underline{u}_s \cdot \underline{F}_{d_s} \, dS + \alpha \int_{\Omega_s} d\Omega \\
&\text{s.t.} \quad
\begin{cases}
\underline{\mathrm{div}}(\underline{\underline{\sigma}}_f) + \underline{f}_f = \rho_f \left(\underline{\underline{\nabla v}}_f \cdot \underline{v}_f\right) & \forall x \in \Omega_f \\
\mathrm{div}(\underline{v}_f) = 0 & \forall x \in \Omega_f \\
\underline{\underline{\sigma}}_f \cdot \underline{n}_f = \underline{0} & \forall x \in \partial\Omega_f^N \\
\underline{v}_f = \underline{v}_f^D & \forall x \in \partial\Omega_f^D \\
\underline{v}_f = \underline{0} & \forall x \in \Gamma^{FSI}
\end{cases} \\
&\quad
\begin{cases}
\underline{\mathrm{div}}(\underline{\underline{\sigma}}_s) + \underline{f}_s = \underline{0} & \forall x \in \Omega_s \\
\underline{\underline{\sigma}}_s \cdot \underline{n}_s = \underline{F}_{d_s} & \forall x \in \partial\Omega_s^N \\
\underline{u}_s = \underline{u}_s^D & \forall x \in \partial\Omega_s^D
\end{cases} \\
&\quad
\left\{ \underline{\underline{\sigma}}_f \cdot \underline{n}_f = -\underline{\underline{\sigma}}_s \cdot \underline{n}_s = \underline{\lambda}_{FSI} \right. \qquad \forall x \in \Gamma^{FSI}
\end{aligned} \tag{2}$$

with

$$\underline{\underline{\sigma}}_f\left(\underline{v}_f, p\right) = -p\underline{\underline{I}} + 2\mu_f\underline{\underline{D}}$$

$$\underline{\underline{D}} = \frac{1}{2}\left(\left(\underline{\underline{\nabla}}\underline{v}_f\right) + \left(\underline{\underline{\nabla}}\underline{v}_f\right)^T\right) = \underline{\underline{\nabla}}^s\underline{v}_f$$

$$\underline{\underline{\sigma}}_s\left(\underline{u}_s\right) = \underline{\underline{\underline{\underline{C}}}} : \underline{\underline{\varepsilon}}_s\left(\underline{u}_s\right)$$

$$\underline{\underline{\varepsilon}}_s = \frac{1}{2}\left(\left(\underline{\underline{\nabla}}\underline{u}_s\right) + \left(\underline{\underline{\nabla}}\underline{u}_s\right)^T\right) = \underline{\underline{\nabla}}^s\underline{u}_s$$

where, regarding the fluid subdomain, $\underline{v}_f$ represents the velocity field, p the pressure, ρ_f the fluid density and μ_f the dynamic viscosity. The fluid-stress tensor is given by $\underline{\underline{\sigma}}_f$, where **D** is the strain-rate tensor, calculated as the symmetric part of the velocity gradient. In the solid subdomain, $\underline{u}_s$ denotes the displacement field, $\underline{\underline{\sigma}}_s$ is the solid stress tensor, $\underline{\underline{\underline{\underline{C}}}}$ is the fourth-order elasticity tensor for a linear elastic material, and $\underline{\underline{\varepsilon}}_s$ is the linearized strain tensor. $\underline{F}_{d_s}$ denotes the external traction applied on the Neumann boundary of the solid. The vectors $\underline{n}$ are the inward unit normals to the specified fluid or solid boundaries, while the terms $\underline{f}$ represent the body forces acting in their respective subdomains.

2.1 Augmented Lagrangian for an FSI TO

The augmented Lagrangian used to solve this problem is expressed as the sum of the multi-objective function J and the constraints multiplied by the corresponding Lagrange multipliers (Eq. 3). The formulation is simplified by employing a single Lagrange multiplier ($\underline{\lambda}_s$) for all equilibrium equations of the solid domain, as shown in [14]. Regarding the fluid domain, three different Lagrange multipliers are introduced: $\underline{\lambda}_f$ for the Navier-Stokes equation and the Neumann boundary condition, $\underline{\lambda}_D$ associated with Dirichlet-type conditions, and λ_I for the incompressibility equation.

$$\begin{aligned}
\mathcal{L}\left(\Omega, \underline{v}_f, p, \underline{u}_s, \underline{\lambda}_f, \lambda_I, \underline{\lambda}_D, \underline{\lambda}_s, \underline{\lambda}_{FSI}\right) &= \int_{\partial\Omega_s} \underline{u}_s \cdot \underline{F}_{d_s}\, dS + \alpha\int_{\Omega_s} d\Omega \\
&+ \int_{\Omega_f}\left(\underline{\mathrm{div}}(\underline{\underline{\sigma}}_f) + \underline{f}_f - \rho_f\left(\underline{\underline{\nabla}}\underline{v}_f\cdot\underline{v}_f\right)\right)\cdot\underline{\lambda}_f\, d\Omega + \int_{\Omega_f}\mathrm{div}\left(\underline{v}_f\right)\lambda_I\, d\Omega + \int_{\partial\Omega_f^N}\underline{\underline{\sigma}}_f\cdot\underline{n}_f\cdot\underline{\lambda}_f\, dS \\
&+ \int_{\partial\Omega_f^D}\left(\underline{v}_f - \underline{v}_f^D\right)\cdot\underline{\lambda}_D\, dS + \int_{\Gamma^{FSI}}\underline{v}_f\cdot\underline{\lambda}_D\, dS + \int_{\Omega_s}\left(\underline{\mathrm{div}}\left(\underline{\underline{\sigma}}_s\right) + \underline{f}_s\right)\cdot\underline{\lambda}_s\, d\Omega + \int_{\partial\Omega_s^D}\left(\underline{u}_s - \underline{u}_s^D\right)\cdot\underline{\lambda}_s\, dS \\
&+ \int_{\partial\Omega_s^N}\left(\underline{\underline{\sigma}}_s\cdot\underline{n}_s - \underline{F}_{d_s}\right)\cdot\underline{\lambda}_s\, dS + \int_{\Gamma^{FSI}}\left(\underline{\underline{\sigma}}_f\cdot\underline{n}_f - \underline{\lambda}_{FSI}\right)\cdot\underline{\lambda}_f\, dS + \int_{\Gamma^{FSI}}\left(\underline{\underline{\sigma}}_s\cdot\underline{n}_s + \underline{\lambda}_{FSI}\right)\cdot\underline{\lambda}_s\, dS \quad (3)
\end{aligned}$$

Using the alternate method as in [7], Ω is considered fixed initially. The problem consists then in finding $\left(\underline{v}_f, p, \underline{u}_s, \underline{\lambda}_f, \lambda_I, \underline{\lambda}_D, \underline{\lambda}_s, \underline{\lambda}_{FSI}\right) \in \left(\mathcal{U}_{v_f}, \mathcal{U}_p, \mathcal{U}_{u_s}, \mathcal{U}_{\lambda_f}, \mathcal{U}_{\lambda_I}, \mathcal{U}_{\lambda_D}, \mathcal{U}_{\lambda_s}, \mathcal{U}_{\lambda_{FSI}}\right)$:

$$\begin{cases}
\mathcal{U}_{v_f} = \{\underline{v}_f \in H^1\left(\Omega_f\right), \underline{v}_f = \underline{v}_f^D \quad \forall \mathrm{x} \in \partial\Omega_f^D, \quad \underline{v}_f = \underline{0} \quad \forall \mathrm{x} \in \Gamma^{FSI}\} \\
\mathcal{U}_p = \{p \in L^2\left(\Omega_f\right)/\mathbb{R}\} \\
\mathcal{U}_{u_s} = \{\underline{u}_s \in H^1\left(\Omega_s\right), \underline{u}_s = \underline{u}_s^D \quad \forall \mathrm{x} \in \partial\Omega_s^D\} \\
\mathcal{U}_{\lambda_f} = \{\underline{\lambda}_f \in H^1\left(\Omega_f\right)\} \\
\mathcal{U}_{\lambda_I} = \{\lambda_I \in L^2\left(\Omega_f\right)\} \\
\mathcal{U}_{\lambda_D} = \{\underline{\lambda}_D \in H^1\left(\Omega_f\right)\} \\
\mathcal{U}_{\lambda_s} = \{\underline{\lambda}_s \in H^1\left(\Omega_s\right), \underline{\lambda}_s = \underline{0} \quad \forall \mathrm{x} \in \partial\Omega_s^D\} \\
\mathcal{U}_{\lambda_{FSI}} = \{\underline{\lambda}_{FSI}|_{\Gamma^{FSI}} \in H_{00}^{1/2}\left(\Gamma^{FSI}\right)\}
\end{cases}$$

such that

$$\delta\mathcal{L} = \delta_{\underline{\lambda}_f}\mathcal{L} + \delta_{\lambda_I}\mathcal{L} + \delta_{\underline{\lambda}_D}\mathcal{L} + \delta_{\underline{\lambda}_s}\mathcal{L} + \delta_{\underline{\lambda}_{FSI}}\mathcal{L} + \delta_{\underline{v}_f}\mathcal{L} + \delta_p\mathcal{L} + \delta_{\underline{u}_s}\mathcal{L} = 0$$

$$\forall\left(\delta\underline{\lambda}_f, \delta\lambda_I, \delta\underline{\lambda}_D, \delta\underline{\lambda}_s, \delta\underline{\lambda}_{FSI}, \delta\underline{v}_f, \delta p, \delta\underline{u}_s\right) \in \left(\mathcal{U}_{\delta\lambda_f}, \mathcal{U}_{\delta\lambda_I}, \mathcal{U}_{\delta\lambda_D}, \mathcal{U}_{\delta\lambda_s}, \mathcal{U}_{\delta\lambda_{FSI}}, \mathcal{U}_{\delta v_f}, \mathcal{U}_{\delta p}, \mathcal{U}_{\delta u_s}\right)$$

$$
\begin{cases}
\mathcal{U}_{\delta\underline{\lambda}_f} = \{\delta\underline{\lambda}_f \in H^1(\Omega_f)\} \\
\mathcal{U}_{\delta\lambda_I} = \{\delta\lambda_I \in L^2(\Omega_f)\} \\
\mathcal{U}_{\delta\underline{\lambda}_D} = \{\delta\underline{\lambda}_D \in H^1(\Omega_f),\, \delta\underline{\lambda}_D = \underline{0} \quad \forall \mathrm{x} \in \partial\Omega_f^D \cup \Gamma^{FSI}\} \\
\mathcal{U}_{\delta\underline{\lambda}_s} = \{\delta\underline{\lambda}_s \in H^1(\Omega_s)\} \\
\mathcal{U}_{\delta\lambda_{FSI}} = \{\delta\lambda_{FSI} \in H_{00}^{-1/2}(\Gamma^{FSI})\} \\
\mathcal{U}_{\delta\underline{\mathrm{v}}_f} = \{\delta\underline{\mathrm{v}}_f \in H^1(\Omega_f),\, \delta\underline{\mathrm{v}}_f = \underline{0} \quad \forall \mathrm{x} \in \partial\Omega_f^D \cup \Gamma^{FSI}\} \\
\mathcal{U}_{\delta p} = \{\delta p \in L^2(\Omega_f)/\mathbb{R}\} \\
\mathcal{U}_{\delta\underline{\mathrm{u}}_s} = \{\delta\underline{\mathrm{u}}_s \in H^1(\Omega_s),\, \delta\underline{\mathrm{u}}_s = \underline{0} \quad \forall \mathrm{x} \in \partial\Omega_s^D\}
\end{cases}
$$

3 Stationary conditions and interface velocity

The derivative of $\mathcal{L}$ with respect to the Lagrange multipliers $\underline{\lambda}_f, \lambda_I$ and $\underline{\lambda}_s$ yields the governing equations, showing $\underline{\mathrm{v}}_f, p$ and $\underline{\mathrm{u}}_s$ must be solution of the proposed FSI problem (Eq. 4).

$$
\frac{\partial\mathcal{L}}{\partial\underline{\lambda}_f}\cdot\delta\underline{\lambda}_f + \frac{\partial\mathcal{L}}{\partial\lambda_I}\cdot\delta\lambda_I + \frac{\partial\mathcal{L}}{\partial\underline{\lambda}_s}\cdot\delta\underline{\lambda}_s = 0 \quad \forall\delta\underline{\lambda}_f, \delta\lambda_I, \delta\underline{\lambda}_s \in \mathcal{U}_{\delta\underline{\lambda}_f}, \mathcal{U}_{\delta\lambda_I}, \mathcal{U}_{\delta\underline{\lambda}_s}
$$

$$
\Leftrightarrow \begin{cases}
\underline{\mathrm{div}}\left(\underline{\underline{\sigma}}_f\right) + \underline{f}_f = \rho_f\left(\underline{\underline{\nabla}}\underline{\mathrm{v}}_f \cdot \underline{\mathrm{v}}_f\right) & \forall \mathrm{x} \in \Omega_f \\
\mathrm{div}\left(\underline{\mathrm{v}}_f\right) = 0 & \forall \mathrm{x} \in \Omega_f \\
\underline{\underline{\sigma}}_f \cdot \underline{\mathrm{n}}_f = 0 & \forall \mathrm{x} \in \partial\Omega_f^N \\
\underline{\underline{\sigma}}_f \cdot \underline{\mathrm{n}}_f = \underline{\lambda}_{FSI} = \underline{\underline{\sigma}}_s \cdot \underline{\mathrm{n}}_s & \forall \mathrm{x} \in \Gamma^{FSI} \\
\underline{\mathrm{div}}\left(\underline{\underline{\sigma}}_s\right) + \underline{f}_s = \underline{0} & \forall \mathrm{x} \in \Omega_s
\end{cases} \tag{4}
$$

The derivative with respect to λ_{FSI}, enforces the consistency of the kinematic and dynamic coupling at the F-S interface, and thus providing a relation between $\underline{\lambda}_f$ and $\underline{\lambda}_s$ on Γ^{FSI} (Eq. 5).

$$
\frac{\partial\mathcal{L}}{\partial\underline{\lambda}_{FSI}}\cdot\delta\underline{\lambda}_{FSI} = 0 \quad \forall\delta\underline{\lambda}_{FSI} \in \mathcal{U}_{\delta\underline{\lambda}_{FSI}}
$$

$$
\Leftrightarrow \left\{\underline{\lambda}_f = \underline{\lambda}_s \quad \forall \mathrm{x} \in \Gamma^{FSI} \right. \tag{5}
$$

Taking the derivative of the Lagrangian with respect to the primal variables $\underline{\mathrm{v}}_f$ and p leads to the Euler-Lagrange equations whose stationary conditions define $\underline{\lambda}_f$ and $\underline{\lambda}_I$. These are obtained by solving the linear system given in Equation 6. It is worth mentioning that this system is consistent with the results for adjoint-based approaches as in [8].

$$
\frac{\partial\mathcal{L}}{\partial\underline{\mathrm{v}}_f}\cdot\delta\underline{\mathrm{v}}_f + \frac{\partial\mathcal{L}}{\partial p}\cdot\delta p = 0 \quad \forall\delta\underline{\mathrm{v}}_f, \delta p \in \mathcal{U}_{\delta\underline{\mathrm{v}}_f}, \mathcal{U}_{\delta p}
$$

$$
\Leftrightarrow \begin{cases}
-\underline{\mathrm{div}}\left(\underline{\underline{\sigma}}_f\left(\underline{\lambda}_f, \lambda_I\right)\right) + \rho_f\left(\underline{\underline{\nabla}}\underline{\mathrm{v}}_f \cdot \underline{\lambda}_f - \underline{\underline{\nabla}}\underline{\lambda}_f \cdot \underline{\mathrm{v}}_f\right) = \underline{0} & \forall \mathrm{x} \in \Omega_f \\
\mathrm{div}\left(\underline{\lambda}_f\right) = 0 & \forall \mathrm{x} \in \Omega_f \\
\underline{\underline{\sigma}}_f\left(\underline{\lambda}_f, \lambda_I\right) \cdot \underline{\mathrm{n}}_f + \rho_f\left(\underline{\mathrm{v}}_f \cdot \underline{\mathrm{n}}_f\right)\underline{\lambda}_f = \underline{0} & \forall \mathrm{x} \in \partial\Omega_f^N
\end{cases} \tag{6}
$$

The Euler-Lagrange equation with respect to $\underline{\mathrm{u}}_s$ results in the stationary condition of $\underline{\lambda}_s$ given by Equation 7. Finally, the Lagrange multiplier $\underline{\lambda}_D$, related to the fluid Dirichlet and the kinematic coupling condition on the F-S interface, is only introduced to guarantee the well-posedness of the problem and does not require explicit computation.

$$
\frac{\partial\mathcal{L}}{\partial\underline{\mathrm{u}}_s}\cdot\delta\underline{\mathrm{u}}_s = 0 \quad \forall\delta\underline{\mathrm{u}}_s \in \mathcal{U}_{\delta\underline{\mathrm{u}}_s}
$$

$$\Leftrightarrow \begin{cases} \underline{\mathrm{div}}\left(\underline{\underline{\sigma}}_s\left(\underline{\lambda}_s\right)\right) = \underline{0} & \forall \mathrm{x} \in \Omega_s \\ \underline{\underline{\sigma}}_s\left(\underline{\lambda}_s\right) \cdot \underline{\mathrm{n}}_s = -\underline{F}_{d_s} & \forall \mathrm{x} \in \partial\Omega_s^N \\ \underline{\underline{\sigma}}_s\left(\underline{\lambda}_s\right) \cdot \underline{\mathrm{n}}_s = \underline{0} & \forall \mathrm{x} \in \Gamma^{FSI} \end{cases} \tag{7}$$

Thus, the shape derivative method can be used to define the normal velocity for the H-J equation that drives the decrease of the chosen multi-objective function [16]. In this case, the choice of V that ensures a descent direction for J is $V = \underline{\underline{\sigma}}_s : \underline{\underline{\varepsilon}}_s - \alpha$. Further details on its derivation can be found in [7].

4 Algorithm overview

The flowchart of the optimization algorithm is shown in Fig. 3. Python was chosen as the main programming language to implement this method, as it handles multiple tasks: initializing the computational domain and the SDF structured mesh, initializing the level-set and density scalar fields, interpolating the computed stress on the F-S interface, computing the objective function, sensitivities and interface velocity, solving the H-J equation, reinitializing ϕ to a SDF and update the density scalar field. Furthermore, it also serves as a bridge between Simcenter^{TM} STAR-CCM+^{TM} and Cast3M.

Simcenter^{TM} STAR-CCM+^{TM} is used to solve the fluid flow via the Finite Volume Method (FVM) through a Java macro, while the Cast3M solves the structural problem using the Finite Element Method (FEM) through a Gibiane code. The optimization process iterates until a stopping criterion is met, which is defined either by reaching a maximum number of iterations or by the stabilization of the objective function.

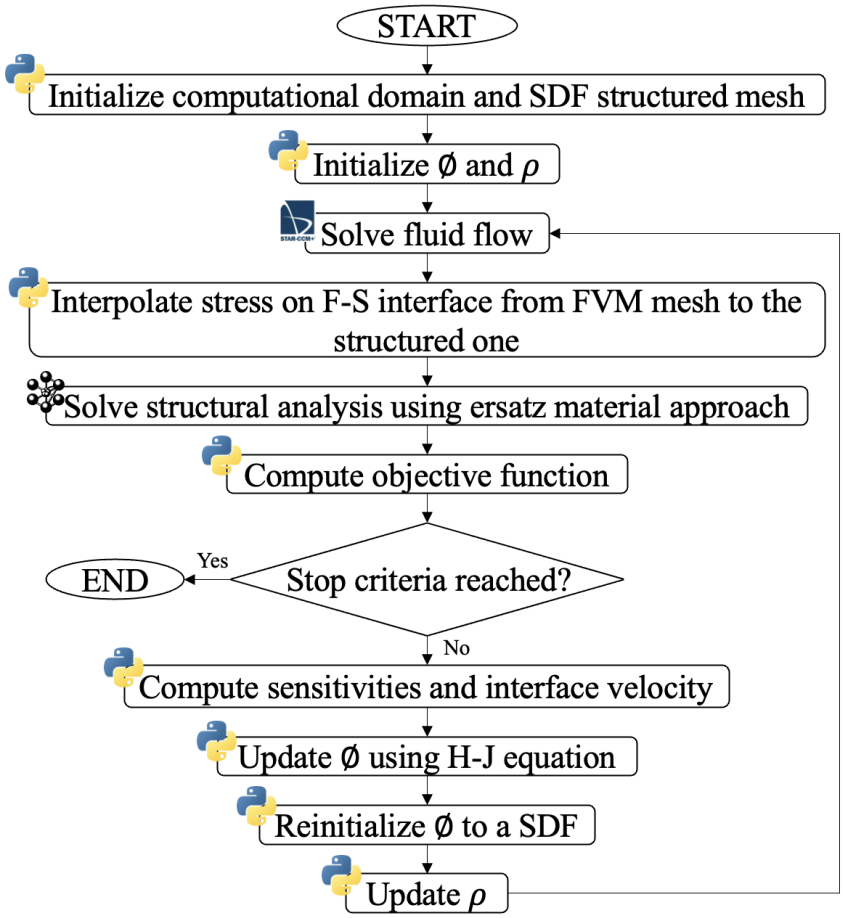


Figure 3: Algorithm flowchart

For now, the ersatz material approach [3] is used in the FEM to penalize the fluid regions. An improvement for the following of this work is to solve the structural problem exclusively on the solid subdomain by remeshing it at each iteration as the geometry evolves.

5 Study case

To validate the implementation of the algorithm, a well-known FSI TO benchmark was selected: a beam inside a pipe with fluid flow, as illustrated in Figure 4. This setup is similar to cases studied in [8], [12] and [15], with the yellow region representing a non-designable area. The pipe has dimensions 2.0×0.5 m, and the structure measures 0.4×0.3 m. Fluid and material properties are the same as in [8].

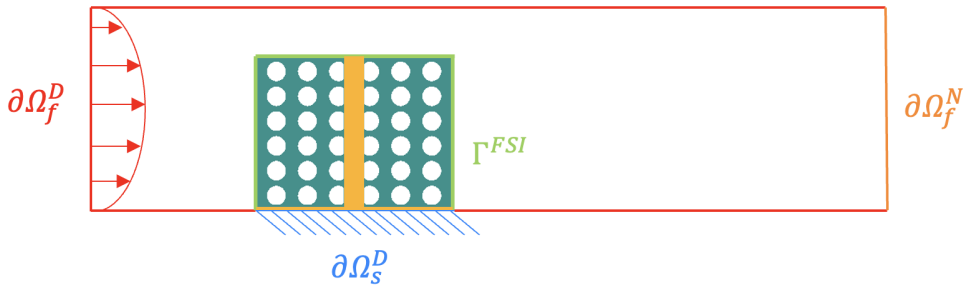
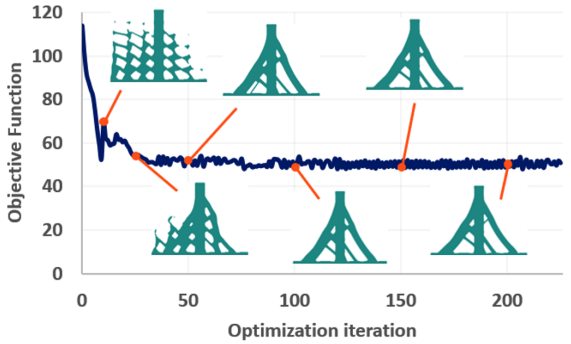
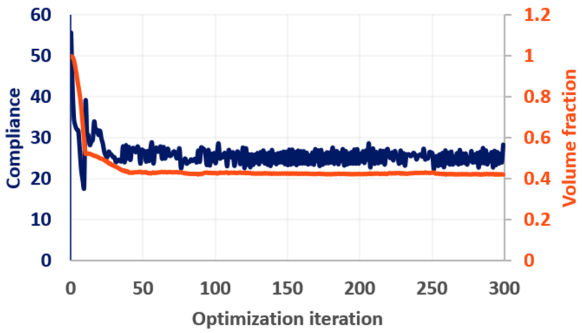


Figure 4: FSI TO problem study case

It is reminded that the goal is to minimize both structure compliance and mass, as described in Section 2. The optimization was run for 300 iterations, and the results are presented in Figure 5. However, the geometry has not changed significantly from iteration 150 (Fig 5a). Additionally, the objective function was successfully minimized. Figure 5b illustrates how the algorithm managed the trade-off between compliance minimization and mass reduction for $\alpha = 750$.



(a) Objective function evolution



(b) Compliance and volume fraction

Figure 5: Optimization result

Comparing the final geometry obtained (Fig. 6a) with results from the literature (Figs. 6b, 6c, and 6d), similarities can be observed, although the solutions are not identical. This is expected, as TO problems do not generally yield a unique solution, and the results strongly depend on modeling choices and parameter settings. In addition, the referenced works do not use the exact same domain, mesh, or geometric representation. Several also rely on different optimization approaches, such as Solid Isotropic Material with Penalization (SIMP) or Bidirectional Evolutionary Structural Optimization (BESO), instead of the level-set method.

(a) Proposed algorithm

(b) Feppon et al., 2019 [8]

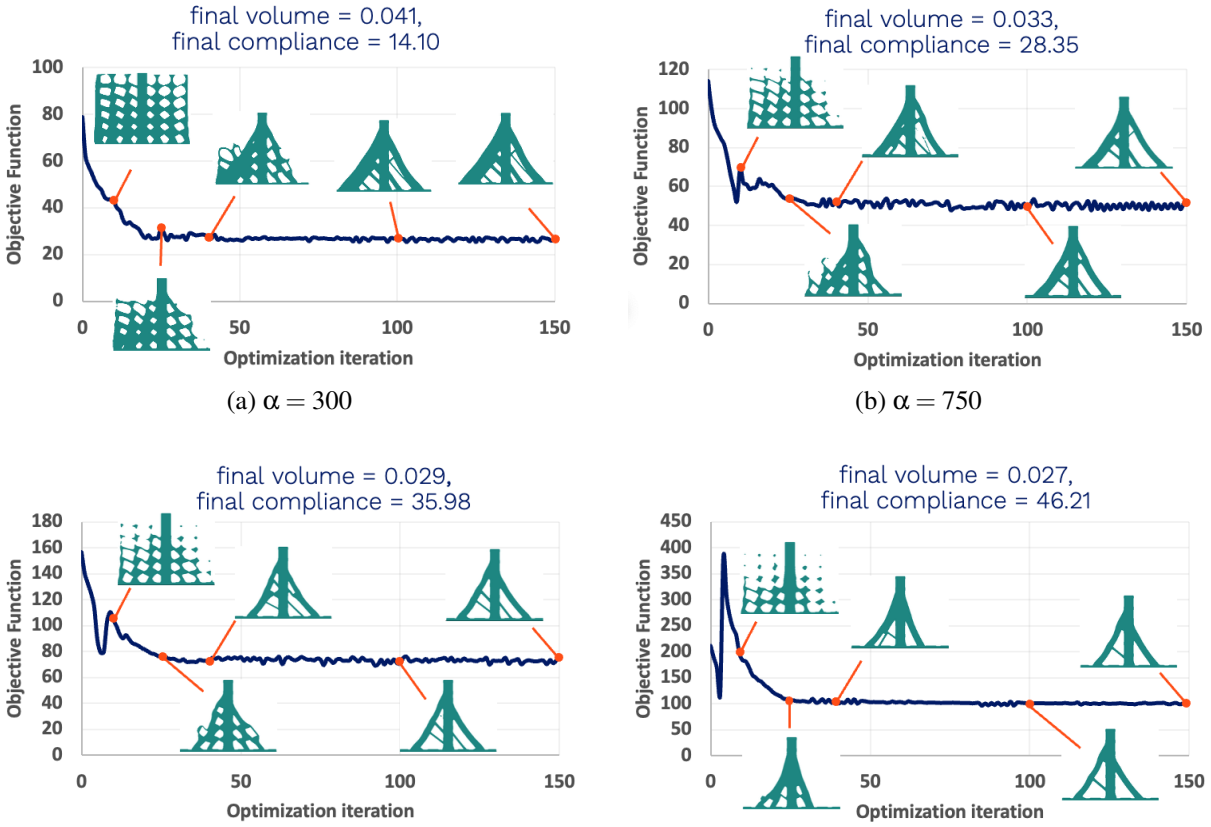
(c) Lundgaard et al., 2018 [12]

(d) Picelli et al., 2017 [15]

Figure 6: Comparison of the proposed algorithm with 3 similar literature cases.

Moreover, most studies impose a target mass as an inequality constraint, whereas in the present work mass reduction is embedded directly into the multi-objective function. Therefore, the comparison aims mainly at verifying whether the optimized topology exhibits similar qualitative features. The key validation outcome, however, is the correct minimization of the proposed objective function, which was successfully achieved.

Finally, a sensitivity analysis was carried out on the parameter α. Three additional optimizations were performed for $\alpha = 300$, 1300, and 2000 (Fig. 7). Since this weight is applied to the mass term of J, a higher value of α places greater emphasis on mass reduction relative to compliance minimization. Therefore, an increase in α is expected to produce a lower final volume at the expense of a higher compliance. This trend was indeed observed, with the optimized structures becoming progressively lighter as α increases.



(a) $\alpha = 300$ (b) $\alpha = 750$

(c) $\alpha = 1300$ (d) $\alpha = 2000$

Figure 7: α sensibility study

6 Conclusion and perspectives

In summary, a non-intrusive level-set-based TO algorithm for FSI problems, coupling commercial software and Python is proposed. The geometry is represented through a SDF, and the constrained optimization problem is solved using an augmented Lagrangian approach, with the computation of the shape derivative and the update of the level-set field via the H-J equation. The method was validated using the academic case of an elastic beam clamped at the bottom inside a pipe.

A sensitivity study on the mass weighting parameter α was performed to study the expected trade-off between compliance and mass: higher values of α led to lighter structures at the expense of stiffness. Future developments will aim to improve the structural model by removing the ersatz material approach to solving the structural problem only on the solid domain. Furthermore, a more general 3D case, in which properties inside both the fluid and solid subdomains are minimized simultaneously, is currently under development. Finally, since applications in the nuclear industry typically involve turbulent flows, future work will also focus on extending the framework to handle TO under turbulent regimes.

References

[1] G. Allaire, F. Jouve, A-M. Toader. *Structural optimization using sensitivity analysis and a level-set method*, Journal of Computational Physics, 194(1):363–393, 2004.

[2] G. C. Andrade, S. A. Santos. *A level-set-based topology optimization strategy using radial basis functions and a Hilbertian velocity extension*, Applied Mathematical Modelling, 111:108-125, 2022.

[3] M. P. Bendsøe, N. Kikuchi. *Generating optimal topologies in structural design using a homogenization method*, Computer Methods in Applied Mechanics and Engineering, 71(2):197–224, 1988.

[4] M. P. Bendsøe, O. Sigmund. *Topology Optimization Theory, Methods, and Applications*, Springer, 2004.

[5] Cast3M. *Cast3M – Finite Element Code*, Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA), France, 2024. Available at: `https://www-cast3m.cea.fr`.

[6] C. Dapogny, P. Frey, F. Omnès, Y. Privat. *Geometrical shape optimization in fluid mechanics using FreeFem++*, Structural and Multidisciplinary Optimization, 58(6):2761-2788, 2018.

[7] T. Djourachkovitch, N. Blal, N. Hamila, A. Gravouil. *Multiscale topology optimization of 3D structures: A micro-architectured materials database assisted strategy*, Computers & Structures, 255:106574. 2021.

[8] F. Feppon, G. Allaire, F. Bordeu, J. Cortial, C. Dapogny. *Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework*, SeMA Journal, 76(3):413-458, 2019.

[9] S. Kubo, A. Koguchi, K. Yaji, T. Yamada, K. Izui, S. Nishiwaki. *Level set-based topology optimization for two dimensional turbulent flow using an immersed boundary method*, Journal of Computational Physics, 446, 2021.

[10] A. Laurain. *A level set-based structural optimization code using FEniCS*, Structural and Multidisciplinary Optimization, 58(3):1311–1334, 2018.

[11] H. Li, T. Kondoh, P. Jolivet, K. Furuta, T. Yamada, B. Zhu, K. Izui, S. Nishiwaki. *Three-dimensional topology optimization of a fluid–structure system using body-fitted mesh adaption based on the level-set method*, Applied Mathematical Modelling, 101:276-308, 2022.

[12] C. Lundgaard, J. Alexandersen, M. Zhou, C. S. Andersen, O. Sigmund. *Revisiting density-based topology optimization for fluid-structure interaction problems*, Structural and Multidisciplinary Optimization, 2018.

[13] S. Osher, R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces, volume 153 of Applied Mathematical Sciences*, Springer New York, 2003.

[14] D. K. N. Pham *Mécanique computationnelle pilotée par des données - Extension pour l'élasto-plasticité et application thermomécanique*,INSA de Lyon, 2023.

[15] R. Picelli, W. M. Vicente, R. Pavanello. *Evolutionary topology optimization for structural compliance minimization considering design-dependent FSI loads*, Finite Elements in Analysis and Design, 135:44-55, 2017.

[16] J. Sokolowski, J-P. Zolesio. *Introduction to shape optimization*, Springer, 1992.

[17] Y. Wang, Z. Kang. *MATLAB implementations of velocity field level set method for topology optimization: an 80-line code for 2D and a 100-line code for 3D problems*, Structural and Multidisciplinary Optimization, 64(6):4325–4342, 2021.

[18] M. Y. Wang, X. Wang, D. Guo. *A level set method for structural topology optimization*, Computer Methods in Applied Mechanics and Engineering, 192(1-2):227–246, 2003.

[19] Z. Zhuang, F. Xu, J. Ye, W. Tong, Z. Chen, Y. Weng. *A 262-line Matlab code for the level set topology optimization based on the estimated gradient field in the body-fitted mesh*, Structural and Multidisciplinary Optimization, 67(9):171, 2024.