



**HAL**  
open science

# Machine-Learning Enhanced Predictors for Accelerated Convergence of Partitioned Fluid-Structure Interaction Simulations

Azzeddine Tiba, Thibault Dairay, Florian de Vuyst, Iraj Mortazavi,  
Juan-Pedro Berro Ramirez

► **To cite this version:**

Azzeddine Tiba, Thibault Dairay, Florian de Vuyst, Iraj Mortazavi, Juan-Pedro Berro Ramirez. Machine-Learning Enhanced Predictors for Accelerated Convergence of Partitioned Fluid-Structure Interaction Simulations. 2024. hal-04576920

**HAL Id: hal-04576920**

**<https://hal.science/hal-04576920v1>**

Preprint submitted on 15 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

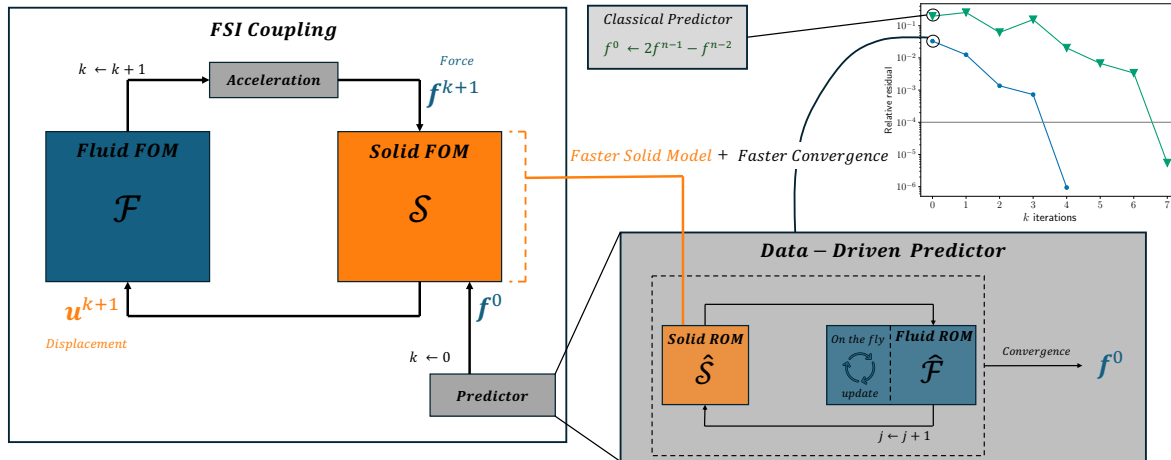


Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

# Graphical Abstract

## Machine-Learning Enhanced Predictors for Accelerated Convergence of Partitioned Fluid-Structure Interaction Simulations

TIBA Azzeddine, DAIRAY Thibault, DE VUYST Florian, MORTAZAVI Iraj, BERRO RAMIREZ Juan-Pedro



## Highlights

### **Machine-Learning Enhanced Predictors for Accelerated Convergence of Partitioned Fluid-Structure Interaction Simulations**

TIBA Azzeddine, DAIRAY Thibault, DE VUYST Florian, MORTAZAVI Iraj, BERRO RAMIREZ Juan-Pedro

- Novel predictor to accelerate convergence of fluid-structure interactions problems.
- Coupled solution of solid and fluid reduced models is used as the next initial guess.
- Reduced models take the form of encoder-regressor-decoder data-driven models.
- Online adaptation of the reduced models is used for more robust extrapolation.
- Faster convergence and speedups up to 3.2 versus classical-predictor based coupling.

# Machine-Learning Enhanced Predictors for Accelerated Convergence of Partitioned Fluid-Structure Interaction Simulations

TIBA Azzeddine<sup>a</sup>, DAIRAY Thibault<sup>b,c</sup>, DE VUYST Florian<sup>d</sup>, MORTAZAVI Iraj<sup>a</sup>, BERRO RAMIREZ Juan-Pedro<sup>e</sup>

<sup>a</sup>*M2N, CNAM, 2, Rue Conté, Paris, 75003, France*

<sup>b</sup>*Manufacture Française des Pneumatiques Michelin, Place des Carmes-Dechaux, Clermont-Ferrand, 63000, France*

<sup>c</sup>*Centre Borelli, CNRS, Université Paris Saclay, ENS Paris Saclay, 4, Avenue des Sciences, Gif-sur-Yvette, 91190, France*

<sup>d</sup>*Université de Technologie de Compiègne, CNRS, Laboratory of Biomechanics and Bioengineering, Rue du docteur Schweitzer, Compiègne, 60203, France*

<sup>e</sup>*Altair Engineering France, 5, Rue de la Renaissance, Antony, 92160, France*

---

## Abstract

Stable partitioned techniques for simulating unsteady fluid-structure interaction (FSI) are known to be computationally expensive when high added-mass is involved. Multiple coupling strategies have been developed to accelerate these simulations, but often use predictors in the form of simple finite-difference extrapolations. In this work, we propose a non-intrusive data-driven predictor that couples reduced-order models of both the solid and fluid subproblems, providing an initial guess for the nonlinear problem of the next time step calculation. Each reduced order model is composed of a nonlinear encoder-regressor-decoder architecture and is equipped with an adaptive update strategy that adds robustness for extrapolation. In doing so, the proposed methodology leverages physics-based insights from high-fidelity solvers, thus establishing a physics-aware machine learning predictor. Using three strongly coupled FSI examples, this study demonstrates the improved convergence obtained with the new predictor and the overall computational speedup realized compared to classical approaches.

*Keywords:* Physically relevant reduced order model, Fluid-structure interaction, Partitioned approach, ROM-FOM coupling, Data-driven model, Fixed-point acceleration.

---

## 1. Introduction

Engineering applications involving Fluid-Structure Interaction (FSI) phenomena are numerous, and are present in various fields. This includes aeroelasticity, biomechanics, microfluidics, hydrodynamics and many more. Modeling these FSI problems is a challenging task due to the usually high complexity of the coupling between the solid system and the fluid system.

Physics-based numerical simulations are considered as one of the leading options of modeling FSI problems, benefiting from years of advances in computational mechanics and from the increase of accessible computing power. These computations often have the goal of modeling the complicated coupling between the kinematics of a solid body and a fluid flow along with the mechanical loads associated with it. This results in very complicated problems, with nonlinearities present in both the fluid and the solid systems, in addition to the nonlinearities of the coupling itself. Furthermore, highly complex dynamics can be present due for example to the turbulent nature of the flow and/or to the interaction of the dynamical effects of the two systems involved.

Different discretization methods exist for handling FSI problems, especially from the fluid side, to deal with moving bodies: we mention for example immersed boundary methods [1], smoothed particle hydrodynamics (SPH) [2], particle finite elements (PFEM) [3] and finite elements and finite volumes with the arbitrary Eulerian-Lagrangian formulation (ALE) [4] which is the method used in the present study.

The approaches to solve FSI problems numerically can be classified commonly into two main classes: monolithic and partitioned methods. In the former, the governing equations of the fluid and the solid behavior are solved simultaneously, with the coupling conditions (equality of the forces, displacements and velocities at the interface) respected exactly [5, 6, 7, 8]. However, computing the coupling terms (e.g the cross-derivatives in the Jacobian matrices) in the context of global Newton iterations can pose some serious computational problems. In addition to the expensive cost, it is very difficult to implement the monolithic schemes when different fluid and solid solvers are considered, especially if different discretization strategies are adopted for the two subproblems. This also becomes a significant disadvantage from a practical standpoint, in e.g an industrial context where there is a need for a non-intrusive combination of existing fluid and solid solvers.

In the partitioned approach, each subproblem is solved separately, making it possible to use specialized solvers for the solid and the fluid, even when using different nonconforming grids on both sides and/or different discretization strategies on each domain (e.g. finite elements on the solid side vs SPH [9] or PFEM [10] on the fluid side). This constitutes one of the main advantages of this approach, explaining its popularity. The partitioned coupling is achieved thanks to the communication of relevant quantities between the solvers at the interface, namely the displacement, velocity and stress fields at the interface, which are then used to enforce the adequate boundary conditions on each problem. This is the main idea behind the *Dirichlet-Neumann* formulation for example, where the displacements and velocities stemming from the solid computations are enforced at the fluid boundary, and the forces resulting from the fluid stresses are imposed as a load on the solid domain. Unfortunately, an inherent instability appears when dealing with partitioned approaches: Due to the time delay that exists between the instants of solving the solid and the fluid problems, ensuring stability of the coupling conditions is not straightforward. In problems where the coupling is not very strong, i.e when the effect of one sub-problem on the other is much smaller than the effect of the other direction, coupling schemes can be made with one solver call per time step. This is commonly called *staggered schemes* or *loosely coupled schemes*. Extensive development of staggered schemes was done in early works [11, 12, 13, 14, 15, 16], where strategies were introduced with improved synchronization of the two subsystems [13], improved order of accuracy [11, 16], and improved stability [12, 14], which generally involves clever choices of the time integrator in each solver, and a crucial choice of the good predictors of force or displacement at the beginning of each time step. Although the performance of such strategies was well demonstrated in aeroelastic applications with compressible flows, loosely coupled scheme perform much worse in problems with stronger coupling and incompressible flows. This is due to the added mass instability, a phenomenon related to the effect of the displaced (due to the solid displacement) fluid mass, on the solid itself. It was shown in [17, 18, 19] that this effect is more significant when the ratio of the added fluid mass on the solid mass increases, and that it also depends on the geometry of the two subdomains. It was also shown that, when the added mass is large, the partitioned schemes - even with full subiterations - can, at worst, fail to converge quickly, and at best, be conditionally stable, irrespective of the time step. In fact, smaller time steps may even cause worse convergence [17, 18, 20]. As a consequence, FSI problems involving light structures and a high added mass become computationally expensive, due to the large number of sub iterations needed at each time step or the very small time steps needed for segregated schemes. To deal with this, important work was done using various strategies, some of which is intrusive, like semi-implicit coupling with a segregated fluid solver [21], others are somewhat intrusive, like the *Robin-Neumann* formulations [22], adding artificial compressibility to the fluid system to ease convergence [23, 24], or enforcing coupling conditions through domain decomposition methods [25, 26, 27]. Other strategies can be implemented completely non-intrusively: One such approach is segregated *Dirichlet-Neumann* schemes with specific force predictors to deal with relatively high added-mass and small time steps [28, 29]. Another approach - suitable for even higher added mass ratios - is using iterative schemes with convergence acceleration techniques. This includes either Aitken acceleration [30], or quasi-newton (QN) acceleration methods for the FSI problem formulated as a fixed point problem [31, 32, 33, 34, 35] (see a review in [36]). The latter strategy is the one we focus on in this work thanks to its high flexibility for both problems with high added-mass and black-box coupling with existing solvers. We note that in all these

partitioned schemes, there is often the notion of a predictor (for forces, displacements or velocities), the choice of which can be crucial for the acceleration of convergence.

To reduce the computational cost associated with strongly coupled FSI simulations, some non-intrusive reduced order models (ROMs) have been developed in the past years. These ROMs are generally data-driven models that combine various Machine Learning (ML) methods to predict solutions in the time-parameter space. We mention for example the works in [37, 38, 39, 40] where Proper Orthogonal Decomposition (POD) was combined with machine learning to predict parametric dynamical FSI solutions, or using spectral submanifolds for non-linearizable cases [41, 42]. In a recent work [43], we constructed data-driven ROMs for the solid subproblem intended to be coupled with high fidelity (HF) fluid full order models (FOMs), making thus a strong ROM-FOM coupling, capable of finding the FSI problem solutions with a much lower cost than the FOM-FOM while maintaining good accuracy.

In the present work, we combine ideas of partitioned coupling acceleration and data-driven ROM-FOMs to construct more effective predictors for a faster convergence of iterative partitioned FSI schemes. Moreover, we propose a strategy to adaptively update these predictors using HF data generated from the fluid FOM that is still activated during online computations. In doing so, the proposed methodology leverages physics-based insights from the fluid solver, thus establishing a physics-aware machine learning predictor. This method was inspired by our previous work in [43] where the replacement of the solid FOM with a ROM, albeit reduces the computational cost, can contribute to slower convergence due to the ROM inaccuracies. Hence, equipping this ROM-FOM coupling with stronger predictors can ensure even greater speedups. Importantly, this enhanced predictor can be used with classical FOM-FOM FSI coupling as well.

Other recent works have been focused on accelerating convergence of solving nonlinear problems using ML; interesting examples include [44, 45, 46, 47] where ML techniques (PCA dimensionality reduction, neural networks and random forests) are used to obtain correct parameters of nonlinear numerical solutions (initial guess, pseudo time steps for pseudo-transient continuation and Aitken relaxation parameter). In a similar work [48], although not using data-driven models, surrogate models (using e.g simplified faster models) have been utilized to obtain faster convergence of the Interface Quasi-Newton Inverse Least-Squares (IQN-ILS) method [32]. The novelty of the work presented here however is the use of data-driven techniques to accelerate convergence of partitioned FSI coupling.

Particularly, we use POD dimensionality reduction, followed by two regression models approximating the load to displacement (solid solver) and the displacement to load (fluid solver) operators to construct fast ROMs that can be coupled in few iterations to produce an initial guess for the next time step problem. In addition, the fluid ROM can be updated using the HF data obtained online and thus keeps a satisfying accuracy. The predictor thus takes advantage of the available HF data in a smarter way than a simple extrapolator does. A brief illustrative explanation of the complete methodology is presented in Figure 1. In this study, we focus on solid models with neglected inertia, but a natural extension can be used for dynamical models, which we intend to pursue in future work.

The remainder of this paper is structured as follows: In Sect. 2, the governing equations of FSI and the partitioned formulation are presented. Then, in Sect. 3, the proposed data-driven predictor approach is detailed. The results of the evaluation of this in terms of convergence acceleration are presented in Sect. 4: 3 test cases are used to demonstrate the efficiency of this approach. Finally, a conclusion is given in Sect. 5.

## 2. FOM-FOM fluid-structure interaction black-box coupling

In FSI problems, the nonlinear global problem is expressed by both the dynamic and kinematic coupling conditions at the interface  $\Gamma_{fsi}$ :

$$\begin{cases} \sigma_f \cdot \mathbf{n} = -\sigma_s \cdot \mathbf{n} & \text{at } \Gamma_{fsi} \\ \mathbf{v} = \dot{\mathbf{u}} & \text{at } \Gamma_{fsi} \end{cases} \quad (1)$$

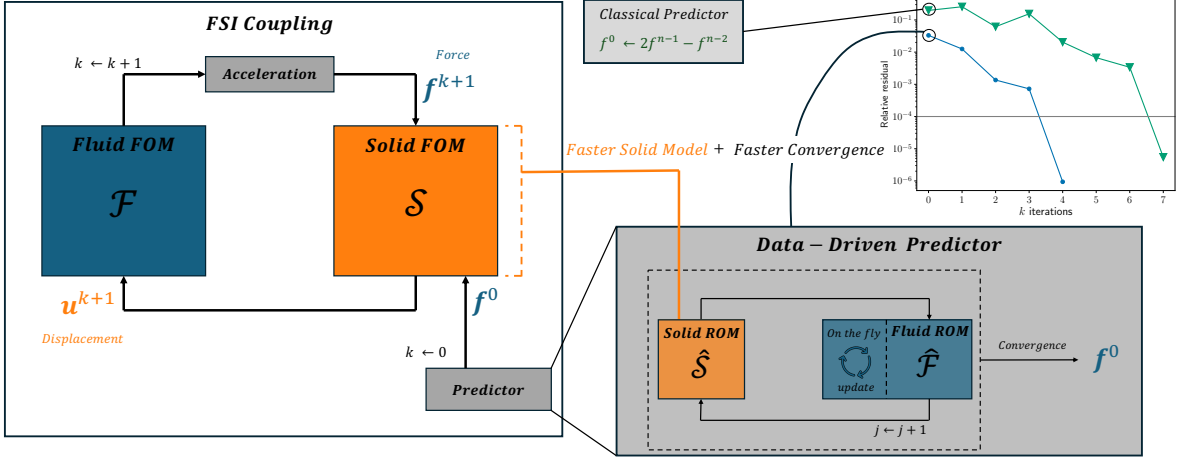


Figure 1: Brief illustrative representation of the presented approach.

where  $\sigma_f$  and  $\sigma_s$  are the Cauchy stresses applied by the fluid and the solid respectively,  $\mathbf{v}$  is the fluid velocity and  $\dot{\mathbf{u}}$  is the solid velocity. In mesh-based methods for the fluid problem, the superposition of the two *wet* interfaces translates into an additional coupling condition

$$\mathbf{x} = \mathbf{u} \text{ at } \Gamma_{fsi} \quad (2)$$

where  $\mathbf{x}$  is the fluid mesh displacement field and  $\mathbf{u}$  is the solid displacement.

The *Dirichlet-Neumann* coupling formulation represents the two solvers as distinct operators that exchange their input and output at each iteration. We represent the fluid solver operator as  $\mathcal{F}$ :

$$\mathcal{F} : \mathbb{R}^{2N} \rightarrow \mathbb{R}^N ; (\mathbf{u}_{|\Gamma_{fsi}}, \dot{\mathbf{u}}_{|\Gamma_{fsi}}) \rightarrow \mathbf{f}_{|\Gamma_{fsi}} \quad (3)$$

where  $\mathbf{u}_{|\Gamma_{fsi}}$  is the displacement field  $N$  is the number of interface degrees of freedom (*e.g.* the number of interface grid points times the number of components in mesh-based methods) and  $\mathbf{f}_{|\Gamma_{fsi}}$  represents the fluid viscous and pressure forces at  $\Gamma_{fsi}$ :

$$\mathbf{f}_{|\Gamma_{fsi}} = \sigma_f \cdot \mathbf{n}_{f|\Gamma_{fsi}}. \quad (4)$$

Similarly, the solid operator  $\mathcal{S}$  is defined as:

$$\mathcal{S} : \mathbb{R}^N \rightarrow \mathbb{R}^{2N} ; \mathbf{f}_{|\Gamma_{fsi}} \rightarrow (\mathbf{u}_{|\Gamma_{fsi}}, \dot{\mathbf{u}}_{|\Gamma_{fsi}}). \quad (5)$$

In strongly coupled schemes, the coupling conditions are enforced at the iterations convergence. This nonlinear problem can be written as the fixed-point:

$$(\mathcal{F} \circ \mathcal{S})(\mathbf{f}_{|\Gamma_{fsi}}) = \mathbf{f}_{|\Gamma_{fsi}}. \quad (6)$$

In this work, we consider problems where the solid inertia is negligible compared to the fluid one, i.e the solids studied here have negligible mass and are solved under quasi-static load. These assumptions have been relevant in modeling many FSI problems, like collapsible channels [49, 50, 51, 6]. Together with the flow incompressibility, this means that the FSI coupling in these cases is very strong and that the added mass is extremely large, as shown in [17, 18]. Standard Gauss-Seidel will not converge in the majority of such cases. Otherwise, QN-accelerated iterative schemes will take many iterations for each time step, making the potential acceleration provided by the new predictor even more advantageous. Moreover, given that only elastic (albeit nonlinear) material laws are used, the solid problem considered in this work is hence path-independent.

We note that the quasi-static load setting means that no velocity field is computed from the solid solver, and the mesh velocity can be computed using a proper time differentiation scheme [52] in the ALE mesh motion solver (usually part of the fluid solver). Consequently, the Dirichlet-Neumann formulation (6) becomes:

$$\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N ; \mathbf{u}|_{\Gamma_{f_{si}}} \rightarrow \mathbf{f}|_{\Gamma_{f_{si}}} \quad (7)$$

$$\mathcal{S} : \mathbb{R}^N \rightarrow \mathbb{R}^N ; \mathbf{f}|_{\Gamma_{f_{si}}} \rightarrow \mathbf{u}|_{\Gamma_{f_{si}}}. \quad (8)$$

In the following, for clarity, we will remove the interface subscript from  $\mathbf{f}|_{\Gamma_{f_{si}}}$  and  $\mathbf{u}|_{\Gamma_{f_{si}}}$ . In order to pass the field of interest at the interface, a mapping should be done to interpolate between the two grids. Energy-conserving mapping methods should be performed in the case of non-matching grids between the two systems. In the paper, we represent the mapping operators as  $\mathcal{M}_{\mathcal{F} \rightarrow \mathcal{S}}$  (from fluid grid to solid grid) and  $\mathcal{M}_{\mathcal{S} \rightarrow \mathcal{F}}$  (from solid grid to fluid grid). In the presented cases here, all the grids are matching on the interface, and the nearest-neighbor method is used. The reader is referred to [53, 54, 55] for additional details on mapping algorithms. In the rest of the paper, each field will be represented on the grid defined for the solver from which it is computed, *i.e.*  $\mathbf{u}$  on the solid mesh and  $\mathbf{f}$  on the fluid mesh, otherwise, proper subscripts will indicate if they are defined elsewhere, using the notations:  $\mathbf{f}_{\mathcal{F}}$ ,  $\mathbf{f}_{\mathcal{S}}$ ,  $\mathbf{u}_{\mathcal{F}}$  and  $\mathbf{u}_{\mathcal{S}}$ .

### 2.1. Coupling scheme

To solve (6), an iterative solution can be found by calling the 2 solvers sequentially at each iteration  $k$ :

$$\begin{cases} \mathbf{f}^k = \mathcal{F}(\mathbf{u}^k) \\ \mathbf{u}^{k+1} = \mathcal{S}(\mathbf{f}^k) \end{cases} \quad (9)$$

These are sometimes called Picard iterations, and the scheme is referred to as the Gauss-Seidel scheme. Note that, depending on which subproblem is computed first, the unknown may change: if the fluid solver is called first, (6) is replaced with the -generally- equivalent formulation:

$$(\mathcal{S} \circ \mathcal{F})(\mathbf{u}) = \mathbf{u}. \quad (10)$$

In this work, we use (6) because:

1. Force predictors (and (6)) were reported to be more efficient and result in fewer iterations than displacement predictors (and (10)) [56, 28].
2. Although out of the scope of this work, when dynamical solutions are solved in the solid problem, enforcing the kinematic coupling conditions at the interface means that the velocity field needs to be passed from the solid solver to the fluid solver, especially in black-box coupling when there is no knowledge about the time integration scheme in the solid solver. In this context, it is easier to accelerate the fluid forces field than it is for the displacement, where careful considerations should be done and/or additional cost to update the velocity field is spent.
3. The accuracy of the regression models of our data-driven ROMs will be higher when these models are trained on updated/relaxed force fields compared to when they are trained on non-relaxed fields. This is due to the low variance of the data points when they correspond to relaxed fields. This point will be highlighted with a numerical experiment in the results Section 4.2.

Equation (6) can also be rewritten as a block fixed-point system

$$\begin{pmatrix} \mathcal{S} \circ \mathcal{F} & \mathbf{0} \\ \mathbf{0} & \mathcal{F} \circ \mathcal{S} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix}. \quad (11)$$

Alternatively, a parallel formulation of the fixed-point can be sought if we write it as a Jacobi system:

$$\begin{pmatrix} \mathbf{0} & \mathcal{S} \\ \mathcal{F} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix} \quad (12)$$



which allows the simultaneous solution of the two solvers in parallel. While the ROM strategy we present in this work can be equivalently used in all these different formulations, we will focus here on the Gauss-Seidel system (6). The FSI fixed-point iterations to find the solution of (6) are called henceforth the *global iterations*.

## 2.2. Convergence acceleration

The simple use of Picard iterations, where the resultant forces field ( $\tilde{\mathbf{f}}$ ) is passed directly as an input to the solid solver ( $\mathbf{f} = \tilde{\mathbf{f}}$ ), shows poor convergence or may diverge in strongly coupled problems due to the added-mass instabilities [17]. To remedy this, convergence acceleration is done at the end of each iteration. A simple method is to use Aitken acceleration [57], where at the end of iteration  $k$ ,  $\tilde{\mathbf{f}}_{|\Gamma_{fsi}}$  is modified as:

$$\mathbf{f}^k = w^k \tilde{\mathbf{f}}^k + (1 - w^k) \mathbf{f}^{k-1} \quad (13)$$

where

$$w^k = -w^{k-1} \frac{\mathbf{r}^{k-1 T} (\mathbf{r}^k - \mathbf{r}^{k-1})}{\|\mathbf{r}^k - \mathbf{r}^{k-1}\|_2^2} \quad (14)$$

and  $\mathbf{r}^k$  is the residual at iteration  $k$ :  $\mathbf{r}^k = \tilde{\mathbf{f}}^k - \mathbf{f}^{k-1}$

Another approach is to formulate the problem (6) as a nonlinear problem to be solved using Quasi-Newton (QN) iterations:

$$\mathcal{R}(\mathbf{x}) = \mathbf{0} \quad (15)$$

where  $\mathcal{R}(\cdot)$  represents  $\mathcal{R}(\mathbf{f}) = (\mathcal{F} \circ \mathcal{S})(\mathbf{f}) - \mathbf{f}$  in the case of (6) and  $\mathcal{R}(\mathbf{u}) = (\mathcal{S} \circ \mathcal{F})(\mathbf{u}) - \mathbf{u}$  in the case of (10). Then, at iteration  $k$ , instead of passing the solver output  $\tilde{\mathbf{x}}^k$  directly, a QN relaxation is computed as

$$\mathbf{x}^k = \mathbf{x}^{k-1} - \mathbf{J}^{-1} \mathbf{r}^k \quad (16)$$

where

$$\mathbf{r}^k = \tilde{\mathbf{x}}^k - \mathbf{x}^k. \quad (17)$$

In this class of acceleration methods, the Jacobian (or inverse Jacobian) needed for the QN algorithm is approximated non-intrusively using snapshots of the past iterations. This generally gives faster convergence rate than Aitken relaxation [32, 35, 36]. Several acceleration methods have been developed, each differing in how the Jacobian is approximated, which unknown to consider (For example,  $\mathbf{x} = \mathbf{u}$ ,  $\mathbf{x} = \mathbf{f}$  or  $\mathbf{x} = \begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix}$  in the case of block formulation (11)) or how the past iteration information is used. In the test cases we show, we used the Quasi-Newton Inverse Least-Squares (IQN-ILS) method, introduced first in [32]. The original IQN-ILS algorithm 4 is recalled in the appendix.

## 2.3. Predictors

Another component of iterative FSI schemes is the predictor used to "kick-start" the next time step. Because of the time delay between the two solvers, there is no available "updated" solution for the solid solver to use in the first iteration. Usually, the last converged solution of the previous time step is used as a first guess

$$\mathbf{f}^{0,n} = \mathbf{f}^{n-1} \quad (18)$$

where the first superscript refers to the iteration number, and the second to the time step index for the current time step, and one superscript for the previous time steps indicates the converged solution. This notation will be used in the rest of the paper.

A linear extrapolation from the previous time steps such as

$$\mathbf{f}^{0,n} = 2\mathbf{f}^{n-1} - \mathbf{f}^{n-2} \quad (19)$$

can also be used, or, alternatively, a quadratic variation:

$$\mathbf{f}^{0,n} = 3\mathbf{f}^{n-1} - 3\mathbf{f}^{n-2} + \mathbf{f}^{n-3} \quad (20)$$

but whether this gives better convergence highly depends on each FSI problem and can result in bad performance if the time step is not sufficiently small. In this work, we propose an alternative approach to construct this predictor, as explained in the next section.

### 3. Non-intrusive ROM-FOM fluid-structure interaction acceleration

In this work, the strategy of data-driven acceleration of FSI problems consists in a combination of two approaches:

1. A data-driven ROM of the solid subproblem similar to the one proposed in [43].
2. An adaptive data-driven force predictor that acts as an efficient generator of a good initial guess of the fixed point FSI problem at each time step, enabling an easier convergence and thus faster overall computations.

Note that these two components, although sharing the same methods internally, are independent and can be used separately from one another. In particular, the force predictor can be used in more general cases, and can achieve good computational economies with practically no loss of accuracy, since the final solution depends on the state at the convergence, and rarely on the initial guess. On the other hand, a ROM for the solid part can achieve a speedup up of orders of magnitudes on the solid subproblem, with a small loss of accuracy, and this can be particularly useful if the computation time is predominantly due to the solid solver. In addition, using the two components enable the accumulation of the speedups as well as preventing the slowdown of the convergence rate due to the addition of the solid ROM.

#### 3.1. Solid reduced-order model

A data-driven reduced order model for the solid behavior is constructed following the approach presented in [43]. The ROM approximates the force to displacement function

$$\hat{\mathcal{S}} : \mathbb{R}^N \rightarrow \mathbb{R}^{N_S} ; \mathbf{f}|_{\Gamma_{fsi}} \rightarrow \mathbf{d} \quad (21)$$

where  $\mathbf{d}$  is the approximated displacement field of the solid domain and  $N_S$  is the dimension of the discretized solid domain. The approximation of the solid operator output is then :

$$\mathbf{u} = \mathcal{S}(\mathbf{f}) \approx \mathbf{d}|_{\Gamma_{fsi}} = \hat{\mathcal{S}}(\mathbf{f})|_{\Gamma_{fsi}}. \quad (22)$$

Hence, the solid ROM can predict the displacement solution in online computations with a much reduced cost and in a non intrusive manner. The ROM has three main components: the *encoder*, the *regressor* and the *decoder*.

##### 3.1.1. Forces encoder $\mathcal{E}_{\mathcal{F}}(\cdot)$

At the input of the ROM, an encoder reduces the dimensionality of the forces field at the interface. This is done using the popular POD method [58], hereby looking for a low dimensional linear subspace where the force field is projected:

$$\mathbf{f}(\mathbf{X}, t, \boldsymbol{\mu}) = \sum_i^{r_f} \boldsymbol{\Phi}_{fi}(\mathbf{X}) \mathbf{f}_{r,i}(t, \boldsymbol{\mu}) \quad (23)$$

where  $\boldsymbol{\mu}$  is a parameter of the FSI problem simulated, which, in this work, is associated to the fluid subproblem alone. The rank  $r_f$  is the dimension of the POD subspace, ideally very small  $r_f \ll N$ , and  $\boldsymbol{\Phi}_{fi}$  are the POD modes. Written in the discretized form, the force field  $\mathbf{f} \in \mathbb{R}^N$  can be represented as:

$$\mathbf{f} = \boldsymbol{\Phi}_f \mathbf{f}_r \quad (24)$$

where  $\boldsymbol{\Phi}_f \in \mathbb{R}^{N \times r_f}$  the matrix of POD modes and  $\mathbf{f}_r \in \mathbb{R}^{r_f}$  is the coordinates vector of the force in the reduced POD subspace.

In the offline phase, the POD modes are computed using Singular Value Decomposition (SVD) of snapshot matrix of HF results data  $\mathbf{F} \in \mathbb{R}^{N \times m}$  where  $m$  is the total number of available snapshots.

$$\mathbf{F} - \bar{\mathbf{F}} = \Phi_f \Sigma_f \mathbf{V}_f^* \quad (25)$$

where  $\bar{\mathbf{F}}$  is the temporal mean of the fluid forces,  $\Sigma_f$  is the singular values diagonal matrix, and  $\mathbf{V}_f^*$  is the conjugate transpose of the POD time coefficients. In fact, in this work, simulation results are collected from different values of  $\boldsymbol{\mu} \in \mathcal{P}$  of cardinal  $n_\mu$  and from all the simulated  $n_t$  timesteps corresponding to  $t \in [0., T]$ , including the non converged global iterations  $n_{k,i} \forall i \in \{1, \dots, n_t n_\mu\}$ ., giving

$$m = \sum_i^{n_t n_\mu} n_{k,i}. \quad (26)$$

Accordingly, and since the POD modes are orthogonal, the encoder acts as the dimensionality reducer of the force field quantity as :

$$\mathcal{E}_F(\mathbf{f}) : \mathbb{R}^N \rightarrow \mathbb{R}^{r_f} ; \mathbf{f}_r = \Phi_f^T(\mathbf{f} - \bar{\mathbf{F}}). \quad (27)$$

### 3.1.2. Regressor $\mathcal{I}_S(\cdot)$

The regressor approximates the relationship between the two low dimensional representations of the forces and the displacements

$$\mathcal{I}_S : \mathbb{R}^{r_f} \rightarrow \mathbb{R}^{r_u} ; \mathbf{f}_r \rightarrow \mathbf{u}_r. \quad (28)$$

Different existing methods can accomplish this task. In our experiments, the regression methods that provided the best accuracy are Reduced Basis Function (RBF) interpolation [59, 60] and second-order polynomial sparse approximation. In the former, the function is modeled as

$$\mathcal{I}_S(\mathbf{f}_r) = \sum_i^m \mathbf{w}_i \phi(\|\mathbf{f}_r - \mathbf{f}_{r,i}\|) + \mathbf{P}(\mathbf{f}_r) \quad (29)$$

where  $\phi(\cdot)$  is a kernel function,  $\mathbf{P}$  is the 1<sup>st</sup> order polynomial and  $\mathbf{f}_{r,i}$  are the RBF centers, chosen as the training points of the reduced forces, resulting eventually in a linear system to be solved for the RBF weights  $w_i$ . Alternatively, a polynomial regression of order 2 is used. The Force-Displacement relationship is thus modeled as a second order polynomial, here written in a discretized form:

$$\mathcal{I}_S(\mathbf{f}_r) = \mathbf{W}[\mathbf{f}_r \otimes \mathbf{f}_r] \quad (30)$$

where  $\otimes$  is the Kronecker product. The polynomial coefficients are arranged in  $\mathbf{W} \in \mathbb{R}^{r_u \times \hat{r}_f}$ , where  $\hat{r}_f = (r_f + 1)(r_f + 2)/2$ . The number of polynomial coefficients can be very large if  $r_f = \mathcal{O}(10)$ , and it is highly unlikely that all the polynomial terms are important for modeling  $\mathcal{I}(\cdot)$ . We thus propose using the Lasso regularization [61] in order to obtain a sparse model with as fewest terms as possible, the minimization is written as:

$$\mathbf{W}_i = \arg \min_{\mathbf{W}_i} \|\mathbf{u}_{r,i} - \sum_{j=1}^{\hat{r}_f} W_{ij} [\mathbf{f}_r \otimes \mathbf{f}_r]_j\|_2^2 + \lambda \sum_{j=1}^{\hat{r}_f} |W_{ij}| \quad \forall i \in \{1 \dots r_u\}. \quad (31)$$

The parameter  $\lambda$  promotes the sparsity of the solution  $\mathbf{W}$  and usually requires fine-tuning. In this work, we find the solution of (31) using the LARS algorithm presented in [62] and implemented in `scikit-learn` [63]. In order to avoid additional bias from the training data points distribution, a standardization should be done to the input before the inference. As a general rule of thumb, we recommend using (30) when a large amount of data is available (compared to the number of 2<sup>nd</sup> order polynomial terms) and (29) otherwise.

### 3.1.3. Displacement decoder $\mathcal{D}_S(\cdot)$

We use quadratic manifolds as nonlinear decoders for the reconstruction of the displacement field from obtained points in the latent space. Quadratic manifolds model the POD reconstruction error using the  $2^{nd}$  order polynomial terms of the reduced coordinates in the POD subspace, and associating them with a mapping operator  $\bar{\Phi}_U$  [64] :

$$\mathcal{D}_S(\mathbf{u}_r) : \mathbb{R}^{r_u} \rightarrow \mathbb{R}^{N_S} ; \mathbf{d} = \bar{\mathbf{U}} + \Phi_U \mathbf{u}_r + \sum_j^{r_u(r_u+1)/2} \bar{\Phi}_U(\mathbf{u}_r \otimes \mathbf{u}_r)_j. \quad (32)$$

Obtaining the POD modes  $\Phi_U \in \mathbb{R}^{N_S \times r_u}$  in the offline phase is similar to (25) :

$$\mathbf{U} - \bar{\mathbf{U}} = \Phi_U \Sigma_U \mathbf{V}_U^* \quad (33)$$

and a least-squares problem is solved for the quadratic operator  $\bar{\Phi}_U$  :

$$\bar{\Phi}_U = \arg \min_{\bar{\Phi} \in \mathbb{R}^{N_S \times r_u(r_u+1)/2}} \frac{1}{2} \|(\mathbf{I} - \Phi_U \bar{\Phi}_U^T)(\mathbf{U} - \bar{\mathbf{U}}) - \bar{\Phi}[\mathbf{u}_r \otimes \mathbf{u}_r]\|_F^2. \quad (34)$$

This generally enables a greater reconstruction accuracy for a practically negligible added cost (see [64, 43] for more details). Regarding the choice of the latent spaces dimensions,  $r_u$  and  $r_f$ : for  $r_u$ , we use the energy criterion to select the number of modes retaining  $\varepsilon = 99.99\%$  of the energy as:

$$\arg \min_{r_u \in [1, \min(N_S, m)]} S = \frac{\sum_i^{r_u} \sigma_i^2}{\sum_i^{\min(N_S, m)} \sigma_i^2} \quad (35)$$

s.t.  $S \leq \varepsilon$ .

As for  $r_f$ , we use a cross-validation strategy on a small percentage of the available data as test data, until a plateau of the validation error is attained (see [43] for more details about this cross-validation strategy).

### 3.1.4. In a nutshell:

The solid ROM can finally be symbolically represented as :

$$\hat{S}(\cdot) = \mathcal{D}_S \circ \mathcal{I}_S \circ \mathcal{E}_F(\cdot) \quad (36)$$

predicting the displacement field for a given applied force field on the interface:

$$\mathbf{d}(\mathbf{f}) = \hat{S}(\mathbf{f}) = \mathcal{D}_S(\mathcal{I}_S(\mathcal{E}_F(\mathbf{f}))). \quad (37)$$

Note that, for predicting the displacement at the interface only, which is the only necessary information at each global FSI iteration, a simple row selection of the POD modes matrix is used in the decoding phase, giving a new decoding operator  $\mathcal{D}_{S,\Gamma}$

$$\mathcal{D}_{S,\Gamma}(\mathbf{u}_r) : \mathbb{R}^{r_u} \rightarrow \mathbb{R}^N ; \mathbf{u} = \mathbf{d}_{\Gamma_{FSI}} = \bar{\mathbf{u}} + \Phi_u \mathbf{u}_r \quad (38)$$

where  $\Phi_u$  is the matrix of POD modes after removing the rows corresponding to the degrees of freedom (dofs) outside of the interface  $\Gamma_{FSI}$  from  $\Phi_U$ . We accordingly define

$$\hat{S}_\Gamma : \mathbb{R}^N \rightarrow \mathbb{R}^N ; \mathbf{f} \rightarrow \mathbf{u} = \mathcal{D}_{S,\Gamma}(\mathcal{I}_S(\mathcal{E}_F(\mathbf{f}))). \quad (39)$$

Some important remarks are to be made here :

- Since only the displacement at the interface is needed at each iteration, only the local version of the ROM  $\hat{S}_\Gamma$  is used at each iteration to pass the displacement to the fluid solver. The reduced coordinates  $\mathbf{u}_r$  of the converged iteration are stored at each time step (in a small  $\mathbf{U}_r^n \in \mathbb{R}^{r_u \times n_t}$  matrix), so that a reconstruction of the full displacement field in all the time steps can be made in parallel when needed, enabling then a computation of the stress and strain fields for example.

- If the solid ROM is constructed to take as input the fluid forces on the fluid grid directly, the ROM prediction can be made without the need of mapping the forces, simplifying further the coupling procedure and making a slight additional computational gain.
- In the context of mixed formulations of the solid problem, where other degrees of freedom are present (for example, pressure dofs in incompressible solid problems, or rotation dofs in shell elements), additional ROMs should be constructed for these unknowns. These fields are however not needed to pass the interface displacement to  $\mathcal{F}$  and the reduced force coordinates can be stored at each converged iteration to compute the full solution when needed (for example at the end of the simulation). An example of this will be demonstrated in test case n°3.

### 3.2. Adaptive data-driven predictors with fluid and solid ROMs

In this work, we propose to construct a data-driven predictor, based on the use of information from past data, for example past iterations/time steps, or historical simulation results obtained at different parameters values, thus providing a better initial guess than a finite-differences-based extrapolation. In addition to the solid ROM described in 3.1, another surrogate is constructed to approximate the inverse of  $\hat{\mathcal{S}}$  at the interface, *i.e.* approximating the  $\mathbf{u}$  to  $\mathbf{f}$  relationship. Note that, contrary to the solid ROM presented above, this fluid surrogate should take into account the dynamical effects

$$\hat{\mathcal{F}}(\mathbf{u}, t) = \mathbf{f}. \quad (40)$$

This is done using a discrete approach, where the input of the surrogate contains not only the current displacement but also the forces at the previous time step

$$\hat{\mathcal{F}}(\mathbf{u}^{k,n+1}, \mathbf{f}^{k,n}) = \mathbf{f}^{k,n+1}. \quad (41)$$

This is in principle similar to the Dynamic Mode Decomposition with Control (DMDc) [65] concept, the only difference here being the consideration of the system nonlinearity through the nonlinear regressor and/or the update strategy adopted.

#### 3.2.1. Prediction of a better initial guess: Inner coupling

A "reduced FSI coupling" is launched at the start of each time step, where each subsystem  $\mathcal{F}$  and  $\mathcal{S}$  is replaced by its reduced equivalent  $\hat{\mathcal{F}}$  and  $\hat{\mathcal{S}}$ , and the solution

$$\hat{\mathcal{F}}(\hat{\mathcal{S}}(\mathbf{f}^0)) = \mathbf{f}^0 \quad (42)$$

is sought at a fraction of the computational time needed for the FOM-FOM problem solver calls. Those iterations will be called henceforth *local iterations*. In [48], reduced physics-based models were used as surrogates to compute the local iterations, specifically to enhance the inverse Jacobian approximation in IQN-ILS, while the new initial guess did not particularly produce less overall computational time.

We note that using our proposed data-driven models, and constructing them so that they consider inputs and outputs on the same grid (the fluid grid in our case), allows to bypass the need for grid mapping. In addition, we propose an adaptation strategy to update the prediction capability of the fluid surrogate, especially since new HF data from the fluid FOM solver is available at the online computations. We note that the tolerance required for the convergence of these local iterations  $\delta_r$  should not be very small, since the goal here is merely to obtain a closer initial guess than the previous time step solution, and to avoid a large number of iterations that could slow down computations. In the same direction, we propose that these local iterations use a simple Aitken relaxation (14), since using QN-acceleration would add a computational cost to find and store an additional inverse Jacobian approximate, while the tolerance required is already not very small. Furthermore, if the convergence is not reached, the predictor should be deactivated, and the previous time step solution should be used instead, since poor convergence could suggest that the dynamical nonlinearity is not captured by the two ROMs involved, and thus their predictions may be too inaccurate and lead to poor global convergence. In Algorithm 1, we detail the local iterations' procedure of the reduced models.

In the next part, the components of  $\hat{\mathcal{F}}$  will be detailed.

---

**Algorithm 1:** Local ROM-ROM iterations for finding an initial guess -  $Pred(\cdot)$ 

---

**Input:** Fluid ROM  $\hat{\mathcal{F}}$   
Solid ROM  $\hat{\mathcal{S}}$   
Initial relaxation  $w_0$   
Convergence tolerance  $\delta_r = \mathcal{O}(10^{-2})$   
Maximum number of iterations  $M$   
Previous time step fluid forces  $\mathbf{f}^{n-1}, \mathbf{f}^{n-2}, \mathbf{f}^{n-3}$   
**Result:** Next initial guess  $\mathbf{f}^{0,n}$

- 1  $\mathbf{f}^{0,n} = 2\mathbf{f}^{n-1} - \mathbf{f}^{n-2}$  or  $\mathbf{f}^{0,n} = 3\mathbf{f}^{n-1} - 3\mathbf{f}^{n-2} + \mathbf{f}^{n-3}$
- 2  $j = 1$
- 3  $e = 1$
- while**  $e > \delta$  and  $k < M$  **do**
- 4    $\mathbf{u}_r^j = \hat{\mathcal{S}}(\mathbf{f}^{j-1})$
- 5    $\tilde{\mathbf{f}}^j = \hat{\mathcal{F}}(\mathbf{u}^j, \mathbf{f}^{n-1})$
- 6    $\mathbf{r}^j = \tilde{\mathbf{f}}^j - \mathbf{f}^{j-1}$
- 7    $e = \|\mathbf{r}^j\|_2 / \|\tilde{\mathbf{f}}^j\|_2$
- if**  $e > \delta_r$  **then**
- 8     $w^j = -w^{j-1} \frac{\mathbf{r}^{j-1 T} (\mathbf{r}^j - \mathbf{r}^{j-1})}{\|\mathbf{r}^j - \mathbf{r}^{j-1}\|_2^2}$
- 9     $\mathbf{f}^j = w^j \tilde{\mathbf{f}}^j + (1 - w^j) \mathbf{f}^{j-1}$
- else**
- 10     $\mathbf{f}^{0,n} = \tilde{\mathbf{f}}^j$
- 11    **End algorithm**
- 12     $j = j + 1$
- 13  $\mathbf{f}^{0,n} = \mathbf{f}^{n-1}$  // *Failure of convergence*

---

### 3.2.2. Fluid ROM components:

The fluid ROM  $\hat{\mathcal{F}}$  is constructed with the same philosophy as its solid counterpart  $\hat{\mathcal{S}}_\Gamma$ , in the sense of finding a relationship in a latent space. This can be done using two approaches: using separate encoders for the displacement and forces and the reduced representation is composed of the two encoded variables:

$$\hat{\mathcal{F}} : \mathbb{R}^{2N} \rightarrow \mathbb{R}^N ; (\mathbf{u}, \mathbf{f}^{n-1}) \rightarrow \mathbf{f} = \mathcal{D}_F(\mathcal{I}_F([\mathcal{E}_S(\mathbf{u}), \mathcal{E}_F(\mathbf{f}^{n-1})]^T)). \quad (43)$$

The second approach, denoted as  $\hat{\mathcal{F}}_H$  consists of constructing a new unknown from the concatenation of the 2 variables in the high-dimensional space, with a new *hybrid* encoder  $\mathcal{E}_H(\cdot)$ :

$$\hat{\mathcal{F}}_H : \mathbb{R}^{2N} \rightarrow \mathbb{R}^N ; (\mathbf{u}, \mathbf{f}^{n-1}) \rightarrow \mathbf{f} = \mathcal{D}_F(\mathcal{I}_F(\mathcal{E}_H(\mathbf{u}, \mathbf{f}^{n-1}))). \quad (44)$$

The two ROMs' components are summarized in the illustrative Figures 2 and 3.

The fluid ROM  $\hat{\mathcal{F}}$  differs from  $\hat{\mathcal{S}}_\Gamma$  in that:

- The decoder  $\mathcal{D}_F(\cdot)$  and encoder  $\mathcal{E}_S(\cdot)$  are the equivalent of the inverses of their encoder and decoder counterparts  $\mathcal{E}_F(\cdot)$  and  $\mathcal{D}_{S,\Gamma}(\cdot)$ , respectively. They are then defined as :

$$\mathcal{D}_F(\mathbf{f}_r) : \mathbb{R}^{r_f} \rightarrow \mathbb{R}^N ; \mathbf{f} = \bar{\mathbf{F}} + \Phi_f \mathbf{f}_r \quad (45)$$

$$\mathcal{E}_S(\mathbf{u}) : \mathbb{R}^N \rightarrow \mathbb{R}^{r_u} ; \mathbf{u}_r = \Phi_u^T(\mathbf{u} - \bar{\mathbf{U}}). \quad (46)$$

This means that in the case of  $\hat{\mathcal{F}}$ , no additional training cost is spent for obtaining  $\mathcal{D}_F(\cdot)$  and  $\mathcal{E}_S(\cdot)$ , since the modes are already computed during the training of  $\hat{\mathcal{S}}_\Gamma$ . As for  $\mathcal{E}_H(\cdot)$ , this is also done using the POD but on the new variable  $(\mathbf{u}, \mathbf{f}^{n-1})$

$$\mathcal{E}_H(\mathbf{u}, \mathbf{f}^{n-1}) : \mathbb{R}^{2N} \rightarrow \mathbb{R}^{r_h} ; \mathbf{h}_r = \Phi_H^T([\mathbf{u}, \mathbf{f}^{n-1}]^T - [\bar{\mathbf{U}}, \bar{\mathbf{F}}]^T). \quad (47)$$

- A new regressor  $\mathcal{I}_F$  is used that takes into account the dynamical effect using the previous timestep force  $\mathbf{f}^{n-1}$  in the augmented regression input

$$\mathcal{I}_F : \mathbb{R}^{r_f+r_u} \rightarrow \mathbb{R}^{r_f} ; [\mathbf{u}_r, \mathbf{f}_r^{n-1}]^T \rightarrow \mathbf{f}_r. \quad (48)$$

While different regression forms can be used, we use here linear regression or the RBF regressor (29), which in our view, represent a good trade-off option between fitting efficiency and training efficiency, which is crucial for our method since an online update strategy is adopted (more details will be given in Section 3.2.3).

- Note that in order to obtain accurate evaluations of  $\hat{\mathcal{F}}$  and thus reliable predictors, the direct output of the HF fluid solver  $\mathcal{F}$  should be "seen" during training. We recall that in the available HF data, we distinguish between the forces computed from the fluid solver  $\mathbf{f}^k$ , and the QN-updated forces  $\tilde{\mathbf{f}}^k$ , arranged in a new snapshot matrix that we call  $\tilde{\mathbf{F}}$ . Although, for obvious reasons, the new regressor  $\mathcal{I}_F$  should necessarily use  $\tilde{\mathbf{F}}$  (or rather, its reduced coordinates  $\Phi_f^T \tilde{\mathbf{F}}$ ). Only a combination of  $\mathbf{F}$  and  $\tilde{\mathbf{F}}$  suffices to compute the POD modes  $\Phi_f$ . In our experiments, we used the concatenation of all the snapshots  $[\mathbf{F}, \tilde{\mathbf{F}}]$ , but a more efficient choice could be for example to only include the first iteration results in  $\tilde{\mathbf{F}}$  since the other iterations' resultant forces –closer to FSI convergence– would be very close to the updated forces, and would only add little information for computing the POD subspace.

In our numerical experiments, the two approaches  $\hat{\mathcal{F}}$  and  $\hat{\mathcal{F}}_H$  yield nearly identical results, we then proceed with detailing only the first approach  $\hat{\mathcal{F}}$  since its associated offline step is computationally more efficient (dropping the need of learning  $\mathcal{E}_H$ ).

**Remark:** We should note here that, with our choices of the dimensionality reduction methods, the evaluation of  $\mathcal{D}_S(\cdot)$  and  $\mathcal{E}_S(\cdot)$  can be dropped at the local iterations (Algorithm 1) during the

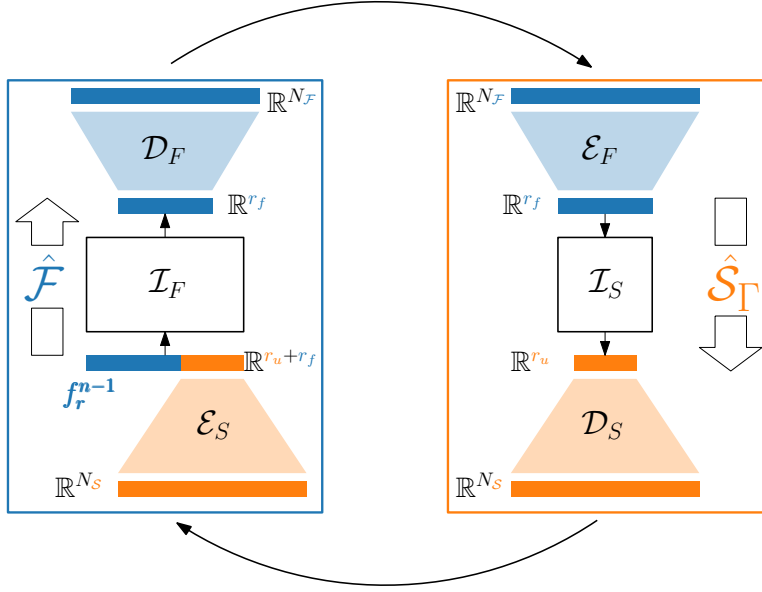


Figure 2: Fluid and solid ROM components

inference of the two ROMs  $\hat{\mathcal{F}}$  and  $\hat{\mathcal{S}}$ , since, in our case, using POD as our encoder-decoder implies that  $\mathcal{E}_S(\mathcal{D}_S(\mathbf{u}_r)) = \mathbf{u}_r$ . However, the full force field must be recovered at each iteration (of the reduced fixed-point problem) because the relaxation used in line 9, together with an initial guess from the fluid solver (line 1) means that  $\mathbf{f}^k$  contains a part that falls outside the POD subspace  $\Phi_f$ . Thus, the convergence should be verified at the high-dimensional space rather than at the latent space. Furthermore, as stated earlier, and as shown in [43], the force field is not easily compressed using POD, and the ignored modes may be necessary as in the contribution to the force values used when checking the convergence.

The offline strategy of the solid and fluid ROMs are outlined in Algorithm 3.

### 3.2.3. Fluid ROM online update:

At the end of the offline phase, the full snapshot matrices are no longer of use, since their reduced representation is now learned, and thus can be freed from memory. However, we store a chunk of size  $p$  from the reduced coordinates of the training input and output data of  $\hat{\mathcal{F}}$ :  $\mathbf{U}_r \in \mathbb{R}^{r_u \times p}$  and  $\tilde{\mathbf{F}}_r \in \mathbb{R}^{r_f \times p}$  as:

$$\mathbf{U}_r = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}_r^1 & \mathbf{u}_r^2 & & \mathbf{u}_r^p \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{F}}_r = \begin{bmatrix} | & | & \dots & | \\ \tilde{\mathbf{f}}_r^1 & \tilde{\mathbf{f}}_r^2 & & \tilde{\mathbf{f}}_r^p \\ | & | & & | \end{bmatrix}. \quad (49)$$

These reduced representations of data points will be updated in the online computations by replacing the data points with the least relevant information by new snapshot couples of  $[\mathbf{u}_r^k | \tilde{\mathbf{f}}_r^k]$  available at each iteration :

$$\mathbf{U}_r \leftarrow \begin{bmatrix} | & | & \dots & | & | \\ \mathbf{u}_r^2 & \mathbf{u}_r^3 & & \mathbf{u}_r^p & \mathbf{u}_r^k \\ | & | & & | & | \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{F}}_r \leftarrow \begin{bmatrix} | & | & \dots & | & | \\ \tilde{\mathbf{f}}_r^2 & \tilde{\mathbf{f}}_r^3 & & \tilde{\mathbf{f}}_r^p & \tilde{\mathbf{f}}_r^k \\ | & | & & | & | \end{bmatrix}. \quad (50)$$

These new (reduced) snapshots represent the high fidelity information stemming from the HF fluid solver operations. It can thus be used to enrich the fluid ROM. Specifically, after a certain advancement along the transient simulation, defined for example by a number  $Z$  (chosen by the user) of global FSI iterations, these now updated matrices  $\mathbf{U}_r$  and  $\tilde{\mathbf{F}}_r$  can be used to retrain the fluid regressor (48). By



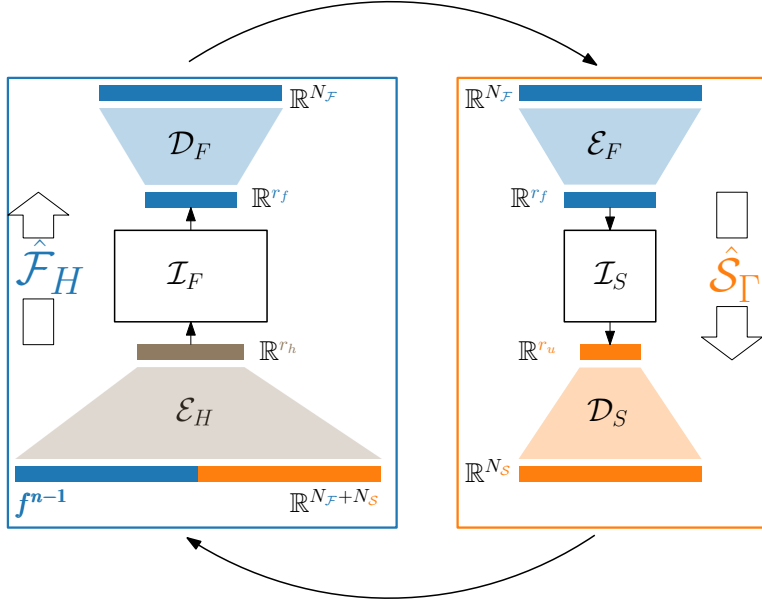


Figure 3: Fluid and solid ROM components - Hybrid variant

doing so, we ensure that the fluid ROM  $\hat{\mathcal{F}}$  maintains a high enough fidelity so that the provided initial guess does indeed help the FSI converge faster, especially since the newest information, from the latest time steps and iterations will be used.

Note that the matrices  $\mathbf{U}_r$  and  $\tilde{\mathbf{F}}_r$  are of small sizes since  $r_u \ll N$  and  $r_f \ll N$ . In addition, the "width" of these matrices can be kept at its maximum  $p$ , also defined by the user, keeping the memory cost small and constant. As for the CPU cost, retraining the regressor (29) or linear regression, i.e solving a linear system of size  $(p+r_u+r_f) \times (p+r_u+r_f)$  or a  $p \times (1+r_u+r_f)$  least-square system, and only after (a large) number of iterations  $Z$ , represents only a small fraction of the FSI timestep solution.

It should be emphasized that this online update strategy is limited to the update of the regressor for simplicity. An extension of this approach could be done by updating the encoder-decoder as well. For example, a recursive method like the one presented in [66] could be used to update the POD bases using data obtained on the fly.

### 3.3. Summary

To summarize, the proposed acceleration method can be easily implemented in a non-intrusive manner. Using HF data obtained with varying fluid parameter simulations, two separate ROMs can be trained efficiently and then used in a new simulation with unseen values of the parameters. This added block in the FSI scheme provides an initial force field that will "jump-start" the coupling at each time step resulting in faster convergence and faster overall computation. During the new unseen simulation, the fluid ROM can be adapted using the online results coming from the FOM fluid solver, constituting a physics-aware predictor that can be used effectively even in extrapolative regions. The global FSI iterations with the new prediction approach is outlined in Algorithm 2. We stress once again the nonintrusivity of this approach, since calling external solvers as black-boxes as done in lines 11 and 6 is completely possible. Moreover, additional calculations can be done in line 24 using the imposed (already computed) displacements to compute for example the stress and strain fields. A sketch of the global methodology is shown in Figure 4 for illustration.

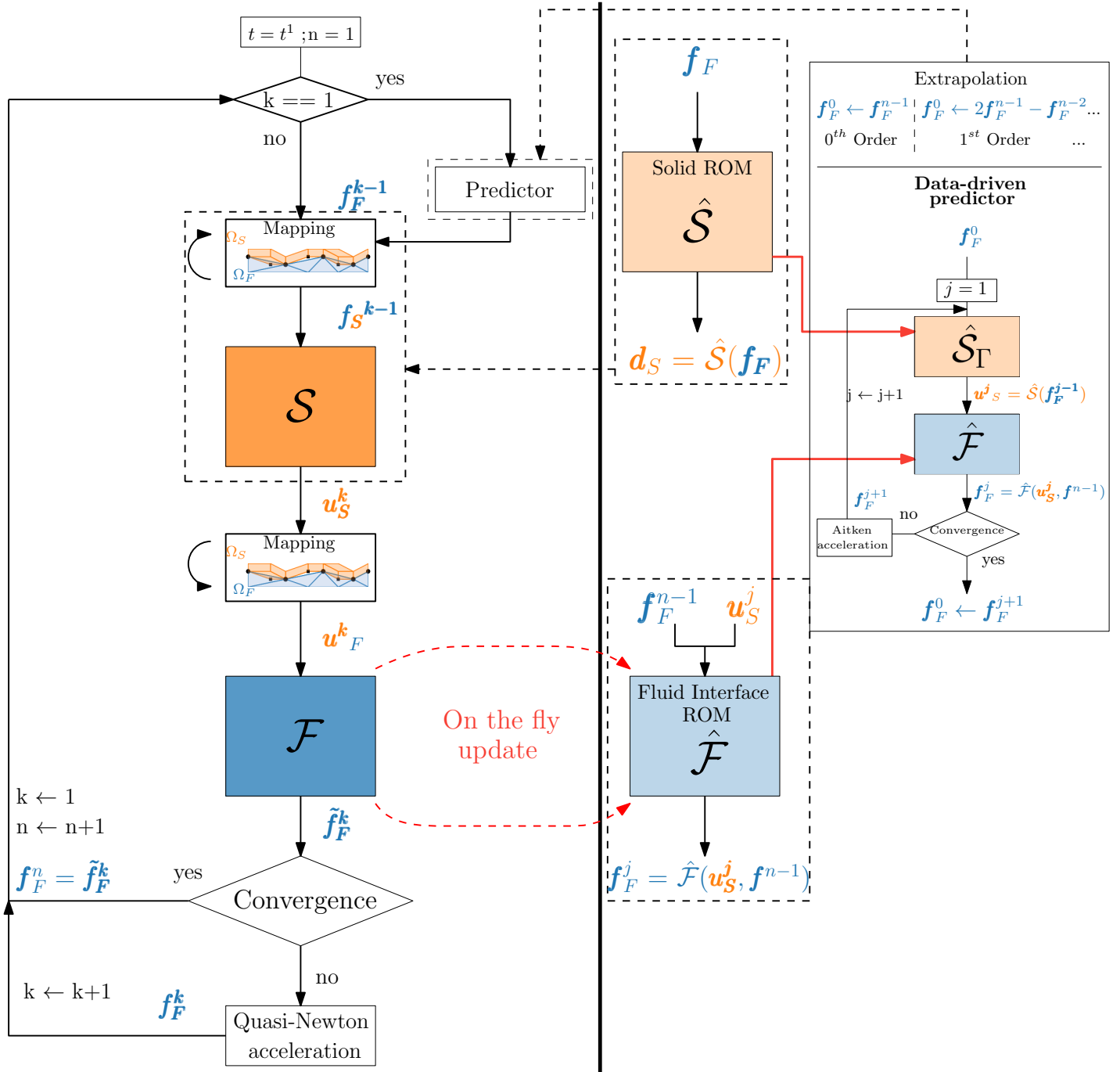


Figure 4: Global FSI coupling scheme with the ROM-FOM coupling and data-driven adaptive predictors.

---

**Algorithm 2:** ROM-FOM - Global FSI scheme
 

---

```

1   $n = 1$ 
2   $z = 1$ 
   while  $n < N_t$  do
3      $k = 1$ 
4      $e = 1$ 
5     Predictor :  $\mathbf{f}^{0,n} = \text{Pred}(\mathbf{f}^{n-1}, \mathbf{f}^{n-2})$  using Algorithm 1
   while  $e > \delta$  and  $k < N_k$  do
6     Solid solver :  $\begin{cases} \mathbf{f}_{|S}^{k-1} = \mathcal{M}_{\mathcal{F} \rightarrow \mathcal{S}}(\mathbf{f}_{|\mathcal{F}}^{k-1}) \\ \tilde{\mathbf{u}}^k = \mathcal{S}(\mathbf{f}_{|S}^{k-1}) \end{cases}$  or  $\tilde{\mathbf{u}}^k = \hat{\mathcal{S}}_{\Gamma}(\mathbf{f}_{|\mathcal{F}}^{k-1})$ 
       if Block Formulation (11) is used then
7          $\Delta \mathbf{u}^k = \text{Acc}(\tilde{\mathbf{u}}^{k-1}, \mathbf{r}^k)$  using block QN acceleration, e.g [33]
8          $\mathbf{u}^k = \mathbf{u}^k + \Delta \mathbf{u}^k$ 
       else
9          $\mathbf{u}^k = \tilde{\mathbf{u}}^k$ 
10      Displacement mapping :  $\mathbf{u}_{|\mathcal{F}}^k = \mathcal{M}_{\mathcal{S} \rightarrow \mathcal{F}}(\mathbf{u}_{|S}^k)$ 
11      Fluid solver :  $\tilde{\mathbf{f}}^k = \hat{\mathcal{F}}(\mathbf{u}_{|\mathcal{F}}^k, t)$ ;
12      Update matrices  $\mathbf{U}_r$  and  $\tilde{\mathbf{F}}_r$  with  $\mathbf{u}^k$  and  $\tilde{\mathbf{f}}^k$  with (50)
13       $z = z + 1$ 
       if  $z > Z$  then
14         Refit  $\mathcal{I}_F(\cdot)$  using the updated  $\mathbf{U}_r$  and  $\tilde{\mathbf{F}}_r$ 
15          $z = 0$ 
16       $\mathbf{r}^k = \tilde{\mathbf{f}}^k - \mathbf{f}^{k-1}$ 
17       $e = \|\mathbf{r}^k\|_2 / \|\tilde{\mathbf{f}}^k\|_2$ 
       if  $e > \delta$  then
18          $\Delta \mathbf{f}^k = \text{Acc}(\tilde{\mathbf{f}}^k, \mathbf{r}^k)$  using e.g Algorithm 4
19          $\mathbf{f}^k = \mathbf{f}^{k-1} + \Delta \mathbf{f}^k$ 
       else
20          $\mathbf{f}^n = \tilde{\mathbf{f}}^k$  // Convergence
21         Store converged reduced coordinate  $\mathbf{u}_r \in \mathbb{R}^{r_u}$  in  $\mathbf{U}_r^n \in \mathbb{R}^{r_u \times n}$ 
22          $k = k + 1$ 
23      $n = n + 1$ 
24 Retrieve the full displacement field using  $\mathbf{U}^n = \Phi_U \mathbf{U}_r^n$ 

```

---

---

**Algorithm 3:** ROM-FOM - Offline stage
 

---

**Input:** Force snapshots matrix  $\mathbf{F}$

Direct Force snapshots matrix  $\tilde{\mathbf{F}}$

Displacement snapshots matrix  $\mathbf{U}$ ,

Number of selected force modes  $r_f$ ,

Number of selected displacement modes  $r_u$ ,

Update batch size  $p$

**Result:**  $\{\Phi_f, \bar{\mathbf{F}}, \Phi_u, \bar{\Phi}_u, \bar{\mathbf{U}}, \mathcal{I}_F(\cdot), \mathcal{I}_S(\cdot), \mathbf{U}_r, \tilde{\mathbf{F}}_r\}$

1 Arrange a combination of  $\mathbf{F}$  and  $\tilde{\mathbf{F}}$ :  $\hat{\mathbf{F}}$

2 Compute mean of  $\hat{\mathbf{F}}$ :  $\bar{\mathbf{F}}$

3 SVD:  $\hat{\mathbf{F}} - \bar{\mathbf{F}} = \Phi_f \Sigma_f \mathbf{V}_f^*$

4 Truncation of  $r_f$  modes:  $\Phi_f \leftarrow \begin{bmatrix} | & | & & | \\ \Phi_{1,f} & \Phi_{2,f} & \dots & \Phi_{r_f,f} \\ | & | & & | \end{bmatrix}$

5 Compute mean of  $\mathbf{U}$ :  $\bar{\mathbf{U}}$

6 SVD:  $\mathbf{U} - \bar{\mathbf{U}} = \Phi_U \Sigma_U \mathbf{V}_U^*$

7 Solve least-squares problem  $\bar{\Phi}_u[\mathbf{u}_r \otimes \mathbf{u}_r] = (\mathbf{I} - \Phi_U \Phi_U^T)(\mathbf{U} - \bar{\mathbf{U}})$  for  $\bar{\Phi}_u$

8 Truncation of  $r_u$  modes:  $\Phi_U \leftarrow \begin{bmatrix} | & | & & | \\ \Phi_{1,U} & \Phi_{2,U} & \dots & \Phi_{r_u,U} \\ | & | & & | \end{bmatrix}$

9 Retrieve the reduced coordinates  $\mathbf{F}_r$  and  $\tilde{\mathbf{F}}_r \equiv \Sigma_f \mathbf{V}_f^*$

10 Retrieve the reduced coordinates  $\mathbf{U}_r \equiv \Sigma_U \mathbf{V}_U^*$

11 Fit  $\mathcal{I}_S(\cdot)$ :  $\mathcal{I}_S(\mathbf{F}_r) \approx \mathbf{U}_r$

12 Retrieve the reduced coordinates of converged iterations  $\mathbf{F}_r^n$  by column-extraction from  $\tilde{\mathbf{F}}_r$

13 Fit  $\mathcal{I}_F(\cdot)$ :  $\mathcal{I}_F(\mathbf{U}_r, \mathbf{F}_r^n) \approx \tilde{\mathbf{F}}_r$

14 Store  $p$ -size batches from  $\mathbf{U}_r$  and  $\tilde{\mathbf{F}}_r$

---

## 4. Results

In the following, we intend to evaluate the performance of the ROM strategy proposed in terms of both the gained of CPU time (speedup) and number of fixed-point iterations. We demonstrate this on three transient FSI test cases with a very strong coupling, and high added mass in  $1D$ ,  $2D$  and  $3D$ , respectively, with low  $Re$  (laminar) flows. In each case, we apply the proposed ROM-based predictors on the problem in 2 different configurations: FOM-FOM and ROM-FOM, with the latter using a solid ROM in the actual solid computations. Since the computational time is almost always dominated by the solvers' internal iterations, and the addition of the predictors only add a fraction of that time, the comparisons will be mainly done using the number of coupling subiterations. We assess how the addition of the new data-driven predictors affects the number of needed iterations. For the sake of completeness, we also compare in terms of the total CPU time spent to ensure that speedups are realized, especially in the last 2 (more realistic) cases. We also check that, in the case of replacing the solid solver with a solid ROM, the errors of the computed fields are sufficiently low due to the predictive ability of the solid ROM. The coupling operations are done using the coupling component in the multiphysics simulation code `KratosMutiphysics` [67]. In what follows, we will denote  $\hat{\mathcal{S}}_\Gamma$  and  $\hat{\mathcal{F}}$  as *SROM* and *FROM* respectively. The standard predictors in (18), (19) and (20) will be denoted by *Constant Extrapolator*, *Linear Extrapolator* and *Quadratic Extrapolator* respectively, while the new proposed predictor will be called *Data-Driven Predictor*.

### 4.1. Example 1: 1D flexible tube model

The model of flexible tube and related HF partitioned solvers proposed by Degroote et al. [20] are used here. The flow is assumed to be incompressible with constant density  $\rho$ . Both fluid mass and momentum conservation equations (neglecting viscosity) read

$$\begin{cases} \partial_t a + \partial_x(av) = 0, \\ \partial_t(av) + \partial_x(av^2) + \frac{a}{\rho} \partial_x p = 0, \quad t > 0, x \in [0, L] \end{cases} \quad (51)$$

where  $v$  is the velocity,  $a$  is the time-dependent tube cross section and  $p$  is the pressure. From the fluid side, the unknowns are both velocity and pressure. For the thin flexible tube with a thickness  $h_s$ , a quasi-static model

$$a = a(p)$$

is used (retaining only the vessel stress in the circumferential direction). The following nonlinear elastic stress-strain law  $\sigma_{\varphi\varphi}(\epsilon_{\varphi\varphi})$  is used [43]:

$$\begin{cases} \sigma_{\varphi\varphi} = E \epsilon_{\varphi\varphi} & \text{if } |\epsilon_{\varphi\varphi}| < \epsilon_0 \\ \sigma_{\varphi\varphi} = E/5 \epsilon_{\varphi\varphi} + 20 & \text{if } \epsilon_{\varphi\varphi} \geq \epsilon_0 \\ \sigma_{\varphi\varphi} = E/5 \epsilon_{\varphi\varphi} - 20 & \text{if } \epsilon_{\varphi\varphi} \leq -\epsilon_0 \end{cases} \quad (52)$$

with  $\epsilon_0 = 2 \cdot 10^{-3}$  and  $E = 12500 \text{ Pa}$ . Figure 5 shows a schematic description of this problem. A non-reflective boundary condition is used on the  $x = L$  boundary as  $\frac{du}{dt} = \frac{1}{c} \frac{dp}{dt}$  with  $c$  the fluid wave speed  $c = \sqrt{\frac{a}{\frac{da}{dp}}}$ . The prescribed inlet ( $x = 0$ ) velocity  $v_0$  is computed using the solution of a nonlinear

Duffing equation in order to evaluate the ROM performances in problems with complex dynamics, and the capacity to benefit from the HF fluid solver output to handle the dynamics:

$$\begin{cases} \ddot{u}(t) = a u(t) + b u(t)^2 + c u(t)^3 + d + p \cos(ft) + e \dot{u}(t) \quad \forall t \in [0, 120] \\ u(0) = 10 ; \quad \dot{u}(0) = 0. \\ v_0(t) = (g u(t) + h) r(t) \\ r(t) = 1 \quad \forall t \in [0, 20] \\ r(t) = 0.9 + 0.1 \sin(t\pi/40) \quad \forall t \in [20, 60], \quad \# \text{Negative ramp function} \\ r(t) = 0.8 \quad \forall t \in [60, 120]. \end{cases} \quad (53)$$

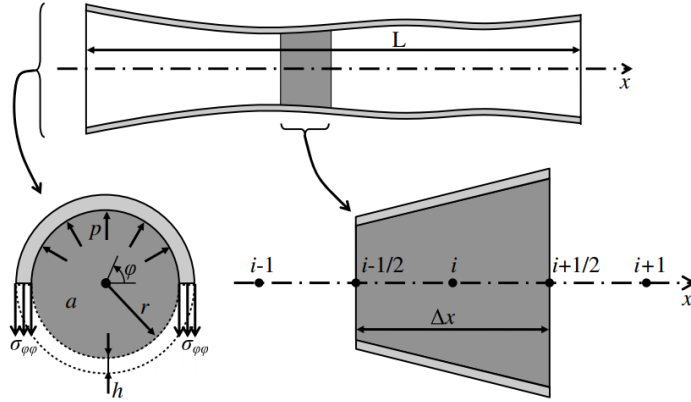


Figure 5: Example 1 - 1D flexible tube test case schematic explanation (from [20]).  $\sigma_{\varphi\varphi}$  is the vessel hoop stress,  $h$  is the thickness and  $\Delta x$  is the length of the finite volume cell.

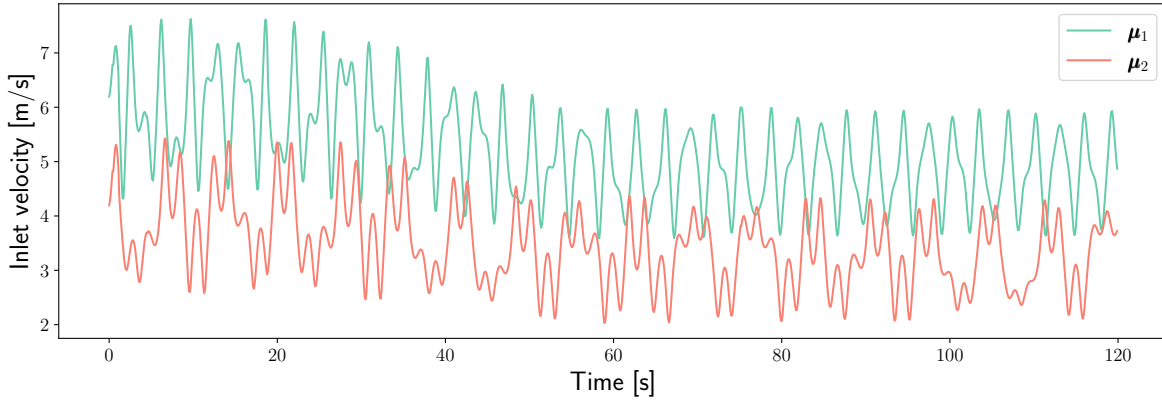


Figure 6: Inlet velocity signals corresponding to  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\mu}_2$  parameters values.

We fix  $(a, b, c, d, e, g, p) = (-1, 0, -0.002, -1, -0.02, 1/60, 360)$  and we parameterize this signal with the parameter vector  $\boldsymbol{\mu} = (f, h)^T$  allowing the generation of different frequencies and amplitudes. For this study, the two values  $\boldsymbol{\mu}_1 = [2, 6]^T$  and  $\boldsymbol{\mu}_2 = [0.9, 4]^T$  have been selected, leading to the signals shown in Figure 6.

The fluid flow equations (51) are solved using a second order finite volume scheme with 100 cells and the solid section  $\mathbf{a}(\mathbf{p})$  is computed at each iteration as the solution of the scalar minimization problem

$$\mathbf{p} \sqrt{\frac{\mathbf{a}}{\pi}} = \sigma_{\varphi\varphi} \left( \frac{\sqrt{\frac{\mathbf{a}}{\pi}} - r_0}{r_0} \right) h_s. \quad (54)$$

The solid subdomain in this case is the interface itself, and the nodes from the solid and fluid sides match each other. The nondimensional time step  $\tau = \frac{u_0 \Delta t}{L}$  and nondimensional stiffness  $\kappa = \sqrt{\frac{E h_s}{2 \rho r_0 v_0(0)^2}}$  are chosen as  $\tau = 0.05$  and  $\kappa = 21$ . Note that the authors in [20] showed that the coupling with values of  $\tau$  and  $\kappa$  of this order is strong and standard Gauss-Seidel iterations quickly fail to converge. FSI subiterations tolerance used here is  $\delta = 10^{-4}$  and the reuse number of IQN-ILS used is  $q = 2$ .

In order to train the ROM model, a FOM-FOM computation is done on a single inlet velocity case corresponding to  $\boldsymbol{\mu}_1$  until  $T = 35$  s. The offline computations of the ROM models are then performed

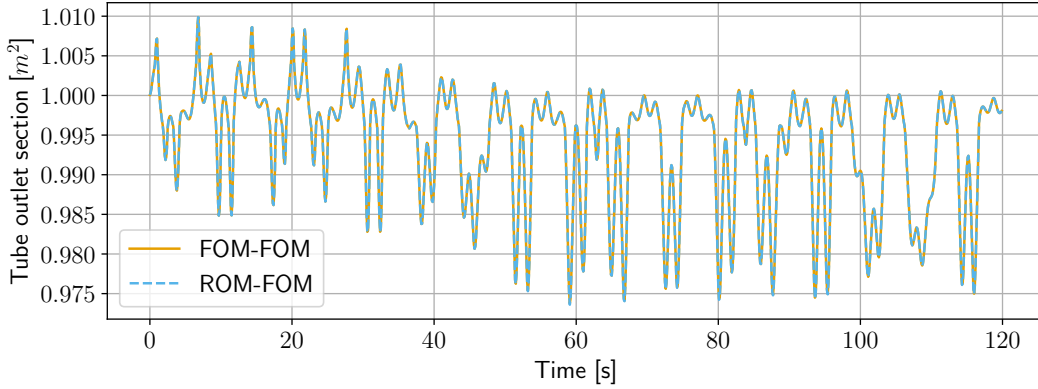


Figure 7: Example 1 - Time-evolution of the outlet tube section, comparing the ROM-FOM and FOM-FOM solutions at  $\boldsymbol{\mu} = \boldsymbol{\mu}_2$ . The time-series are quasi-superposed 'at the eye norm'

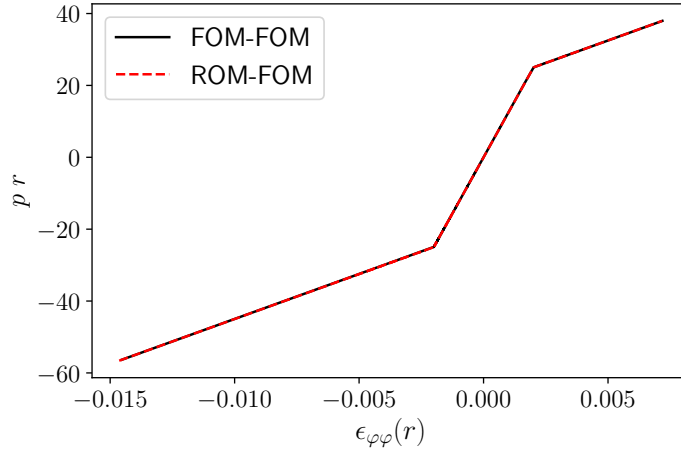


Figure 8: From the results of the ROM-FOM problem, the pressure  $p$  and tube radius  $r$  are retrieved from the results and  $\epsilon_{\varphi\varphi}$  is computed as  $\frac{\sqrt{\pi}r-r_0}{r_0}$ . We can see the nonlinear law used in (52) is well reproduced. Case of  $\boldsymbol{\mu} = \boldsymbol{\mu}_2$

on the available results, giving data snapshots of size  $m = 1759$  (see (26)).

The latent dimension of the displacement POD subspace is  $r_u = 4$  and  $r_f = 10$  for the pressure field. Regarding the regressor, a thin plate spline kernel RBF [68] is used with

$$\phi(\mathbf{x}) = \mathbf{x}^2 \log(\mathbf{x}) \quad (55)$$

for both  $\mathcal{I}_S(\cdot)$  and  $\mathcal{I}_F(\cdot)$ . For the local iterations the convergence criterion is  $\delta_r = 0.02$ , the iteration-frequency of the model update is  $Z = 200$  and  $p = 1640$  is chosen as the batch size.

The performance of the ROMs is evaluated on the future time prediction of the simulation with  $\boldsymbol{\mu} = \boldsymbol{\mu}_1$ , i.e for  $t \in [35s, 120s]$ . In addition, we also test the ROMs and predictors on the unseen parameter value  $\boldsymbol{\mu}_2$ . First, we check the accuracy of the solid ROM prediction in the ROM-FOM coupling: we show in Figure 7 the outlet section evolution in time comparing the FOM-FOM and the ROM-FOM solutions. We also demonstrate the solid nonlinearity well predicted by the ROM in Figure 8, which is expected since the region of deformations reached during the prediction was well present in the training data.

We show in figure 9 the total number of iterations performed using the different predictors for  $\boldsymbol{\mu}_1$ , and figure 10 for  $\boldsymbol{\mu}_2$ .

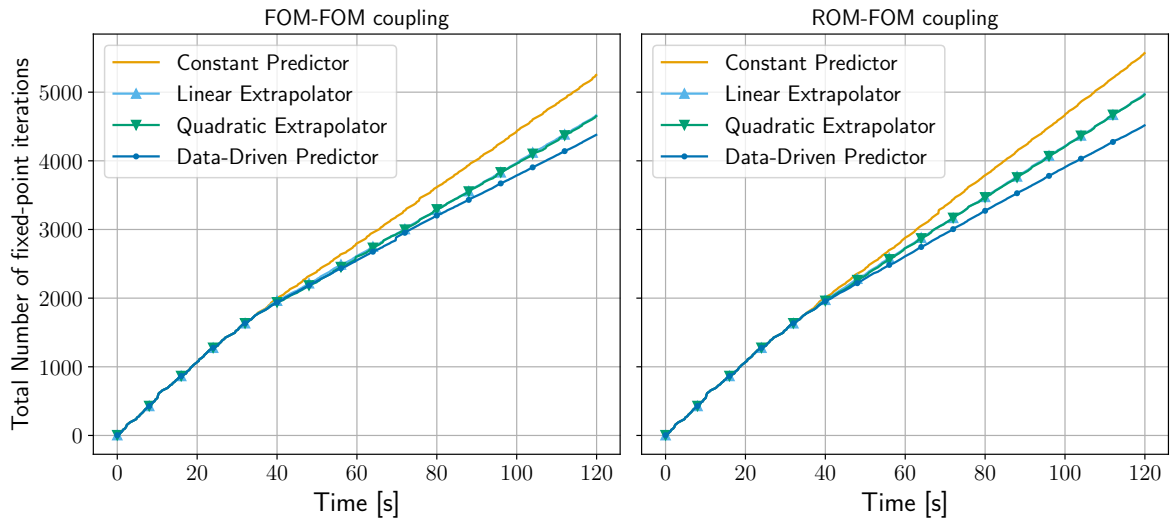


Figure 9: Example 1 - Comparison of total iterations needed for the simulation in a time-prediction regime using  $\boldsymbol{\mu} = \boldsymbol{\mu}_1 = [2, 6]^T$

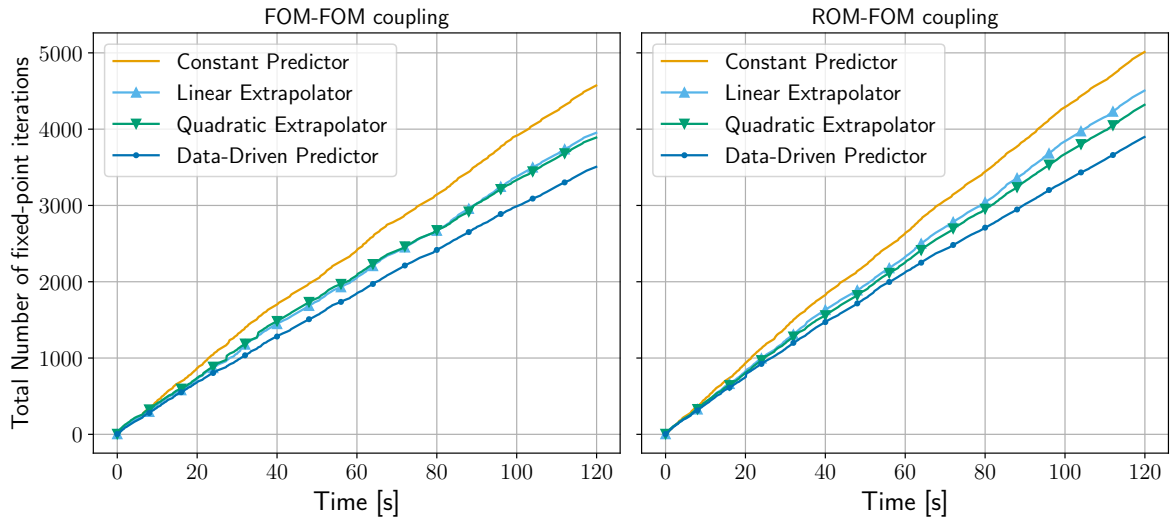


Figure 10: Example 1 - Comparison of total fixed-point iterations needed for the simulation using the unseen parameter  $\boldsymbol{\mu} = \boldsymbol{\mu}_2 = [0.9, 4]^T$  (Time-parameter prediction)



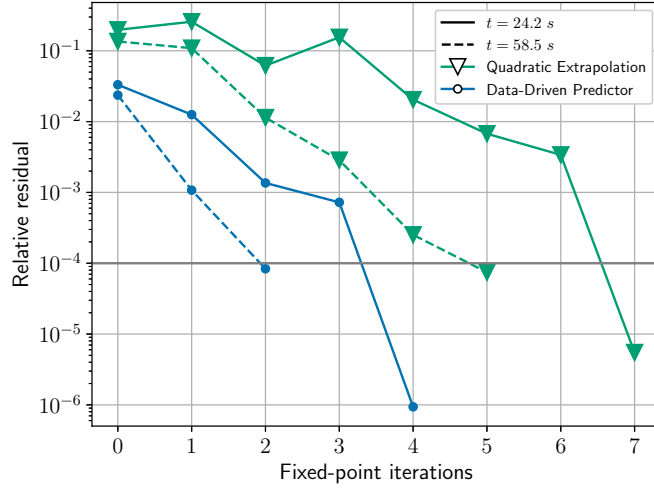


Figure 11: Fixed-point convergence rate using a data-driven predictor and a quadratic extrapolation at two time steps,  $t = 24.2 \text{ s}$  and  $t = 58.5 \text{ s}$ , example 1. Convergence relative tolerance  $\delta = 10^{-4}$ .

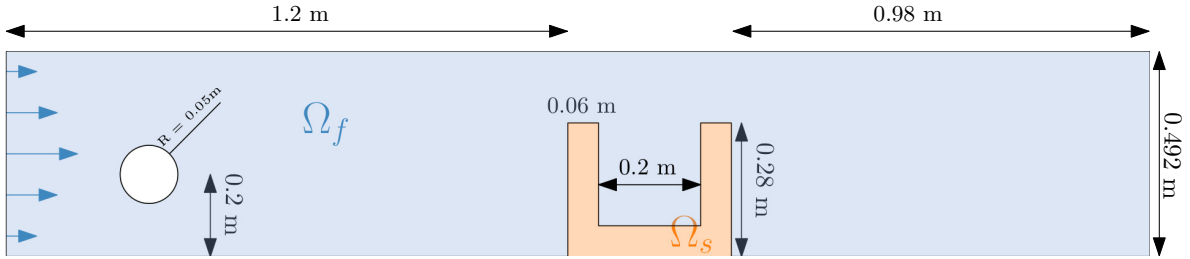


Figure 12: Example 2 - test case schematic explanation and dimensions.

The results clearly show that the new data-driven predictor provides the best efficiency in terms of number of iterations. We can see that using the ROM-FOM with constant predictors (right figures) leads to additional iterations due to the inaccuracy of the solid ROM. This effect is no longer observed when using the data-driven predictor. Moreover, while the use of linear and quadratic predictions result in fewer iterations than the constant predictor case, they are outperformed by the use of data-driven-based initial guesses. In Figure 11 and for two time steps, we show how the data-driven initial prompts a faster rate of convergence, as we can see that the data-driven predictor results in a first iteration with a much lower residual than with a quadratic extrapolation.

#### 4.2. Example 2: Hyperelastic flaps in a channel behind a cylinder wake

In this section, we consider the problem first introduced in [43], and illustrated in figure 12, where an incompressible flow in a 2D channel faces a massless elastic body with two mounted flaps behind a rigid cylinder.

For the fluid, the Navier-Stokes equations read

$$\begin{cases} \rho_f \frac{\partial \mathbf{v}}{\partial t} \Big|_{\bar{\mathcal{A}}} + \rho_f [(\mathbf{v} - \mathbf{w}) \cdot \nabla] \mathbf{v} + \nabla p - 2 \operatorname{div}(\mu_f \mathbf{D}(\mathbf{v})) = 0 & \text{in } \Omega_f(t) \\ \nabla \cdot \mathbf{v} = 0 & \text{in } \Omega_f(t) \end{cases} \quad (56)$$

with  $p$  the fluid pressure,  $\rho_f$  the fluid density,  $\mu_f$  the fluid dynamic viscosity and  $\mathbf{D}(\mathbf{v})$  is the fluid strain rate tensor. The fluid equations are described on a moving domain (using the ALE moving

frame)  $\Omega_f(t)$ . The notation  $\tilde{\mathcal{A}}$  represents the ALE mapping from the reference domain (the  $t = 0$  configuration) to the computational domain and  $\mathbf{w}$  is the ALE velocity.

In this case, we have  $\rho_f = 1000 \text{ kg/m}^3$ ,  $\mu_f = 0.001 \text{ m}^2/\text{s}$  and a fully developed Poiseuille inlet flow is applied, with a maximum velocity of  $v_{max} = 2.5 \text{ m/s}$  starting from  $v = 0 \text{ m/s}$  at  $t = 0 \text{ s}$  and increased by a sinusoidal ramp until reaching  $v_{max}$  at  $t = 1 \text{ s}$ . This corresponds to a Reynolds number of  $Re = 250$ , based on  $v_{max}$  and the cylinder diameter. The boundary condition at the top and bottom walls is a no slip condition, and a zero pressure on the right boundary.

For the solid subproblem, the equilibrium and constitutive equations for a static hyperelastic solid are:

$$\begin{cases} \nabla_{\mathbf{X}} \mathbf{P} = \mathbf{0} & \text{in } \Omega_s \\ \mathbf{P} = \frac{\partial W}{\partial \mathbf{F}_s} \\ \mathbf{u} = \mathbf{0} & \text{in } \Gamma_{D,s} \end{cases} . \quad (57)$$

The equations are written in the Lagrangian frame with  $\nabla_{\mathbf{X}}$  the gradient operator in the original configuration,  $\mathbf{P}$  is the first Piola-Kirchoff stress tensor (PK1) and  $\Gamma_{D,s}$  is the Dirichlet boundary. The material model is described in the stored energy density function  $W$ , here using the hyperelastic Neo-Hookean model:

$$W(\mathbf{F}_s) = \frac{\lambda_s}{2} (\ln(J))^2 - \mu_s \ln(J) + \frac{\mu_s}{2} (\text{trace}(\mathbf{C}_s) - 3) \quad (58)$$

where  $J = \det(\mathbf{F}_s)$  is the determinant of the deformation gradient tensor  $\mathbf{F}_s$ ,  $\mathbf{C}_s = \mathbf{F}_s^T \mathbf{F}_s$  is the right Cauchy Green deformation tensor,  $\mu_s = \frac{E}{2(1 + \nu_s)}$  and  $\lambda_s = \frac{E\nu_s}{(1 + \nu_s)(1 - 2\nu_s)}$ . In this example,  $E = 10 \times 10^6 \text{ Pa}$  and  $\nu_s = 0.3$ . The coupling conditions (1) and (2) are imposed on the FSI interface.

The fluid problem is discretized using 5440 variational multiscale (VMS) finite elements [69] and 1640 quadrilateral plane strain finite elements are used for the structural problem, with 8 ( $X$  and  $Y$ ) displacement degrees of freedom at element nodes, making a total of  $N_S = 3610$  solid dofs. `KratosMutiphysics` [67] was used as the finite elements code for both problems, using the modules `FluidDynamicsApplication` and `StructuralMechanicsApplication` as separate solvers in a partitioned coupling. The fluid time step is  $\Delta t = 8 \times 10^{-3} \text{ s}$  and the second-order "Bossak" time integration scheme is used [70]. The interface grid has matching nodes from the solid and fluid sides and consists of 265 nodes at the interface, giving  $N = 530$ . The relative convergence tolerance used is  $\delta = 0.005$  and IQN-ILS with reuse  $q = 3$  is used as the QN acceleration scheme. Figure 13 shows the ROM-FOM solution at  $t = 7.224 \text{ s}$ .

We define the parameter as the Reynolds number  $\mu = Re$  and we use the results from simulations of three points :  $\mu \in \{180, 205, 250\}$  solved for  $t \in [0, 3.6 \text{ s}]$ , generating a total of  $m = 7410$  snapshots. The evaluation of the new ROM predictor will be then done on an unseen parameter  $\mu = 192$  and on a larger simulation time  $t \in [0, 8 \text{ s}]$ .

For the dimensionality reduction,  $r_u = 9$  displacement modes and  $r_f = 45$  force modes are used. For the *SROM* regressor  $\mathcal{I}_S(\cdot)$ , a  $2^{nd}$  order polynomial regression is used with an  $L_1$  regularization term as described in (31). The *FROM* regressor  $\mathcal{I}_F(\cdot)$  is chosen here as an RBF function with a cubic kernel

$$\phi(\mathbf{x}) = \mathbf{x}^3. \quad (59)$$

For the local iterations convergence, the tolerance is  $\delta_r = 2\%$ , the *FROM* is retrained every  $Z = 200$  iterations, and the reduced data batch size  $p$  is 6900.

The solid ROM gives an accurate model prediction when coupled with the fluid FOM as shown in Figure 14 where the left tip x-displacement is plotted and compared using the ROM-FOM and FOM-FOM coupling. The displacement values used in the offline training at different  $Re$  numbers are

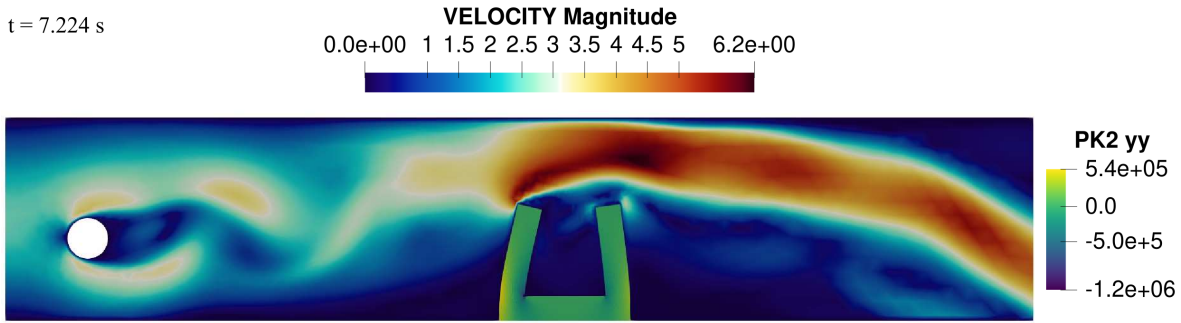


Figure 13: ROM-FOM solution at  $t = 7.224s$ : The velocity magnitude on the fluid domain and the PK2 stress on the solid domain.

also shown in the same figure. The accuracy of the displacement field is reported as the relative error

$$e(t) = \frac{\|\mathbf{d}(t) - \mathbf{d}(t)_{FOM}\|_2}{\langle \|\mathbf{d}(t)_{FOM}\|_2 \rangle} \quad (60)$$

with  $\langle \cdot \rangle$  representing the time-average. We can see in the figure it remains under 7% even for such a long simulation time. In figure 15, we show the strain prediction through the Green-Lagrange strain tensor  $\mathbf{E} = 1/2(\mathbf{C}_s - \mathbf{I})$  and the overall nonlinear material behavior using the response of one element as an example, showing the significant accuracy of the SROM prediction. This reaffirms that the SROM is indeed able to capture the nonlinearity.

The accumulated number of fixed-point iterations over the simulation time is reported in Figure 16 for the different predictors. Once again, the novel data-driven predictor produces the least number of FSI iterations. More details about the average number of iterations are given in Table 1. In addition, in the second column of Table 1, we also demonstrate the efficiency in terms of the CPU time spent when using the different predictors. As the evaluation of the *FROM* and the local *SROM* are inexpensive at the start of each time step, and keeping in mind the negligible cost of the *FROM* update, the data-driven predictor does indeed result in a smaller overall CPU time. Using the solid ROM to replace the FOM allows even greater speedup, especially since, in this problem, the solid FOM solver takes nearly twice as CPU time as the fluid FOM solver. We note that, taking advantage of the fluid FOM, the presented approach performs very well, in terms of the SROM accuracy, and the faster fixed-point convergence in spite of the complex dynamics of this FSI problem. The on-the fly update of the FROM also prevents inaccurate predictions when these unseen complex dynamics take place.

In order to highlight the importance of using the accelerated forces  $\mathbf{f}$  and not the fluid output  $\tilde{\mathbf{f}}$  as the SROM input (See the section 2.1 above), we show in Figure 17 the phase space composed of the first 2 components of their latent representations  $\tilde{\mathbf{f}}_r$  and  $\mathbf{f}_r$ . The values shown correspond to the forces obtained during the training simulations. We can see that the variance is much bigger with  $\tilde{\mathbf{f}}_r$  and this poses difficulties on the training of SROM. Bigger data variance also means that a bigger range of the solid nonlinearity is included in the data, since bigger force amplitudes are reached with  $\tilde{\mathbf{f}}_r$  than with  $\mathbf{f}_r$ .

As a last indication, the effect of the FROM update is first assessed by looking at the difference of total number of fixed-point iterations achieved with and without the update. For the unseen parameters  $Re = 192$ , the benefits of the update is clearly shown in Figure 18 left where the number of fixed-point iterations is always diminished. More importantly, when considering a parameter  $Re = 139$  outside of the Reynolds number training interval  $[180, 250]$ . Figure 18 center shows that hundreds less convergence iterations are needed overall, *i.e* for all the time steps. Second, we wish to demonstrate the evolution of the model error along the update iterations. Offline data are stored from the last 1000 iterations' results of the FOM-FOM simulation on the unseen parameter  $Re = 139$ . The FROM is

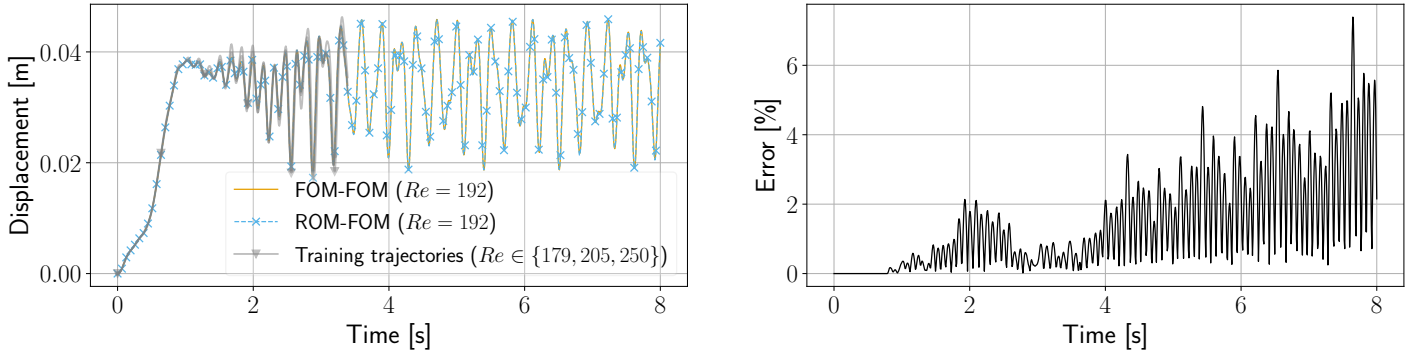
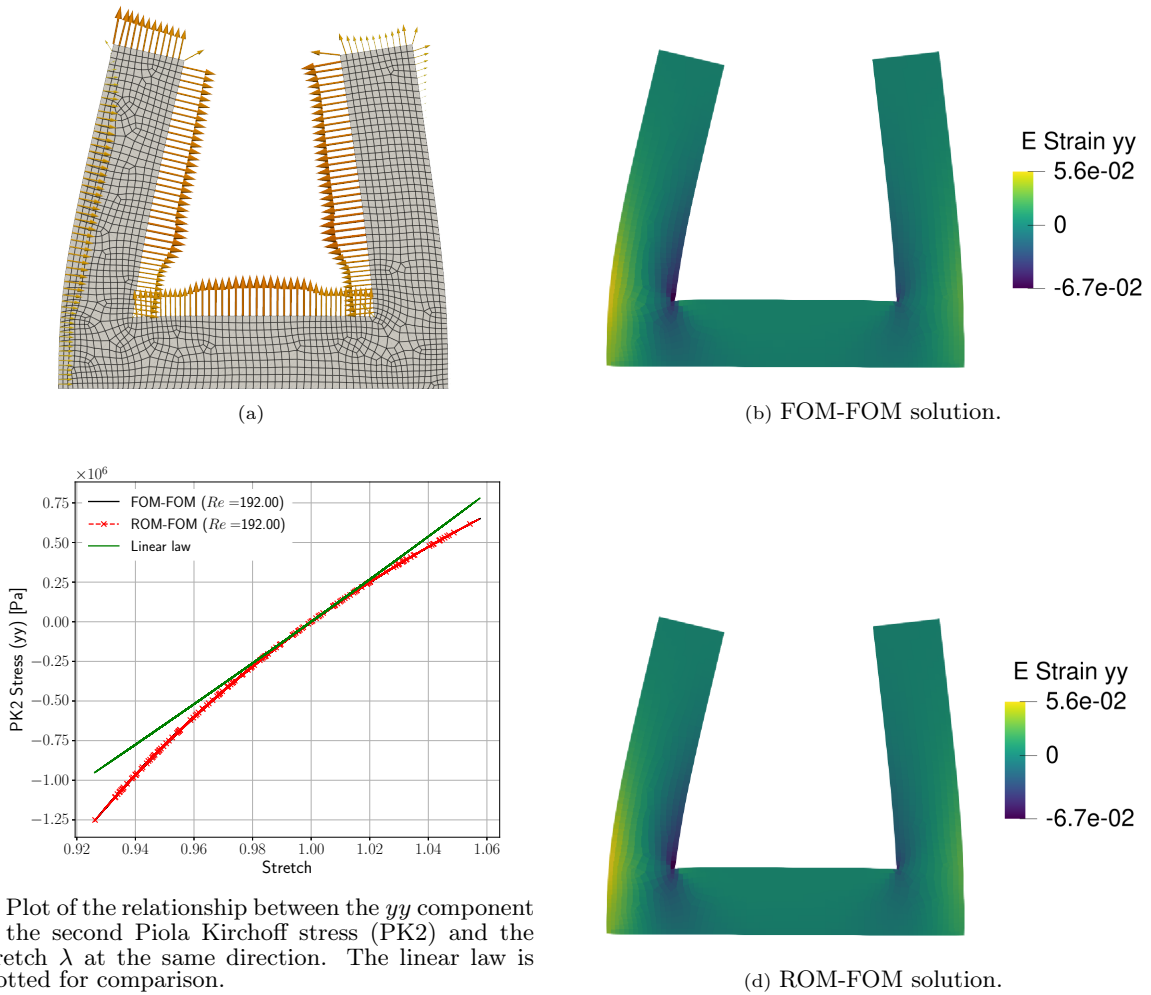


Figure 14: Example 2 - **(Left)** The left tip displacement evolution of the solid comparing the ROM-FOM and the FOM-FOM solution in time, while the training trajectories (in the different Reynolds numbers) of the same quantity are shown in grey. **(Right)** The relative displacement error of the SRM at  $Re = 192$  is computed using (60) and shown as a function of the simulation time.



(c) Plot of the relationship between the  $yy$  component of the second Piola Kirchoff stress (PK2) and the stretch  $\lambda$  at the same direction. The linear law is plotted for comparison.

Figure 15: The SRM prediction evaluated on the solid nonlinear behaviour in example 2. In (a) we see the loads applied on the deformed solid at  $t = 4.7s$ , and (b) and (d) show the  $yy$  component of the Green Lagrange strain at the same time step, comparing the ROM-FOM and FOM-FOM solutions. In (c) the nonlinear behaviour of the Neo-Hook solid is represented using the results of the FOM-FOM and ROM-FOM.

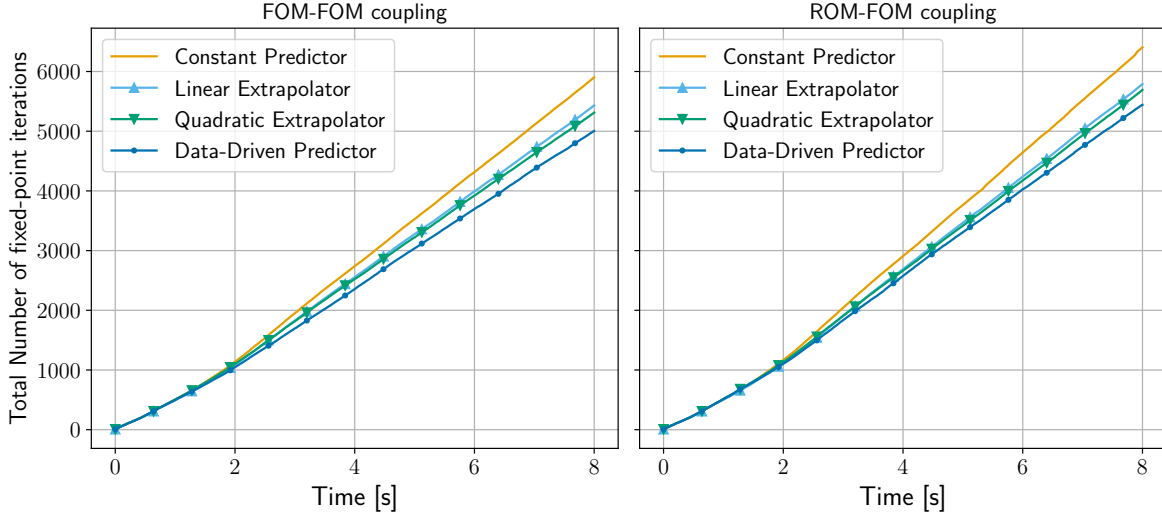


Figure 16: Example 2 - Comparison of total fixed-point iterations needed for the simulation at an unseen  $Re = 192$ , using different predictors.

	FOM-FOM		ROM-FOM	
	Avg. iterations	CPU Time	Avg. iterations	CPU Time
Data-Driven	<b>5.11</b>	2.7	5.6	<b>1.0</b>
Constant	6.12	3.43	6.67	1.13
Linear	5.59	<b>3.16</b>	5.99	1.24
Quadratic	<b>5.46</b>	3.35	5.88	1.01

Table 1: Example 2 - Average number of iterations and the total CPU time of the simulation at the unseen  $Re$  number using the FOM-FOM and ROM-FOM coupling.

then evaluated on this test data and incrementally updated on each  $Z$ -sized batches of offline results from the beginning of the simulations onward. The error is computed after each FROM update as the mean relative maximum error and reported on Figure 18 right. We can clearly see the decrease of the FROM error along the update increments, highlighting the ability of the model to leverage its updates for extrapolation.

#### 4.3. Example 3 : 3D hyperelastic incompressible flaps:

To show the performance of our methodology in larger scale problems, we consider a 3D extension of the previous problem. The figure 19 gives a brief description. The boundary faces corresponding to the top, bottom, back and front walls of the geometry in the left of Figure 19 will be called henceforward  $y+$ ,  $y-$ ,  $z-$  and  $z+$  respectively. To simplify the configuration, no cylinder is present in this problem but a fully developed pulsatile inlet flow is applied with a signal composed of two frequencies ( $f_1, f_2$ ) as illustrated in figure 20 and expressed as:

$$\begin{cases} v_{|x=0}(y, t) = \frac{1}{2}(1 - \cos(\pi t))g(y) & \forall t \in [0, 1] \\ v_{|x=0}(y, t) = \frac{1}{16}(16 + (1 - \cos(2\pi(t - 1))))g(y) & \forall t \in [1, 1.5] \\ v_{|x=0}(y, t) = \frac{1}{16}(16 + (\cos(f_1\pi(t - 1.5))) + (\cos(f_2\pi(t - 1.5)))) & \forall t \in [1.5, 6.6] \\ g(y) = 11.8y(0.492 - y) \end{cases} \quad (61)$$

with  $\mu_1 = (f_1, f_2) = (4 \text{ Hz}, 5 \text{ Hz})$  the parameter values used for training and  $\mu_2 = (f_1, f_2) = (2 \text{ Hz}, 3 \text{ Hz})$  for testing. Slip conditions are used on the  $z-$  and  $z+$  faces, no-slip conditions are

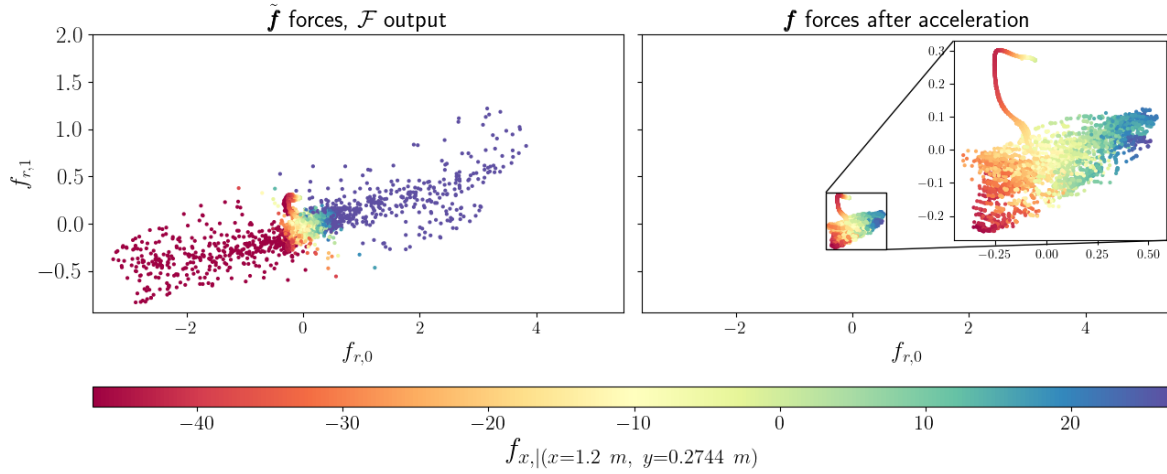


Figure 17: Comparison between the fluid solver output  $\tilde{\mathbf{f}}$  (Left) and the accelerated forces  $\mathbf{f}$  (Right) using the first two components  $f_{r,0}$  and  $f_{r,1}$  of their reduced representations  $\Phi_f^T \tilde{\mathbf{f}}$  and  $\Phi_f^T \mathbf{f}$ . The colormap corresponds to the amplitude of the X-force at a node close to the left tip.

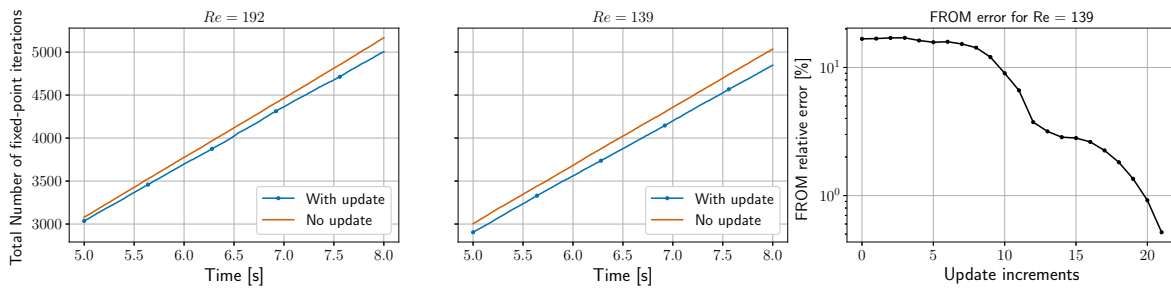


Figure 18: Effect of the FROM update on the accuracy for the  $2^{nd}$  example. The total number of global iterations achieved when using an FROM with and without update, evaluated on unseen Reynolds numbers : (Left)  $Re = 192$ . (Center)  $Re = 139$ . In (Right), the test error of the FROM is plotted along the model update increments for  $Re = 139$ .

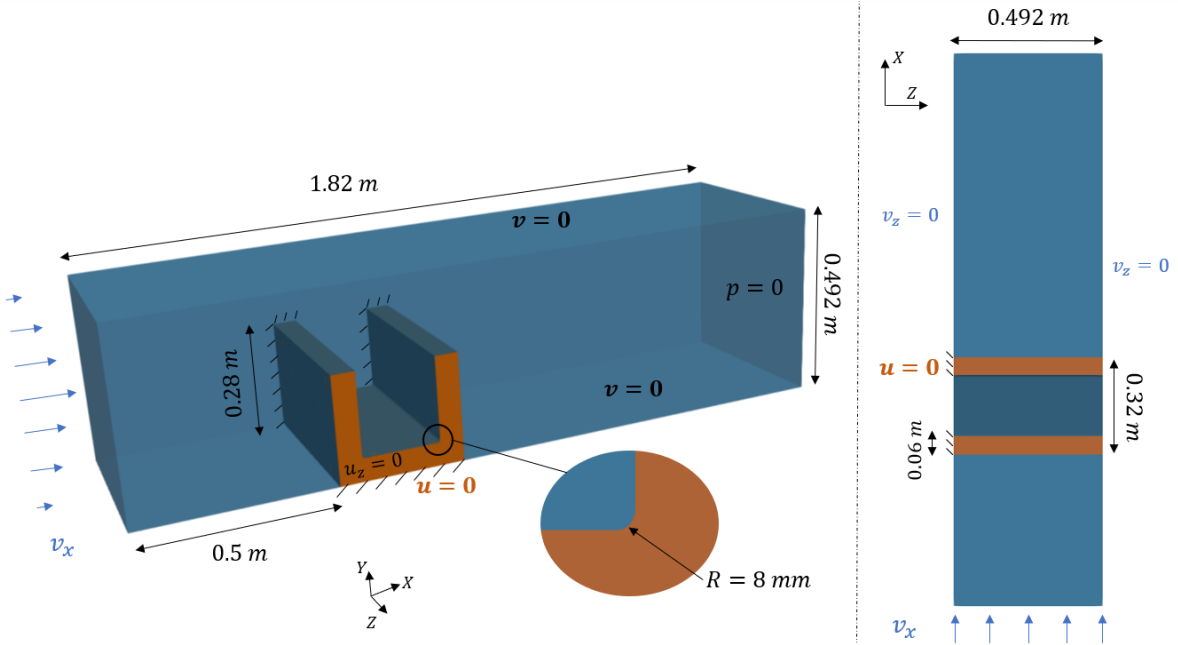


Figure 19: Example 3 - Test case schematic explanation and dimensions.

imposed on the  $y-$  and  $y+$  faces and a zero pressure is imposed on the outlet. The discretization uses 437039 elements with 84988 nodes. This flow setting corresponds to a Reynolds number  $Re = 225$  based on the maximum inlet velocity and the length of the solid flap.

For the solid material, a nearly-incompressible Neo-Hookean material is used with  $\nu_s = 0.485$ , and  $\mathbf{u}-\theta$  mixed tetrahedral elements are used where  $\theta$  represents the Jacobian determinant  $\theta = J \approx 1$ . The mixed formulation is stabilized based on the VMS approach (See [71] for more details on the element used here). As a consequence, two solid ROMs are constructed: an SRM in a similar manner to the previous cases, and a second  $\theta$ -ROM for the  $\theta$  field. In fact, in order to compute the strain and stress *a posteriori*, an accurately computed  $\theta$  field must be available, and in our ROM-FOM approach this is done only after convergence of each time step, since only the displacement at the interface is needed at the other fixed-point iterations. The solid mesh consists of 65400 tetrahedral elements with 15338 nodes. In order to ensure a valid discretization, a mesh independence study was done on the fluid and the solid domains separately. Details on this study are reported on the Appendix B. The second-order Bossak time integration is used by the fluid solver with a time step of  $\Delta t = 0.01$  s. Similarly to the previous test case, matching interface grids are used, consisting of 7575 nodes at the interface, giving  $N = 22725$ , and IQN-ILS with reuse  $q = 3$  is used with a fixed-point tolerance of  $\delta = 5 \times 10^{-4}$ . In Figure 22c, we show the solution of the problem at  $t = 3.37$  s.

The solid regressor  $\mathcal{I}_S(\cdot)$  used here is a thin plate spline RBF interpolator, and  $\mathcal{I}_F(\cdot)$  is a ridge regression with a regularization parameter  $\lambda = 1 \times 10^{-5}$ . For dimensionality reduction,  $r_u = 12$  displacement modes and  $r_f = 280$  forces modes are kept. For the data-driven predictor, the model update is computed after each  $Z = 240$  fixed-point iterations and the batch of the reduced snapshots is of size  $p = 6900$ . The tolerance used here is  $\delta_r = 1 \times 10^{-4}$ . The FOM-FOM simulation using  $\mu_1$  and for all the time steps until  $t = 6.6$  s generates  $m = 2962$  snapshots for the training of SRM and FROM.

The SRM shows a significant accuracy on the displacement prediction, as seen in Figure 21, where the displacement evolution of the left tip and right tip of the  $z+$  face are shown while comparing the FOM-FOM and ROM-FOM solutions. The prediction of the stress field, namely the PK2 stress is

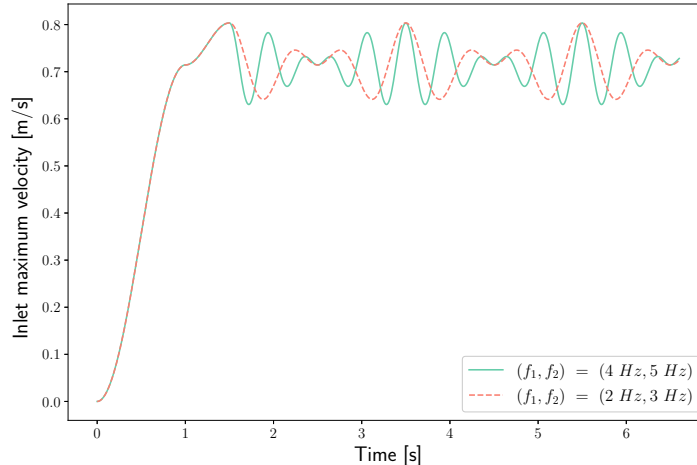


Figure 20: Inlet velocity at the center of the inlet section.

	FOM-FOM		ROM-FOM	
	Avg. iterations	CPU Time	Avg. iterations	CPU Time
Data-Driven	<b>3.75</b>	1.06	4.0	<b>1.0</b>
Constant	<b>4.24</b>	<b>1.20</b>	4.3	1.06
Linear	4.5	1.28	4.97	1.25
Quadratic	4.86	1.36	5.49	1.34

Table 2: Example 3 - Average number of iterations and the total CPU time of the simulation at the unseen  $\mu$  parameter using the FOM-FOM and ROM-FOM coupling.

also obtained with high accuracy as seen in Figures 22a and 22b where the SROM and  $\theta$ ROM accurate output clearly lead to accurate stress predictions as well.

The addition of the FROM and the local iterations for the data-driven initial guess results in much less total fixed-point iterations. This is clearly seen in Figure 23, where the data-driven predictor outperforms the classical approaches. The average number of fixed-point iterations and the total CPU time are also reported in Table 2. Remarkably, the linear and quadratic extrapolators result in slower convergence than the constant extrapolation, while the data-driven predictor ensures a faster convergence of the fixed-point FSI problem, showing the improved robustness of the data-driven approach for predicting the initial guess.

## 5. Conclusions

In this work, a novel data-driven predictor for the acceleration of convergence of unsteady partitioned fluid-structure interaction coupling has been proposed. This predictor provides an enhanced initial guess for the FSI fixed-point problem at each time step. It is obtained by resolving a reduced fixed-point problem that can be solved at the beginning of the time step for a small fraction of the computational time of the regular FSI problem. This is achieved using two reduced order models for the solid and the fluid problems by approximating the force-to-displacement and the displacement-to-force relationships respectively. The two models are then strongly coupled to predict the initial guess when this reduced fixed-point converges. Each reduced order model is constructed from three components: An encoder, a regressor and a decoder. The encoder-decoder uses the POD and quadratic manifolds for the dimensionality reduction and the regressors use either RBF functions or polynomial regression. The data-driven nature of this predictor makes it more robust and efficient than the classical



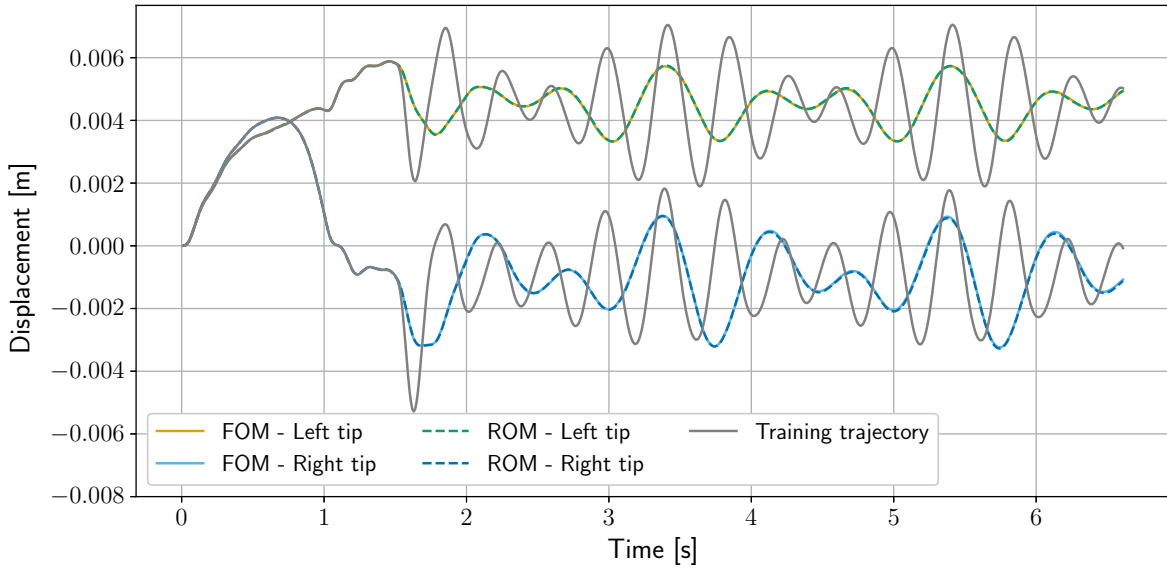


Figure 21: Example 3 - The left and right tip displacement evolution of the solid, comparing the ROM-FOM solution to the FOM-FOM solution.

approach, since it uses the information from the results of the high-fidelity solver, instead of using finite-differences from the last few time steps. Moreover, the regression model of the fluid ROM is updated online using the high-fidelity forces from the fluid FOMs, enriching further the ROM from the latest available HF data. Overall, the proposed methodology leverages physics-based insights from the high fidelity fluid solver, thus establishing a physics-aware machine learning predictor. This enables the use of the predictor in extrapolating regions of the time-parameter space. This predictor can also be combined with the solid ROM presented in [43] for replacing the solid solver as a whole to predict the solution at an even cheaper computational cost. In the paper, and through three examples with strong FSI coupling and neglected solid inertia, we have demonstrated the performance of this novel predictor in achieving faster convergence of the fixed-point problem compared to classical extrapolators. In particular, we showed the significant computational gain that can be achieved with this predictor, even when applying it for unseen time and parameters, even in an extrapolation setting, and even for fairly complex dynamics of the FSI problem, all while keeping a very high accuracy of the ROM when replacing the solid FOM.

We showed that designing an FSI predictor with such a data-driven strategy makes it more robust for easing convergence than the classical extrapolators, since the data-driven ROM benefits from recent HF data more judiciously. The ROM update strategy presented in this work could be further enhanced in order to obtain more accurate and adapted ROMs: for example, instead of a straightforward retraining of the regressor component of the ROM only (as done in this work), an update strategy of the dimensionality reduction part (namely the POD bases update), could also be done using the online HF data. We believe that such a strategy will eventually lead to faster fixed-point convergence, since new force values at prediction time can lie outside the available POD subspace obtained from offline data. This adaptive encoder/decoder approach will be pursued in future researches.

## 6. Acknowledgements

This work has been funded by the ANR (Agence Nationale de la Recherche, France), Altair Engineering and Michelin under the project AHEAD.

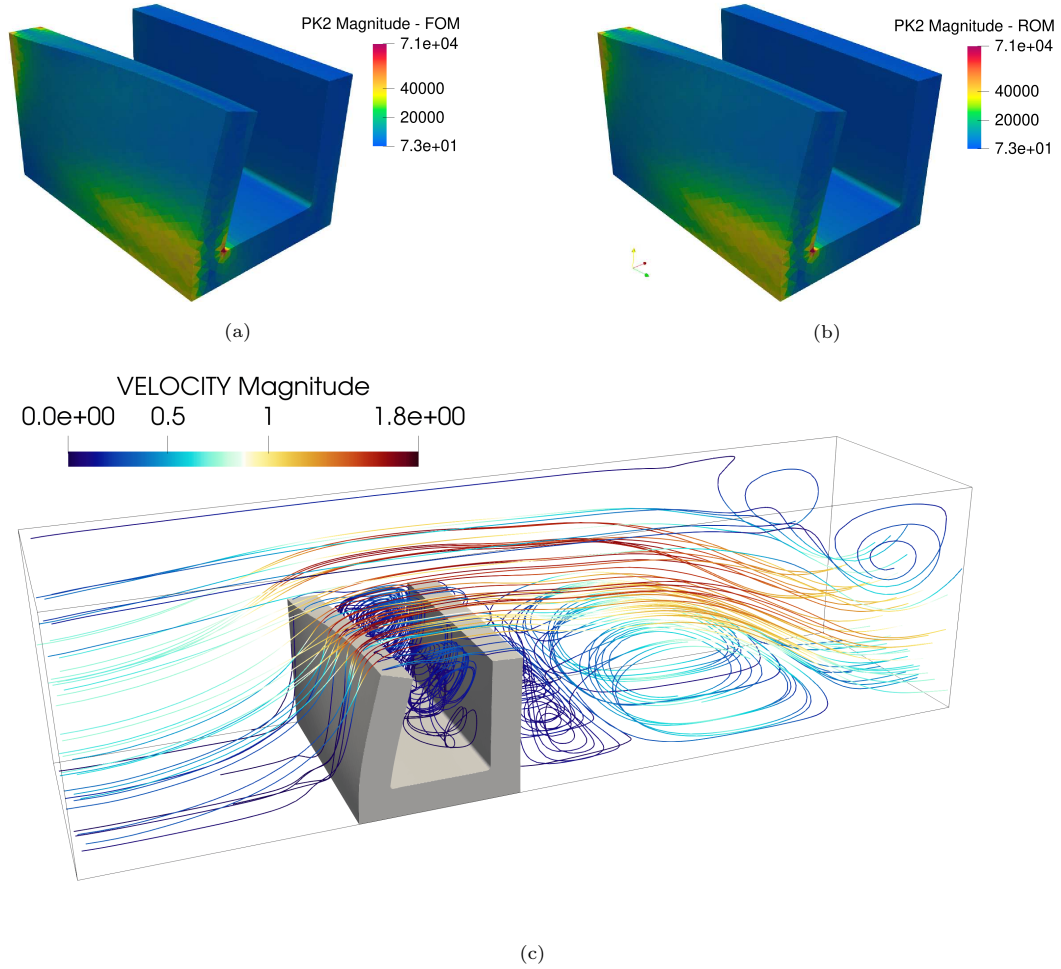


Figure 22: Evaluation of SROM prediction at  $t = 3.37s$  of the PK2 stress magnitude comparing (a) the FOM-FOM solution and (b) the ROM-FOM solution. In (c) the deformed solid and the velocity streamlines are shown at the same time step. The deformation is  $\times 10$  amplified

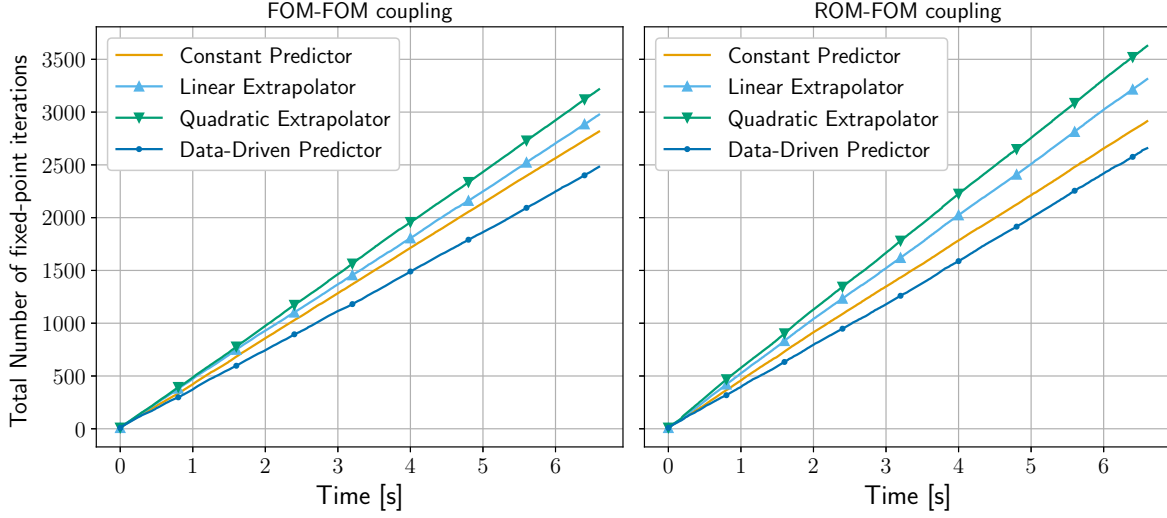


Figure 23: Example 3 - Comparison of total fixed-point iterations needed for the simulation at an unseen parameter  $\mu_2$ , using different predictors.

## Appendix A IQN-ILS Algorithm

---

**Algorithm 4:** Interface Quasi-Newton Inverse Least-Squares [32]

---

**Input:** Iteration  $k$   
Time step  $n$   
Force  $\tilde{\mathbf{f}}^k$   
Force residual  $\mathbf{r}^k$   
Reuse  $q$   
Relaxation  $w_0$   
**Result:**  $\Delta \mathbf{f}^k = \text{Acc}(\tilde{\mathbf{f}}^k, \mathbf{r}^k)$

**if**  $n == 1$  **and**  $k == 1$  **then**

- 1 |  $\Delta \mathbf{f}^k = w_0 \mathbf{r}^k$
- else**
- 2 |  $\mathbf{V} = [\Delta \mathbf{r}^k \ \dots \ \Delta \mathbf{r}^{k-d}]$  from the previous  $d$  fixed-point iterations (including past  $q$  time steps)
- 3 |  $\mathbf{W} = [\Delta \tilde{\mathbf{f}}^k \ \dots \ \Delta \tilde{\mathbf{f}}^{k-d}]$
- 4 | QR decomposition  $\mathbf{V} = \mathbf{Q}\mathbf{R}$
- 5 | QR filtering using e.g the approach in [34]
- 6 |  $\mathbf{b} = -\mathbf{Q}^T \mathbf{r}^k$
- 7 | Solve  $\mathbf{R}\mathbf{c} = \mathbf{b}$  for  $\mathbf{c}$  using back-substitution
- 8 |  $\Delta \mathbf{f}^k = \mathbf{W}\mathbf{c} + \mathbf{r}^k$

---

## Appendix B Mesh convergence study

To conduct the mesh convergence study efficiently, the solid and fluid problems are treated independently, i.e in a decoupled way, as the goal is merely to check the validity of the chosen mesh sizes even if the actual FSI problem is different, assuming the other conditions are close enough

A static surface load is applied on the left solid face, with a load amplitude comparable of that occurring during the FSI problem. The fluid problem for this study differs from the coupled problem

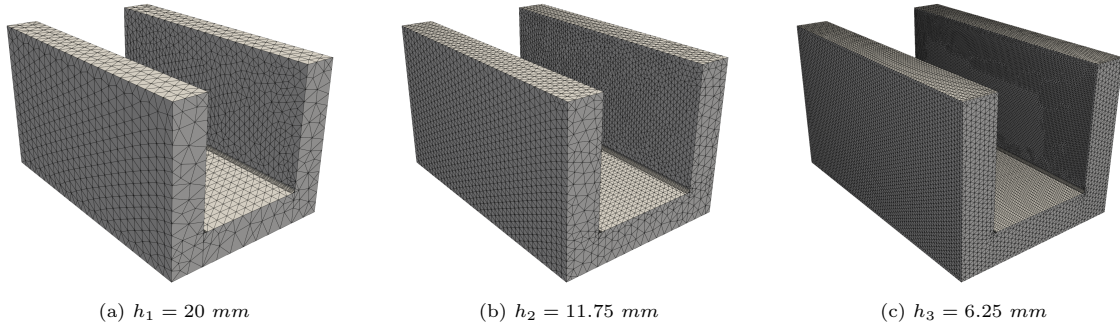


Figure 24: Solid meshes used in the mesh convergence study.

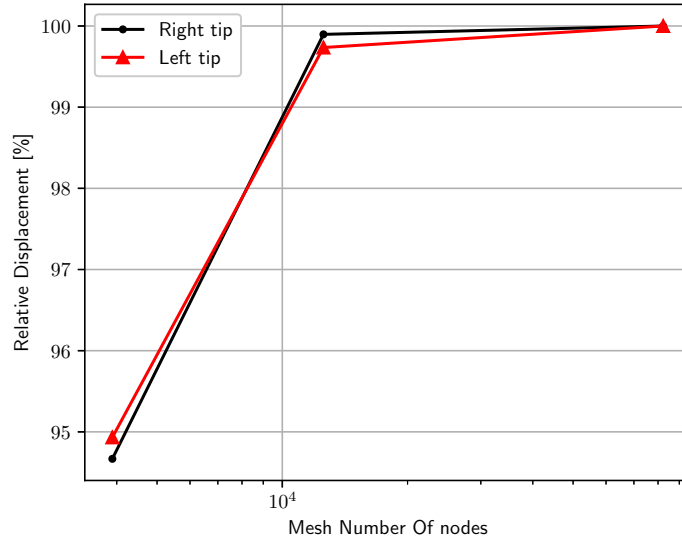


Figure 25: The displacement of the two tips of the solid plotted for each considered mesh. The displacement quantity is relative to that associated to the finer (reference) mesh.

in that only a constant inlet velocity is applied instead of a pulsatile inlet.

### B.1 Solid mesh:

A constant pressure of  $p = 1800 \text{ Pa}$  is applied on the left face and three different meshes with three different (average) mesh sizes on the boundaries are used:  $h_1 = 20 \text{ mm}$ ,  $h_3 = 6.25 \text{ mm}$  and  $h_2 = 11.75 \text{ mm}$ . The three meshes are illustrated in Figure 24. Note that the mesh size changes locally as it is reduced near the corners.

The most interesting quantity in the problem - the displacement - is reported on the right tip and left tip at the  $z+$  face for the three grids and plotted on Figure 25, from which we see that the  $h_2$  displacement falls under 1% from that of the reference value, assumed to be the one associated to the fine  $h_3$  mesh. From there, we conclude that the  $h_2$  grid is sufficient for the use of our FSI problem.

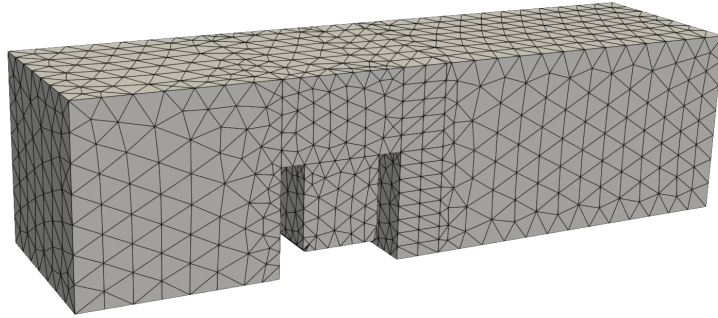
### B.2 Fluid mesh:

Similar to the FSI problem, a Poiseuille inlet flow velocity is applied until  $t = 6.6 \text{ s}$  for this study, with the difference here is that the inlet flow is constant in time whether a pulsatile flow is applied on the FSI example. The value of the inlet velocity corresponds to the maximum attained on the FSI problem inlet.

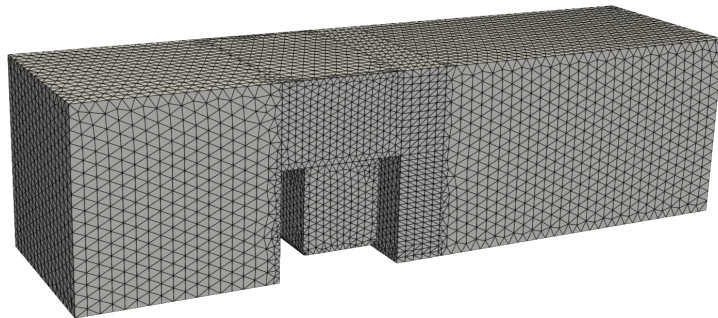
Three grids with varying average mesh sizes on the boundaries  $h_1 = 40 \text{ mm}$ ,  $h_2 = 18 \text{ mm}$  and  $h_3 = 12 \text{ mm}$ . Note again, that locally, the mesh size changes as it is refined near the FSI interface and in the region between the flaps. The three grids are shown in Figure 26.

The axial velocity profile  $120 \text{ mm}$  on the right FSI interface is shown in Figure 28 for the three different grids, using the mean of the time steps corresponding to the last  $4.6 \text{ s}$ . The streamlines of the velocity field at  $t = 6.6 \text{ s}$  are also shown on the mid section of the channel in Figure 27. In addition, we used the Grid Convergence Index (GCI) method from the ASME Journal of Fluids Engineering policy for mesh convergence [72]. The grid refinement factor between each two meshes is indeed greater than 1.3 and using the maximum velocity of the profiles reported in Figure 28 as the main quantity, the GCI index obtained is 4.9% which is we considered acceptable.

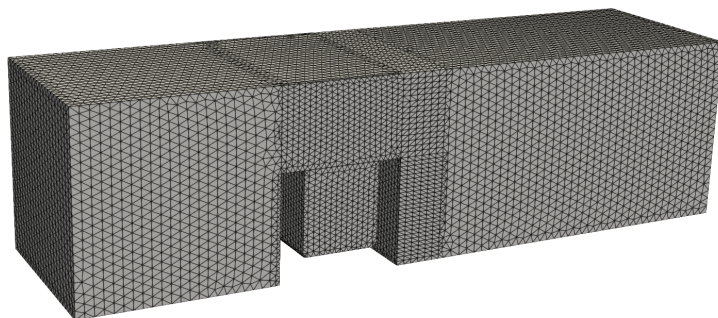
From all these results, we concluded the choice of using the grid with  $h_2$  for our FSI problem.



(a)  $h_1 = 40 \text{ mm}$



(b)  $h_2 = 18 \text{ mm}$



(c)  $h_3 = 12 \text{ mm}$

Figure 26: Outer surfaces of the fluid meshes used in the mesh convergence study.

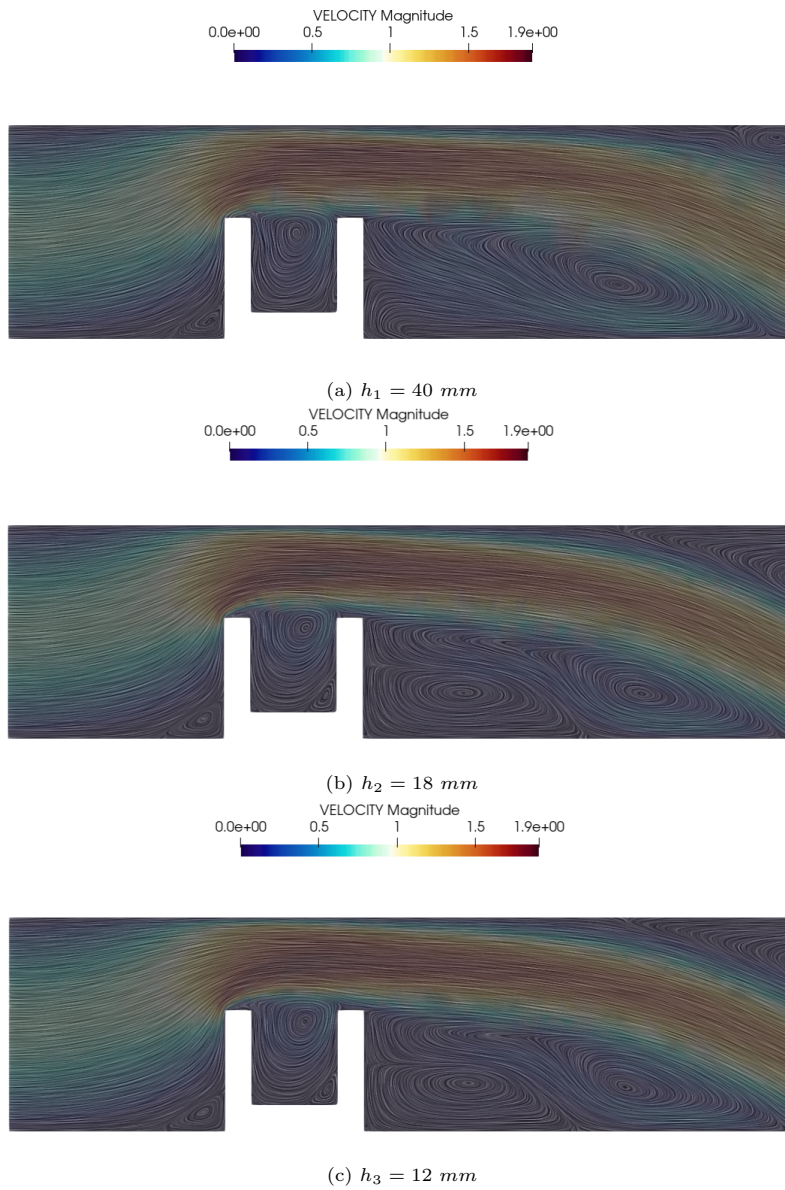


Figure 27: Streamlines of the velocity field on the mid section for the three considered grids at  $t = 6.6 \text{ s}$ .

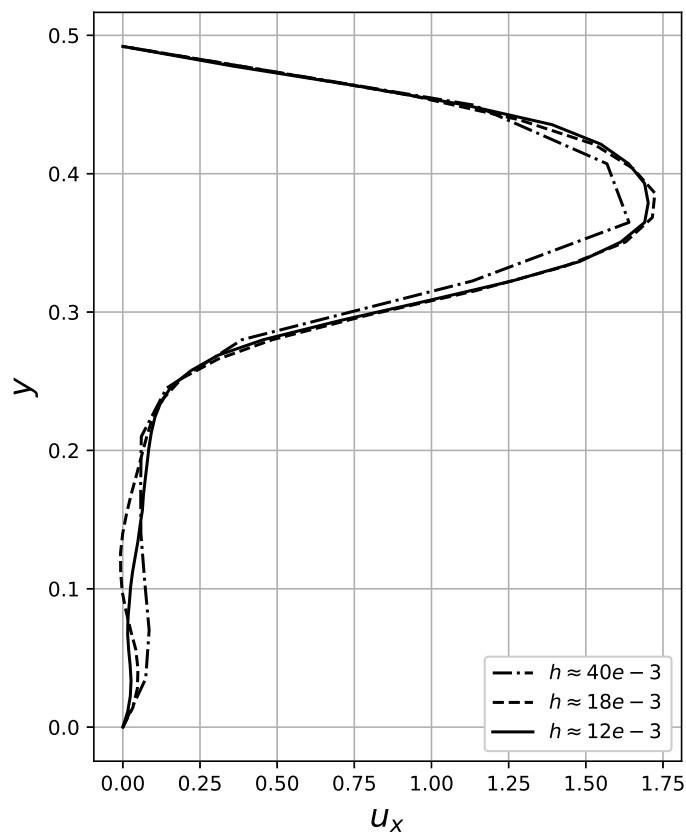


Figure 28: The axial velocity profile for the three considered grids.



## References

- [1] C. S. Peskin, The immersed boundary method, *Acta Numerica* 11 (2002) 479–517. doi:[10.1017/S0962492902000077](https://doi.org/10.1017/S0962492902000077).
- [2] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Monthly Notices of the Royal Astronomical Society* 181 (1977) 375–389. doi:[10.1093/mnras/181.3.375](https://doi.org/10.1093/mnras/181.3.375).
- [3] S. R. Idelsohn, J. Marti, A. Souto-Iglesias, E. Oñate, Interaction between an elastic structure and free-surface flows: experimental versus numerical comparisons using the PFEM, *Computational Mechanics* 43 (2008) 125–132. doi:[doi.org/10.1007/s00466-008-0245-7](https://doi.org/10.1007/s00466-008-0245-7).
- [4] J. Donea, S. Giuliani, J. P. Halleux, An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions, *Computer Methods in Applied Mechanics and Engineering* 33 (1982) 689–723. doi:[10.1016/0045-7825\(82\)90128-1](https://doi.org/10.1016/0045-7825(82)90128-1).
- [5] J. Hron, S. Turek, A monolithic fem/multigrid solver for an ale formulation of fluid-structure interaction with applications in biomechanics, in: H.-J. Bungartz, M. Schäfer (Eds.), *Fluid-Structure Interaction*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 146–170.
- [6] M. Heil, An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems, *Computer Methods in Applied Mechanics and Engineering* 193 (2004) 1–23. doi:[10.1016/j.cma.2003.09.006](https://doi.org/10.1016/j.cma.2003.09.006).
- [7] M. Fernández, M. Moubachir, - an exact block-newton algorithm for the solution of implicit time discretized coupled systems involved in fluid-structure interaction problems, in: K. Bathe (Ed.), *Computational Fluid and Solid Mechanics 2003*, Elsevier Science Ltd, Oxford, 2003, pp. 1337–1341. doi:[doi.org/10.1016/B978-008044046-0.50328-6](https://doi.org/10.1016/B978-008044046-0.50328-6).
- [8] T. E. Tezduyar, S. Sathe, Modelling of fluid-structure interactions with the space-time finite elements: Solution techniques, *International Journal for Numerical Methods in Fluids* 54 (2007) 855–900. doi:[doi.org/10.1002/flid.1430](https://doi.org/10.1002/flid.1430).
- [9] G. Fourey, C. Hermange, D. Le Touzé, G. Oger, An efficient FSI coupling strategy between smoothed particle hydrodynamics and finite element methods, *Computer Physics Communications* 217 (2017) 66–81. doi:[doi.org/10.1016/j.cpc.2017.04.005](https://doi.org/10.1016/j.cpc.2017.04.005).
- [10] S. Meduri, M. Cremonesi, U. Perego, O. Bettinotti, A. Kurkchubasche, V. Oancea, A partitioned fully explicit lagrangian finite element method for highly nonlinear fluid-structure interaction problems, *International Journal for Numerical Methods in Engineering* 113 (2018) 43–64. doi:[doi.org/10.1002/nme.5602](https://doi.org/10.1002/nme.5602).
- [11] M. Lesoinne, C. Farhat, Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems, *AIAA Journal* 36 (1998) 1754–1757. doi:[10.2514/2.7555](https://doi.org/10.2514/2.7555).
- [12] S. Piperno, Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2d inviscid aeroelastic simulations, *International Journal for Numerical Methods in Fluids* 25 (1997) 1207–1226. doi:[doi.org/10.1002/\(SICI\)1097-0363\(19971130\)25:10<1207::AID-FLD616>3.0.CO;2-R](https://doi.org/10.1002/(SICI)1097-0363(19971130)25:10<1207::AID-FLD616>3.0.CO;2-R).
- [13] C. Farhat, M. Lesoinne, Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems, *Computer Methods in Applied Mechanics and Engineering* 182 (2000) 499–515. doi:[doi.org/10.1016/S0045-7825\(99\)00206-6](https://doi.org/10.1016/S0045-7825(99)00206-6).

- [14] S. Piperno, C. Farhat, Partitioned procedures for the transient solution of coupled aeroelastic problems – part ii: energy transfer analysis and three-dimensional applications, *Computer Methods in Applied Mechanics and Engineering* 190 (2001) 3147–3170. doi:[doi:10.1016/S0045-7825\(00\)00386-8](https://doi.org/10.1016/S0045-7825(00)00386-8), advances in Computational Methods for Fluid-Structure Interaction.
- [15] C. A. Felippa, K. C. Park, C. Farhat, Partitioned analysis of coupled mechanical systems, *Computer Methods in Applied Mechanics and Engineering* 190 (2001) 3247–3270. doi:[10.1016/S0045-7825\(00\)00391-1](https://doi.org/10.1016/S0045-7825(00)00391-1).
- [16] C. Farhat, K. G. van der Zee, P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, *Computer Methods in Applied Mechanics and Engineering* 195 (2006) 1973–2001. doi:[doi:10.1016/j.cma.2004.11.031](https://doi.org/10.1016/j.cma.2004.11.031), fluid-Structure Interaction.
- [17] P. Causin, J. F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4506–4527. doi:[10.1016/j.cma.2004.12.005](https://doi.org/10.1016/j.cma.2004.12.005).
- [18] E. van Brummelen, Added Mass Effects of Compressible and Incompressible Flows in Fluid-Structure Interaction, *Journal of Applied Mechanics* 76 (2009) 021206. doi:[10.1115/1.3059565](https://doi.org/10.1115/1.3059565).
- [19] C. Förster, W. A. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, *Computer Methods in Applied Mechanics and Engineering* 196 (2007) 1278–1293. doi:[doi:10.1016/j.cma.2006.09.002](https://doi.org/10.1016/j.cma.2006.09.002).
- [20] J. Degroote, P. Bruggeman, R. Haelterman, J. Vierendeels, Stability of a coupling technique for partitioned solvers in FSI applications, *Computers & Structures* 86 (2008) 2224–2234. doi:[10.1016/j.compstruc.2008.05.005](https://doi.org/10.1016/j.compstruc.2008.05.005).
- [21] M. A. Fernández, J.-F. Gerbeau, C. Grandmont, A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid, *International Journal for Numerical Methods in Engineering* 69 (2007) 794–821. doi:[doi:10.1002/nme.1792](https://doi.org/10.1002/nme.1792).
- [22] S. Badia, F. Nobile, C. Vergara, Fluid–structure partitioned procedures based on robin transmission conditions, *Journal of Computational Physics* 227 (2008) 7027–7051. doi:[doi:10.1016/j.jcp.2008.04.006](https://doi.org/10.1016/j.jcp.2008.04.006).
- [23] J. A. Vierendeels, K. Rienslagh, E. Dick, P. R. Verdonck, Computer Simulation of Intraventricular Flow and Pressure Gradients During Diastole , *Journal of Biomechanical Engineering* 122 (2000) 667–674. doi:[10.1115/1.1318941](https://doi.org/10.1115/1.1318941).
- [24] J. Degroote, A. Swillens, P. Bruggeman, R. Haelterman, P. Segers, J. Vierendeels, Simulation of fluid–structure interaction with the interface artificial compressibility method, *International Journal for Numerical Methods in Biomedical Engineering* 26 (2010) 276–289. doi:[doi:10.1002/cnm.1276](https://doi.org/10.1002/cnm.1276).
- [25] Z. Li, J. Leduc, J. Nunez-Ramirez, A. Combescure, J.-C. Marongiu, A non-intrusive partitioned approach to couple smoothed particle hydrodynamics and finite element methods for transient fluid-structure interaction problems with large interface motion, *Computational Mechanics* 55 (2015) 697–718. doi:[10.1007/s00466-015-1131-8](https://doi.org/10.1007/s00466-015-1131-8).
- [26] S. Meduri, M. Cremonesi, U. Perego, O. Bettinotti, A. Kurkchubasche, V. Oancea, A partitioned fully explicit lagrangian finite element method for highly nonlinear fluid-structure interaction problems, *International Journal for Numerical Methods in Engineering* 113 (2018) 43–64. doi:[doi:10.1002/nme.5602](https://doi.org/10.1002/nme.5602).

- [27] M. Gibert, A new robust co-simulation approach for transient Fluid-Structure Interaction problems, Ph.D. thesis, 2022. URL: <http://www.theses.fr/2022LYSEI071>.
- [28] W. G. Dettmer, D. Perić, A new staggered scheme for fluid–structure interaction, *International Journal for Numerical Methods in Engineering* 93 (2013) 1–22. doi:[doi.org/10.1002/nme.4370](https://doi.org/10.1002/nme.4370).
- [29] W. G. Dettmer, A. Lovrić, C. Kadapa, D. Perić, New iterative and staggered solution schemes for incompressible fluid-structure interaction based on dirichlet-neumann coupling, *International Journal for Numerical Methods in Engineering* 122 (2021) 5204–5235. doi:[doi.org/10.1002/nme.6494](https://doi.org/10.1002/nme.6494).
- [30] W. A. Wall, D. P. Mok, E. Ramm, Iterative substructuring schemes for fluid structure interaction, in: W. Wendland, M. Efendiev (Eds.), *Analysis and Simulation of Multifield Problems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 349–360.
- [31] J. Vierendeels, L. Lanoye, J. Degroote, P. Verdonck, Implicit coupling of partitioned fluid–structure interaction problems with reduced order models, *Computers & Structures* 85 (2007) 970–976. doi:[doi.org/10.1016/j.compstruc.2006.11.006](https://doi.org/10.1016/j.compstruc.2006.11.006), fourth MIT Conference on Computational Fluid and Solid Mechanics.
- [32] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction, *Computers & Structures* 87 (2009) 793–801. doi:[doi.org/10.1016/j.compstruc.2008.11.013](https://doi.org/10.1016/j.compstruc.2008.11.013).
- [33] A. E. J. Bogaers, S. Kok, B. D. Reddy, T. Franz, Quasi-Newton methods for implicit black-box FSI coupling, *Computer Methods in Applied Mechanics and Engineering* 279 (2014) 113–132. doi:[doi.org/10.1016/j.cma.2014.06.033](https://doi.org/10.1016/j.cma.2014.06.033).
- [34] R. Haelterman, A. Bogaers, K. Scheufele, B. Uekermann, M. Mehl, Improving the performance of the partitioned qn-ils procedure for fluid–structure interaction problems: Filtering, *Computers & Structures* 171 (2016) 9–17. doi:[doi.org/10.1016/j.compstruc.2016.04.001](https://doi.org/10.1016/j.compstruc.2016.04.001).
- [35] B. Uekermann, Partitioned Fluid-Structure Interaction on Massively Parallel Systems, Ph.D. thesis, TUM, 2016. doi:[doi.org/10.14459/2016md1320661](https://doi.org/10.14459/2016md1320661).
- [36] N. Delaissé, T. Demeester, R. Haelterman, J. Degroote, Quasi-newton methods for partitioned simulation of fluid–structure interaction reviewed in the generalized broyden framework, *Archives of Computational Methods in Engineering* 30 (2023) 3271–17. doi:[doi.org/10.1007/s11831-023-09907-y](https://doi.org/10.1007/s11831-023-09907-y).
- [37] V. Shinde, E. Longatte, F. Baj, Y. Hoarau, M. Braza, Galerkin-free model reduction for fluid-structure interaction using proper orthogonal decomposition, *Journal of Computational Physics* 396 (2019) 579–595. doi:[doi.org/10.1016/j.jcp.2019.06.073](https://doi.org/10.1016/j.jcp.2019.06.073).
- [38] D. Xiao, P. Yang, F. Fang, J. Xiang, C. C. Pain, I. M. Navon, Non-intrusive reduced order modelling of fluid–structure interactions, *Computer Methods in Applied Mechanics and Engineering* 303 (2016) 35–54. doi:[doi.org/10.1016/j.cma.2015.12.029](https://doi.org/10.1016/j.cma.2015.12.029).
- [39] C. Dupont, F. De Vuyst, A.-V. Salsac, Data-driven kinematics-consistent model-order reduction of fluid–structure interaction problems: application to deformable microcapsules in a stokes flow, *Journal of Fluid Mechanics* 955 (2023) A2. doi:[doi.org/10.1017/jfm.2022.1005](https://doi.org/10.1017/jfm.2022.1005).
- [40] S. Fresca, A. Manzoni, Real-time simulation of parameter-dependent fluid flows through deep learning-based reduced order models, *Fluids* 6 (2021). doi:[doi.org/10.3390/fluids6070259](https://doi.org/10.3390/fluids6070259).
- [41] G. Haller, S. Ponsioen, Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction, *Nonlinear dynamics* 86 (2016) 1493–1534.

- [42] M. Cenedese, J. Axås, B. Bäuerlein, K. Avila, G. Haller, Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds, *Nature communications* 13 (2022) 872.
- [43] A. Tiba, T. Dairay, F. de Vuyst, I. Mortazavi, J.-P. B. Ramirez, Non-intrusive reduced order models for partitioned fluid-structure interactions, 2023. [arXiv:2306.07570](https://arxiv.org/abs/2306.07570).
- [44] L. Luo, X.-C. Cai, PIN<sup>†</sup> : Preconditioned inexact newton with learning capability for nonlinear system of equations, *SIAM Journal on Scientific Computing* 45 (2023) A849–A871. doi:[doi.org/10.1137/22M1507942](https://doi.org/10.1137/22M1507942).
- [45] T. Kadeethum, D. O’Malley, F. Ballarin, I. Ang, J. N. Fuhg, N. Bouklas, V. L. S. Silva, P. Salinas, C. E. Heaney, C. C. Pain, S. Lee, H. S. Viswanathan, H. Yoon, Enhancing high-fidelity nonlinear solver with reduced order model, *Sci Rep* 12 (2022) 20229. doi:[10.1038/s41598-022-22407-6](https://doi.org/10.1038/s41598-022-22407-6).
- [46] A. Zandbergen, T. van Noorden, A. Heinlein, Improving pseudo-time stepping convergence for cfd simulations with neural networks, 2023. [arXiv:2310.06717](https://arxiv.org/abs/2310.06717).
- [47] V. L. S. Silva, P. Salinas, M. D. Jackson, C. C. Pain, Machine learning acceleration for nonlinear solvers applied to multiphase porous media flow, *Computer Methods in Applied Mechanics and Engineering* 384 (2021) 113989. doi:[10.1016/j.cma.2021.113989](https://doi.org/10.1016/j.cma.2021.113989).
- [48] N. Delaissé, T. Demeester, D. Fauconnier, J. Degroote, Surrogate-based acceleration of quasi-newton techniques for fluid-structure interaction simulations, *Computers & Structures* 260 (2022) 106720. doi:[doi.org/10.1016/j.compstruc.2021.106720](https://doi.org/10.1016/j.compstruc.2021.106720).
- [49] M. P. Rast, Simultaneous solution of the navier-stokes and elastic membrane equations by a finite element method, *International Journal for Numerical Methods in Fluids* 19 (1994) 1115–1135. doi:[doi.org/10.1002/flid.1650191205](https://doi.org/10.1002/flid.1650191205).
- [50] M. Heil, Stokes flow in collapsible tubes: computation and experiment, *Journal of Fluid Mechanics* 353 (1997) 285–312. doi:[10.1017/S0022112097007490](https://doi.org/10.1017/S0022112097007490).
- [51] X. Y. Luo, T. J. Pedley, The effects of wall inertia on flow in a two-dimensional collapsible channel, *Journal of Fluid Mechanics* 363 (1998) 253–280. doi:[10.1017/S0022112098001062](https://doi.org/10.1017/S0022112098001062).
- [52] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, *Computer Methods in Applied Mechanics and Engineering* 134 (1996) 71–90. doi:[10.1016/0045-7825\(96\)01028-6](https://doi.org/10.1016/0045-7825(96)01028-6).
- [53] B. Gatzhammer, Efficient and flexible partitioned simulation of fluid-structure interactions, 2015. URL: <https://nbn-resolving.org/urn:nbn:de:101:1-2015022722192>.
- [54] C. Farhat, M. Lesoinne, P. Le Tallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity, *Computer Methods in Applied Mechanics and Engineering* 157 (1998) 95–114. doi:[doi.org/10.1016/S0045-7825\(97\)00216-8](https://doi.org/10.1016/S0045-7825(97)00216-8).
- [55] A. d. Boer, A. H. v. Zuijlen, H. Bijl, Comparison of conservative and consistent approaches for the coupling of non-matching meshes, *Computer Methods in Applied Mechanics and Engineering* 197 (2008) 4284–4297. doi:[doi.org/10.1016/j.cma.2008.05.001](https://doi.org/10.1016/j.cma.2008.05.001).
- [56] S. Sachs, M. Streitenberger, D. Stempel, M. Schäfer, Extrapolation methods for accelerating unsteady partitioned fluid-structure interaction simulations, *The International Journal of Multi-physics* 5 (2011) 287–298. doi:[10.1260/1750-9548.5.4.287](https://doi.org/10.1260/1750-9548.5.4.287).
- [57] B. M. Irons, R. C. Tuck, A version of the aitken accelerator for computer iteration, *International Journal for Numerical Methods in Engineering* 1 (1969) 275–277. doi:[doi.org/10.1002/nme.1620010306](https://doi.org/10.1002/nme.1620010306).

- [58] L. Sirovich, Turbulence and the dynamics of coherent structures. II. Symmetries and transformations, *Quarterly of Applied Mathematics* 45 (1987) 573–582. doi:[10.1090/qam/910463](https://doi.org/10.1090/qam/910463).
- [59] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1990. doi:[10.1137/1.9781611970128](https://doi.org/10.1137/1.9781611970128).
- [60] C. Audouze, F. De Vuyst, P. B. Nair, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, *Numerical Methods for Partial Differential Equations* 29 (2013) 1587–1628. doi:<https://doi.org/10.1002/num.21768>.
- [61] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1996) 267–288.
- [62] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, *The Annals of Statistics* 32 (2004) 407 – 499. doi:[10.1214/009053604000000067](https://doi.org/10.1214/009053604000000067).
- [63] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, Édouard Duchesnay, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research* 12 (2011) 2825–2830. URL: <http://jmlr.org/papers/v12/pedregosa11a.html>.
- [64] R. Geelen, S. Wright, K. Willcox, Operator inference for non-intrusive model reduction with quadratic manifolds, *Computer Methods in Applied Mechanics and Engineering* 403 (2023) 115717. doi:[10.1016/j.cma.2022.115717](https://doi.org/10.1016/j.cma.2022.115717).
- [65] J. L. Proctor, S. L. Brunton, J. N. Kutz, Dynamic mode decomposition with control, *SIAM Journal on Applied Dynamical Systems* 15 (2016) 142–161. doi:[10.1137/15M1013857](https://doi.org/10.1137/15M1013857).
- [66] T. Bou Hamdan, G. Mercère, T. Dairay, R. Meunier, P. Tremblay, P. Coirault, Tracking distributed parameters system dynamics with recursive dynamic mode decomposition with control, *SIAM Journal on Applied Dynamical Systems* 22 (2023) 37–64. doi:[10.1137/22M1478665](https://doi.org/10.1137/22M1478665).
- [67] P. Dadvand, R. Rossi, E. Oñate, An Object-oriented Environment for Developing Finite Element Codes for Multi-disciplinary Applications, *Arch Computat Methods Eng* 17 (2010) 253–297. doi:[10.1007/s11831-010-9045-2](https://doi.org/10.1007/s11831-010-9045-2).
- [68] S. N. Wood, Thin Plate Regression Splines, *Journal of the Royal Statistical Society Series B: Statistical Methodology* 65 (2003) 95–114. doi:[10.1111/1467-9868.00374](https://doi.org/10.1111/1467-9868.00374).
- [69] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, *Computer Methods in Applied Mechanics and Engineering* 190 (2001) 2681–2706. doi:[10.1016/S0045-7825\(00\)00260-7](https://doi.org/10.1016/S0045-7825(00)00260-7).
- [70] W. L. Wood, M. Bossak, O. C. Zienkiewicz, An alpha modification of newmark’s method, *International Journal for Numerical Methods in Engineering* 15 (1980) 1562–1566. doi:[doi.org/10.1002/nme.1620151011](https://doi.org/10.1002/nme.1620151011).
- [71] G. Scovazzi, R. Zorrilla, R. Rossi, A kinematically stabilized linear tetrahedral finite element for compressible and nearly incompressible finite elasticity, *Computer Methods in Applied Mechanics and Engineering* 412 (2023) 116076. doi:[doi.org/10.1016/j.cma.2023.116076](https://doi.org/10.1016/j.cma.2023.116076).
- [72] P. J. Roache, K. N. Ghia, F. M. White, Editorial Policy Statement on the Control of Numerical Accuracy, *Journal of Fluids Engineering* 108 (1986) 2–2. doi:[10.1115/1.3242537](https://doi.org/10.1115/1.3242537).