Strongly Coupled Fluid Structure Interaction

Robustness Analysis of Interface Quasi Newton Temporal Coupling Framework

K. Sripathy

Faculty of Aerospace Engineering, Aerodynamics

TU Delft Delft University of Technology

Challenge the future

Strongly Coupled Fluid Structure Interaction

Robustness Analysis of Interface Quasi Newton Temporal Coupling Framework

by

K. Sripathy

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Wednesday January 27, 2021 at 3:00 PM.

Student number:	4419804	
Project duration:	July 1, 2019 – January 27, 2021	
Supervisor:	Dr. ir. A. H. van Zuijlen	
Thesis committee:	Dr. ir. B. W. van Oudheusden	TU Delft
	Dr. ir. A. C. Viré	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft Delft University of Technology

Acknowledgments

Firstly I would like to thank my supervisor Dr. ir. Alexander van Zuijlen for offering me an opportunity to do my masters thesis under his guidance. He was thoroughly helpful in demystifying complex theories from scientific literature and answering my queries in a swift manner. He offered me complete freedom to conduct the research at my own pace. He was understanding of the difficulties arising from the pandemic, and ensured a seamless transition from on campus education to distance learning. I am grateful for his support during the weekly meetings, which helped me be motivated on my thesis.

The preCICE software community have been completely helpful in explaining the nuances involved in compiling and running various open source codes and their dependencies. This is my first time to be involved in an active open source collaboration, which was intimidating to begin with. Their quick responses for my queries have made it easier to use preCICE during the early stages of my thesis.

I owe a great deal to my sister Lekha and her friend Aiswarya for their expertise in signal analysis and processing, which was completely alien to me. Their suggestions helped me formulate the noise measurement and filtering frameworks used in my research.

My sincerest thanks to Nico and Colette for making my stay at the basement trouble-free. I thank my table mates Abhinandh and Yuyang in the basement for helping me get into ease with Linux and LaTeX ecosystem, which were a mainstay of my research. I thank Anand from 3ME for helping me understand FEM. My thanks to Derek for his tips on using preCICE and setting me up with the maintainers at preCICE. I am fond of the enlightening conversations with Nikhil and Henrik during our tea breaks. I greatly appreciate the shared experiences and activities with my colleagues Ankith, Athreya, Kushaal, Sampath, Sidhart and Shubham ranging from badminton at the sports center to the frisbee sessions just outside the High Speed Lab.

I am grateful to Gautam, Pugazh, Rahul and Venkatesan for their invaluable inputs in making me a better version of myself during my stay at Delft. Finally, I would like to express my deepest gratitude to my parents Swathi and Sripathy for their love and support from a great distance, even when I was not in a position to reciprocate it properly during my masters at TU Delft.

K. Sripathy
Den Haag, January 2021

Abstract

External and internal flows involving flexible structures exhibit interaction between fluid and the accompanying structure. Such phenomenon is observed in aerodynamic lifting/control surfaces, civil structures and cardiovascular systems. Analytical treatment of fluid-structure interaction is not a trivial task. Therefore, FSI problems are generally handled using experimental and computational analysis. Commercial numerical FSI software suites use partitioned FSI to solve problems in the above mentioned applications. In partitioned FSI, physics pertaining to fluid and structure domains are simulated separately in CFD and CSM solvers respectively. One way of coupling of fluid and structure domain in partitioned FSI is through interpolation of kinematic/kinetic quantity from solid/fluid to fluid/solid domain at the boundary between fluid and structure domains in a sequential manner. This sequential process can be represented mathematically using a root finding problem or fixed point problem with the location of the fluid-structure interface as unknown. Root finding problems employing Newton-Raphson iterations exhibit faster convergence in comparison to fixed point problems employing Gauss-Seidel iterations for strongly coupled FSI with large deformations.

IQN-ILS(interface quasi-Newton with inverse Jacobian from a least squares model) framework is a novel method to treat root finding problem, wherein the Jacobian is constructed using extrapolated location of the interface from previous iterations in a least squares fashion. In this research, IQN-ILS framework is employed on a FSI benchmark case — Cylinder with trailing flap to evaluate its robustness. This evaluation is performed on a test-bench comprising of `OpenFOAM` as CFD solver, `CalculiX` as CSM solver and `preCICE` as partitioned coupling library. Forces computed on the flap using IQN-ILS framework are plagued with numerical noise. It was not the case when the above mentioned FSI benchmark is treated with fixed point iteration, at the cost of longer computation time. Therefore, this research was involved with the investigation of numerical noise in force data obtained using IQN-ILS framework and addressing the same. Various strategies ranging from playing with the displacement tolerance to relative number of degrees of freedom on the fluid-structure interface were attempted. Although these strategies helped in reducing the noise in force data, it was difficult to identify the cause of noise in IQN-ILS framework. In spite of these noise prevention strategies, the force data obtained using IQN-ILS framework is still noisier than that of fixed point iteration.

However, the mechanism for noise generation in IQN-ILS framewok was understood. It was observed that the extrapolation of displacement in a least-square fashion from previous iterations seems to impose a minuscule noise on interface displacement. When these updated displacements at the flap interface are used to compute the force on the flap, the minuscule noise in displacement gets amplified when it is differentiated with respect to time. This issue is especially aggravated when a smaller time step is employed for the simulation. To counteract this issue, the displacements at flap interface can be smoothened before being interpolated to the CFD solver. To this end, three filtering techniques were proposed in this research to smoothen the displacements — Smoothing spline filter, Savitzky Golay filter, Sinusoidal curve fitting filter. Results from Smoothing spline filter and Savitzky Golay filter were promising when tested on a displacement series that was already computed by the test-bench. However, these filters were not tested in a live simulation. It requires the routines for the above mentioned filters to be integrated in `preCICE` through python callback interface.

Contents

List of Figures

List of Tables

Abbreviations

ALE	Arbitrary Lagrangian Eulerian Formulation
CFL	Courant Freidrichs Lewy
CPS	Conventional Parallel Staggered
CSS	Conventional Serial Staggered
$DGCL$	Discrete Geometric Conservation Law
$GMRES$	Generalized Minimal Residual
$IBQN-LS$	Interface Block Quasi Newton with Jacobian from a Least Squares Model
$IQN-ILS$	Interface Quasi Newton with Inverse Jacobian from a Least Squares Model
ISS	Improved Serial Staggered
$JFNK$	Jacobian Free Newton Krylov
$MFNK$	Matrix Free Newton Krylov
$OpenFOAM$	Open Source Field Operation and Manipulation
$PISO$	Pressure Implicit Split Operator
$preCICE$	Precise Code Interaction Coupling Environment
$PRIME$	Pressure Implicit Momentum Explicit
RBF	Radial Basis Functions
$SIMPLE$	Semi Implicit Method for Pressure Linked Equations
$SPOOLES$	Sparse Object Oriented Linear Equations Solver
TVM	Total Variation Method

Nomenclature

Latin alphabets

$\mathbf{a}$	Acceleration
$\mathbf{d}_\Gamma$	Interface displacement
h	Number of old samples for noise filtering
k	Iterator for sub-iterations
m	Iterator for time steps
p	Pressure
$\mathbf{r}_\Gamma$	Interface residual
t	Time
$\mathbf{u}$	Displacement in solid domain
$\mathbf{v}$	Velocity in fluid domain
$\mathbf{x}$	Interface position
$\mathbf{F}_\Gamma$	Interface force
$\mathbf{J}_\Gamma$	Interface Jacobian
K	Number of time periods in signal
N	Dimension of interface displacement
P	Cross correlation coefficient
$\mathbf{P}$	Piola-Kirchoff stress tensor

Greek alphabets

Δt	Time step
Δx	Representative mesh size
ϵ	Absolute coupling tolerance
ϵ'	Relative coupling tolerance
ρ	Density
$\boldsymbol{\sigma}$	Cauchy stress tensor
ω	Relaxation factor
Γ	Common interface
Θ	Smoothness function

1

Introduction

The functioning of various man-made and natural systems are governed by the same laws of physics. Existence of such systems is dictated by the four fundamental forces varying from electromagnetism, strong and weak interactions at microscopic level to gravitation at macroscopic level; and their interaction with one another. Natural systems have had millions of years of head start to adapt itself to this concoction of forces. Through acquisition of knowledge over a past couple of millennia, humanity has been able to catch up with nature in shaping the earth. Still, most of the complex phenomenon involving interaction of multiple branches of physics elude our understanding to this day. Sometimes, naturally occurring solutions to these complex problems have been copied by humans in the design of such complex systems. For example, the design of the nose of the Shinkansen 500 train is inspired from the shape of beak of kingfisher bird to reduce sonic booms when travelling at high speeds through narrow tunnels[2]. And, the design of ornithopter based drones is inspired by the flapping wings of insects to produce high lift at low Reynolds's number[3]. Still it is vital to keep pursuing research in these complex phenomena in hopes of coming up with solutions that are much more optimized than their natural counterparts, which is what multi-physics research strives to achieve.

Multi-physics is an interdisciplinary area, encompassing many science and engineering disciplines that treat coupled processes or systems involving multiple physical models[4]. The two examples discussed above can be construed as multi-physics problems. Analytical treatment of multi-physics systems is a tricky affair, owing to the absence of closed form of solutions. Therefore, they are generally studied using experimental or numerical simulations. The focus of this research is on computational simulation of multi-physics phenomena, specifically the fluid structure interaction. The flapping wing aerodynamics of ornithopter discussed earlier is a fluid structure interaction problem. Fluid Structure Interaction deals with the study of external and internal flows involving flexible structure that can interact with the fluid. Such phenomenon is observed in aerodynamic lifting/control surfaces, civil structures and cardiovascular systems.

IQN-ILS(Interface Quasi Newton with Inverse Jacobian from a Least Squares model) algorithm discussed in [5] is a state of the art FSI coupling algorithm that reduces the computational time required for resolving strongly coupled systems by a significant margin (30% - 80% faster) in comparison to the traditional algorithms[6]. However, most of the commercial FSI packages still employ the traditional methods for resolving FSI problems. IQN-ILS algorithm is mostly employed in academia and open source packages. In view of promoting the use of IQN-ILS algorithm in the industry, it was decided to narrow down this research to the IQN-ILS algorithm, specifically the evaluation of its robustness, where robustness of an algorithm is the measure of its stability and accuracy. And, the standard FSI benchmark — Cylinder with trailing flap is chosen to test the robustness of IQN-ILS algorithm. Therefore the goal of this research is:

This research aims to evaluate the robustness of IQN-ILS algorithm by testing it with various FSI benchmarks from the Cylinder with Trailing flap test case.

However due to complications involved in achieving the above objective, it has been revised to tackle the complications that arose during the middle stages of this research. Therefore the revised objective for this research is:

This research aims to investigate the noise formation in forces from FSI3 benchmark and address it using noise filters.

The workflow involved with this research can be broadly split into four stages, which were documented in the following four chapters as follow. In chapter 2, a review of the literature relevant to this research has been provided. Chapter 3 depicts the various milestones involved in the construction and validation of the test-bed required for performing numerical experiments with IQN-ILS algorithm. Chapter 4 is concerned with investigation and prevention of noise generated by the IQN-ILS algorithm in FSI3 benchmark. Chapter 5 deals with the mitigation of noise generated by the IQN-ILS algorithm using three noise filtering frameworks. In addition to the four chapters mentioned above, the final chapter consists of a summary of various conclusions made during the course of this research, and the recommendations made by the author for future research in this field.

2

Partitioned Fluid Structure Interaction

In this chapter, a summary of the literature pertaining to partitioned fluid structure interaction, specifically the various temporal coupling frameworks and their effectiveness in regard to convergence. In section 2.1, a concise account of balance laws of continuum mechanics is provided. A brief description of competing philosophies in numerical simulation of fluid structure interaction is provided in section 2.2. Section 2.3 describes the various steps involved in realizing any partitioned FSI simulation. In sections 2.4 and 2.5, a summary of various tight temporal coupling frameworks is provided. Finally, section 2.6 discusses the parameters that affect the stability of temporal coupling frameworks, and benchmarks the effectiveness of tightly coupled frameworks discussed in sections 2.4 and 2.5.

2.1. Balance laws of Continuum Mechanics

Behaviour of matter can be categorised into microscopic and macroscopic scale. Microscopic scale deals with phenomena at atomic level, whereas macroscopic scale deals with behaviour of bulk of matter that is visible to naked eye. Behaviour at macroscopic scale is a consequence of the behaviour at microscopic scale. Many engineering applications involve interpreting the behaviour at macroscopic scales, where one is not interested in the behaviour of atoms and molecules. In such cases, approximation of response at microscopic scales through suitable statistical functions serves as a good model for the macroscopic response, and have been recently adopted in CFD for resolving complex flow phenomena, see for example [7, 8].

Another approach to model macroscopic behaviour of matter, pioneered by Augustin-Louis Cauchy in 19^{th} century, is through fundamental balance laws of Continuum Mechanics and the relevant constitutive relations. Here, the matter is assumed to be uniformly distributed in space without voids, thereby allowing the representation of kinematic and kinetic quantities as continuous functions in space[10]. Thus, continuum mechanics can be described as the study of interaction between action(kinetic quantities) on matter and response(kinematic quantities) of matter on a macroscopic scale. It follows that matter has two states or configurations. Initial configuration at time $t = 0$ is termed as reference configuration and deformed configuration after the application of force at time $t > 0$ is called as current configuration[11]. Let $\mathbf{X}$ and $\mathbf{x}$ be the position vectors in reference and current configurations respectively, related by the mapping $\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}, t)$ as shown in figure 2.1. Then the deformation gradient tensor for the transformation is defined as:

$$\mathbf{F} := \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \boldsymbol{\nabla} \mathbf{x} \qquad [9] \tag{2.1}$$

Kinematic and kinetic fields can be expressed in the form of Lagrangian or Eulerian description. In Lagrangian description, the properties of a group of particles in a matter are tracked from the reference to current configuration. Whereas in Eulerian description, the properties are tracked at fixed locations in space from initial to final time. In Lagrangian description, displacement is normally used as the fundamental variable to describe other fields. It is defined as:

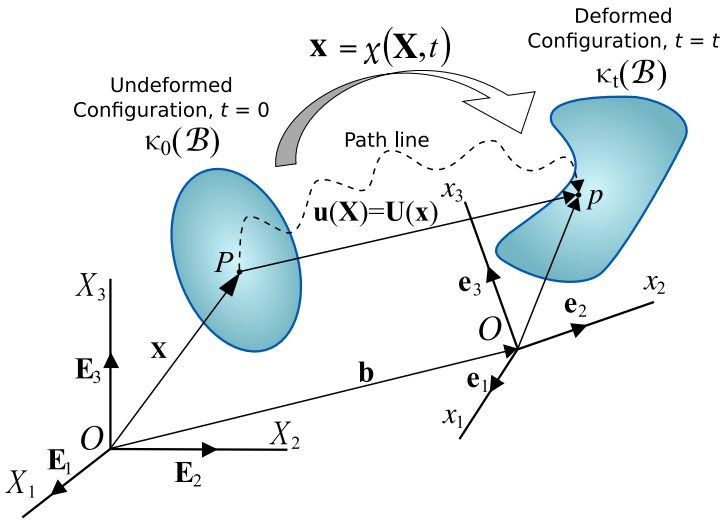


Figure 2.1: Continuum Mechanics — Configurations of Matter[9]

$$\mathbf{u}(\mathbf{X},t) := \chi(\mathbf{X},t) - \mathbf{X} \qquad [9] \tag{2.2}$$

In Eulerian description, velocity is normally employed as the fundamental variable to describe other fields. It is defined as:

$$\mathbf{v} := \frac{\partial \chi(\mathbf{X},t)}{\partial t} \qquad [12] \tag{2.3}$$

Consider $f(\mathbf{x},t)$ to be a physical quantity that is flowing through matter. Let $g(\mathbf{x},t)$ be the source on the surface bounding the matter and $h(\mathbf{x},t)$ be the source within the matter. The above mentioned fields can be either of scalar, vector or tensor valued functions. The position vector can be with respect to reference or current configuration. Also, consider ω and $\partial\omega$ to be the volume and surface area of the matter respectively. If v_n is the speed at which the boundary moves along its normal direction $\mathbf{n}(\mathbf{x},t)$, and $\mathbf{v}(\mathbf{x},t)$ be the velocity field of matter over which the physical quantity is to be determined, then the generalized balance law is expressed as:

$$\frac{D}{Dt}\left[\int_\Omega f(\mathbf{x},t)\mathrm{dV}\right] = \int_{\partial\Omega} f(\mathbf{x},t)\left[v_n(\mathbf{x},t) - \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t)\right]\mathrm{d}\mathbf{S} + \int_{\partial\Omega} g(\mathbf{x},t)\mathrm{d}\mathbf{S} + \int_\Omega h(\mathbf{x},t)\mathrm{dV}[9] \tag{2.4}$$

The temporal derivative term on LHS is known as material derivative. If reference coordinates are used, then the material derivative is simply:

$$\frac{D}{Dt}\left[f_{ij\ldots}(\mathbf{X},t)\right] := \frac{\partial}{\partial t}\left[f_{ij\ldots}(\mathbf{X},t)\right] \qquad [9] \tag{2.5}$$

If current coordinates are used, then the material derivative is defined as:

$$\frac{D}{Dt}\left[f_{ij\ldots}(\mathbf{x},t)\right] := \frac{\partial}{\partial t}\left[f_{ij\ldots}(\mathbf{x},t)\right] + \frac{\partial}{\partial x_k}\left[f_{ij\ldots}(\mathbf{x},t)\right]\frac{dx_k}{dt} \qquad [9] \tag{2.6}$$

In the remainder of this chapter, material derivative for any physical quantity $f(\mathbf{x},t)$ other than position vector $\mathbf{x}$ and displacement $\mathbf{u}$ shall be concisely represented as $\dot{f}(\mathbf{x},t)$. The generalized balance law in (2.4) was framed for a matter occupying certain space, i.e. finite body. If the integrands in (2.4) are continuous in ω and $\partial\omega$, localization principle can be employed to transform the integral balance law into its differential or local form. This implies that microscopic behaviour is identical to macroscopic behaviour for a matter in continuum mechanics. This approximation holds well as long as the dimensions of the finite body are much larger than the characteristic lengths(interatomic spacings in solids, mean free paths in fluids) in microscopic scale[10].

2.1.1. Localized balance laws in Eulerian description

In Eulerian description fields are expressed with respect to current coordinates. As mentioned earlier the temporal derivatives that shall be used to compose the balance laws are the material derivatives. If $\rho(\mathbf{x}, t)$ is the mass density in current coordinates than conservation of mass is expressed as:

$$\dot{\rho} + \rho \boldsymbol{\nabla} \cdot \mathbf{v} = 0 \qquad [9] \tag{2.7}$$

For incompressible flow, density of a particle moving along a streamline is constant, i.e. the material derivative $\dot{\rho} = 0$. Thus the conservation of mass reduces to a compatibility condition:

$$\boldsymbol{\nabla} \cdot \mathbf{v} = 0 \tag{2.8}$$

Let $\boldsymbol{\sigma}(\mathbf{x}, t)$ be the Cauchy stress tensor due to surface forces and $\mathbf{b}(\mathbf{x}, t)$ be the body force density. Then the conservation of linear momentum is expressed as:

$$\rho \dot{\mathbf{v}} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} - \rho \mathbf{b} = 0 \qquad [9] \tag{2.9}$$

The Cauchy stress tensor becomes symmetric from conservation of angular momentum, i.e.:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \qquad [9] \tag{2.10}$$

If $e(\mathbf{x}, t)$ is the internal energy per unit mass, $\mathbf{q}(\mathbf{x}, t)$ is heat flux and $s(\mathbf{x}, t)$ is energy source per unit mass, then conservation of energy is expressed as:

$$\rho \dot{e} - \boldsymbol{\sigma} : (\boldsymbol{\nabla} \mathbf{v}) + \boldsymbol{\nabla} \cdot \mathbf{q} - \rho s = 0 \qquad [9] \tag{2.11}$$

The above mentioned localized form of balance laws are valid only when the fields are continuous and if volume of the finite body does not change with respect to time.

2.1.2. Localized balance laws in Lagrangian description

In Lagrangian description fields are described with respect to reference coordinates. If ρ_0 is the mass density in reference coordinates then conservation of mass is expressed as:

$$\rho \det(\mathbf{F}) - \rho_0 = 0 \qquad [9] \tag{2.12}$$

In reference configuration, Piola-Kirchhoff stress tensor of first type will be used in lieu of Cauchy stress tensor. It is defined as:

$$\mathbf{P} := \det(\mathbf{F}) \boldsymbol{\sigma} \mathbf{F}^{-T} \qquad [9] \tag{2.13}$$

Therefore, conservation of linear momentum is expressed as:

$$\rho_0 \ddot{\mathbf{x}} - \boldsymbol{\nabla}_\circ \cdot \mathbf{P}^T - \rho_0 \mathbf{b} = 0 \qquad [9] \tag{2.14}$$

Conservation of angular momentum is expressed as:

$$\mathbf{F} \mathbf{P}^T = \mathbf{P} \mathbf{F}^T \qquad [9] \tag{2.15}$$

Finally, conservation of energy is expressed as:

$$\rho_0 \dot{e} - \mathbf{P}^T : \dot{\mathbf{F}} + \boldsymbol{\nabla}_\circ \cdot \mathbf{q} - \rho_0 s = 0 \qquad [9] \tag{2.16}$$

Unlike Eulerian representation of balance laws, localized Lagrangian conservation laws are applicable for varying volume of finite body with respect to time. Still, these laws are applicable for continuous fields only.

2.1.3. Lagrangian or Eulerian Description?

Both types of interpretation of Continuum Mechanics balance laws are equally valid for modelling the behaviour of a continuous matter. So far in the discussion of balance laws no comment was made on the nature of matter. Based on the ability to resist shear force, matter can be classified into fluid and solid. Fluids cannot resist shear force indefinitely, and will flow along the direction of applied shear force[10]. Whereas, solids attain a new configuration that is in equilibrium with the applied force. Once the force is removed, some solids return to their original resting configuration, whereas others remain in their new configuration. Irrespective of the ability to regain original configuration or not, the deformed configuration of the structure will be in the proximity of its original configuration, i.e. matter does not leave or enter solid domain. Therefore for transient mechanical analyses of solids, number of numerical calculations required for resolving Lagrangian description of governing laws will not increase exponentially with time, since there is no need to add new degrees of freedom to keep track of new matter.

In fluid dynamics, for any fluid domain of interest old matter leaving the domain is replaced by new matter entering the domain. And, for external flows boundaries of the flow domain are at the discretion of the user. Accurate prediction of properties of the fluid at boundaries of a flow domain is not a trivial task. Therefore, boundaries of the flow domain that are far away from the area of interest have to be employed. Approximate values of properties at such far field boundaries will have a negligible impact on the solution at the area of interest. Also, if the flow is dominated by convection, then matter enters and leaves the domain frequently. Large flow domain in combination with the need to account for new matter frequently makes resolution of Lagrangian description of balance laws highly unfeasible. Therefore for external flow analyses, Eulerian description of balance laws is employed for numerical fluid simulations in the industry. Owing to the philosophy of analysing the fluid properties at specific locations in a given region, the cost of numerical fluid simulations is independent of the rate at which fluid enters and leaves the flow domain.

2.1.4. Constitutive relations

The balance laws of continuum mechanics on its own are not a well defined problem. To obtain any meaningful results from the governing laws, they have to be supplemented with constitutive relations. As discussed earlier, governing laws are identical for solid and fluid material. It is through constitutive relations that a distinction is made between solids and fluids. Thus, a well posed problem can be formulated for any continuous matter by employing appropriate constitutive relations for that material. Normally, constitutive relations are formulated as mathematical models from the empirical research on behaviour of appropriate materials. In isotropic Newtonian fluids for instance, a linear relationship between stress tensor and velocity gradient was proposed by Stokes. If $\mathbf{L} := \boldsymbol{\nabla}\mathbf{v}$ is the velocity gradient in Eulerian description, then the general deformation law for isotropic Newtonian fluids is defined as[13]:

$$\sigma_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \delta_{ij}\lambda\boldsymbol{\nabla}\cdot\mathbf{L} \tag{2.17}$$

where p is the pressure arising from momentum exchange due to collisions of atoms; μ and λ are the first and second coefficients of viscosity respectively. The incompressible form of the Eulerian representation of balance laws in combination with the above described constitutive relation for stress tensor are the famous Navier-Stokes equations. Likewise, different types of constitutive relations are available for different types of solid materials. Let $\mathbf{C} := \mathbf{F}^T\mathbf{F}$ be the right Cauchy-Green strain tensor and $\mathbf{E} := \frac{1}{2}(\mathbf{C} - \mathbf{I})$ be the Green-Lagrange strain tensor. If $\mathbf{S} := \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T}$ is the Piola-Kirchhoff stress tensor of second type, then the constitutive relation for St.Venant-Kirchhoff material is described as:

$$\mathbf{S} = \lambda(\operatorname{tr}\mathbf{E})\mathbf{I} + 2\mu\mathbf{E} \qquad [11] \tag{2.18}$$

where λ and μ are referred to as Lame's first and second parameters. They are analogous to the coefficients of viscosity in Stoke's law. The constitutive relations for Newtonian fluid and St.Venant-Kirchhoff material were described with respect to Eulerian and Lagrangian frame of references respectively. Alternate formulations for the above mentioned constitutive relations are available with respect to other reference frames.

2.2. Numerical Fluid Structure Interaction

From section 2.1, it can be seen that the balance laws of continuum mechanics are non-linear partial differential equations, whose analytical treatment is possible only after simplifying the underlying physics phenomenon and the associated boundary conditions, **E.g:** Potential flow theory, Euler-Bernoulli beam theory. Analytic solution of such balance laws for complex physical phenomenon with real-life complex boundary conditions is impossible as of this writing. Presently, solutions of such system of partial differential equations are obtained by transforming it into a system of linear algebraic equations using discretization, and solving it. Unlike closed-form of solution from analytical analyses, in discretization solution is obtained only for certain discrete locations in the domain. Therefore, analytical solution, if available, can be considered as truth, and subject to appropriate refinements for various discretization parameters, numerical solution can converge to analytical solution. Owing to the complexity of modelling multiple types of matter in a given domain, solution of problems in Fluid Structure Interaction is generally obtained using the above mentioned numerical procedure. Based on how different matter are handled in any FSI simulation, numerical FSI can be classified into monolithic and partitioned FSI.

2.2.1. Monolithic FSI

In monolithic FSI, fluid and structure domains are treated as single entity with respect to the balance laws of continuum mechanics, i.e. Eulerian or Lagrangian description of the balance laws is employed for both the domains. It follows that identical schemes has to be employed for the discretization of spatial and temporal terms in the balance laws. It is through constitutive relations for the balance laws, that a distinction is made between the fluid and structure domain. By treating fluid and solid matter as a single entity, the physics governing the FSI is inherently coupled, resulting in solutions that are close to the ground truth. Systems involving fluid and structure domains are said to be coupled when the kinematic and dynamic boundary conditions at the interface of both the domains are equal. See section 2.3 for more information on coupled systems. However, creation of a monolithic FSI solver is not a trivial task, requiring the software architect to be adept in multiple disciplines of Mechanics, in this case — Solid and Fluid Mechanics. As the name **monolith** implies, monolithic FSI solver is not modular in nature, i.e. existing CFD and CSM codes cannot be repurposed for numerical FSI simulation in monolithic approach. Owing to the above two reasons, commercial monolithic FSI solvers are not available as of this writing. Development of monolithic FSI solvers is currently an active field of research, see for example [14–18].

2.2.2. Partitioned FSI

In partitioned FSI, fluid and structure domains are treated as separate entities with respect to the balance laws of continuum mechanics. Thus, unlike monolithic FSI solver, partitioned FSI solver involves integration of multiple single-physics solvers. Owing to the reasons described in section 2.1.3, Eulerian and Lagrangian balance laws are the de-facto mathematical models in CFD and CSM solvers respectively. Due to the ability to incorporate balance laws of different reference frames, partitioned FSI solver is capable of repurposing the above mentioned physics solvers for FSI. Since, partitioned FSI builds on decades of research involved in the development of commercial single-physics CFD and CSM solvers, proper implementation of partitioned FSI is not as complex as building a monolithic FSI solver. However, by considering the fluid and structure domain as separate entities, the physics of the respective domains are not inherently coupled in partitioned FSI. This generates partitioning error in the computed quantities. If partitioning error is removed from the system, then the solutions obtained using partitioned FSI is equaivalen to its monolithic counterpart. Depending on the application, the consequence of such physics decoupling ranges from a minor deviation against the ground truth to a completely non-physical solution. Subject to additional computational runtime, different strategies are available to enforce multi-physics coupling in partitioned FSI. Efficiency improvement of multi-physics coupling in partitioned FSI is the subject of this thesis.

2.3. Partitioned coupling

Partitioned FSI is realized by integration of CFD and CSM solver though partitioned coupling. There are three aspects to partitioned coupling — spatial coupling, mesh motion and temporal coupling.

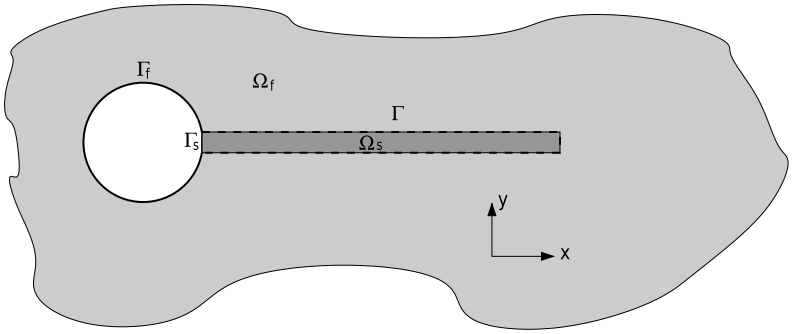


Figure 2.2: Partitioned Coupling — Representative Domain[19]

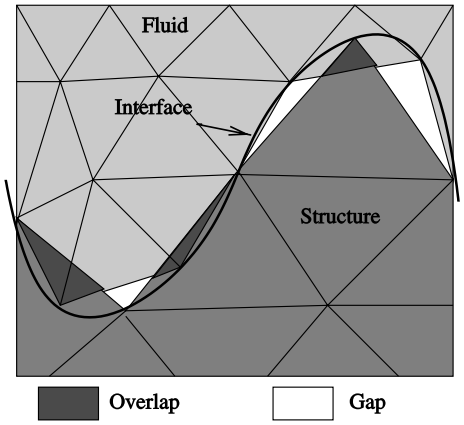


Figure 2.3: Partitioned Coupling — Spatial coupling of non-matching meshes[22]

2.3.1. Spatial coupling

As shown in figure 2.2, any FSI problem has fluid domain (Ω_f) and structure domain (Ω_s) bounded by non-common boundaries Γ_f and Γ_s respectively, and a common boundary Γ at the fluid-structure interface. Balance laws of continuum mechanics with constituent relations distinguishes only the behaviour between fluid and structure domain. It is through domain decomposition that a spatial distinction is made between the fluid and structure domain. Substructuring techniques[20] are the most common form of domain decomposition employed in FSI owing to its ability to treat the CFD and CSM solvers as black box physics solvers[21]. Since the fluid and structure solver are treated as isolated entities in structuring methods, it is through boundary conditions at the interface that a CFD solver acknowledges the presence of a structure domain, and the CSM solver acknowledges the presence of a fluid domain. The act of communication of boundary conditions between the fluid and structure domains on the interface is referred to as spatial coupling. In section 2.1, it was mentioned that the balance laws of continuum mechanics requires the space that it models to be a continuum. Thus spatial coupling can be thought as a mechanism to enforce continuity at the fluid-structure interface.

Among substructuring techniques, Dirichlet-Neumann partitioning is widely employed for spatial coupling in FSI. Let $\mathbf{t}_\Gamma = \boldsymbol{\sigma}_\Gamma \cdot \mathbf{n}_\Gamma$ be the traction exerted by the fluid on the structure at the interface, and $\mathbf{u}_\Gamma$ be the displacement of the structure domain at the interface. Due to different mesh resolutions for the fluid and structure domain, non-matching mesh typically exists at the interface as shown in figure 2.3. In such cases, data has to be interpolated from one domain to the other domain before spatial coupling can be executed. Let $\mathbf{s}_\Gamma = t(\mathbf{t}_\Gamma)$ be the interpolated traction from fluid to structure domain at the interface, and $\mathbf{d}_\Gamma = u(\mathbf{u}_\Gamma)$ be the interpolated displacement from structure to fluid domain at the interface. Thus, for realizing Dirichlet-Neumann partitioning, equilibrium(kinetic) and compatibility(kinematic) conditions have to be satisfied. The equilibrium condition is described as:

$$\mathbf{P}_\Gamma \cdot \mathbf{n}_\Gamma = \mathbf{s}_\Gamma \tag{2.19}$$

The compatibility condition for viscous fluid (no-slip boundary) is described as:

$$\mathbf{v}_\Gamma = \dot{\mathbf{d}}_\Gamma \tag{2.20}$$

The compatibility condition for inviscid flow (impermeable boundary) is described as:

$$\mathbf{v}_\Gamma \cdot \mathbf{n}_\Gamma = \dot{\mathbf{d}}_\Gamma \cdot \mathbf{n}_\Gamma \tag{2.21}$$

Finally, the compatibility conditions for the pseudo-structure of the fluid mesh are described as:

$$\begin{aligned} \tilde{\mathbf{x}}_\Gamma &= \tilde{\mathbf{x}}_{\Gamma,0} + \mathbf{d}_\Gamma \\ \dot{\tilde{\mathbf{x}}}_\Gamma &= \dot{\mathbf{d}}_\Gamma \end{aligned} \tag{2.22}$$

Pseudo-structure of a fluid mesh is an attribute of fluid mesh motion, and shall be appropriately treated in the next section. Many interpolation schemes are available for the interpolation functions $t(\mathbf{t}_\Gamma)$ and $u(\mathbf{u}_\Gamma)$. See [22] for a comprehensive review of various interpolation schemes in FSI.

2.3.2. Mesh motion

If displacement of the structure domain from the initial position is very small, then for the initial interface boundary defined on Γ_0 the viscous flow compatibility condition can be approximated as:

$$\mathbf{v}_{\Gamma_0} = \dot{\mathbf{d}}_{\Gamma_0} \tag{2.23}$$

The above approximation in combination with the Eulerian form of balance laws is referred to as transpiration approach[23], which was employed for aeroelastic analyses around vibrating bodies. In case of large structural displacments, Eulerian form of the balance laws cannot be employed, since the differential equation was formulated for fixed volume. Whereas, support for flow domain motion and deformation is inbuilt in Lagrangian description. Viscous flow domain is generally characterized by high velocity gradients in the proximity of an obstacle, leading to formation of vortices. In regions of vorticity, mixing or intersection of fluid occurs. For such scenarios, Lagrangian form of the balance laws fails, since the mesh warps and distorts, which is not the case with Eulerian form of balance laws. Therefore, it can be seen that both forms of the balance laws of continuum mechanics have their pros and cons. By combining the benefits of both the forms of balance laws, ALE[1] form of the balance laws was formulated to tackle fluid mesh motion[24].

Let $\tilde{\mathbf{x}}$ represent the position vector of a mesh node. If, $\dot{\tilde{\mathbf{x}}} = 0$ in Eulerian description, and $\dot{\tilde{\mathbf{x}}} = \mathbf{v}$ in Lagrangian description, then in ALE description $\dot{\tilde{\mathbf{x}}} \neq \mathbf{0}; \dot{\tilde{\mathbf{x}}} \neq \mathbf{v}$. The velocity of mesh motion is any arbitrary value, hence the adjective **arbitrary** in ALE. Temporal derivative for a physical quantity $\mathbf{f}$ in ALE formulation is defined as:

$$\dot{\tilde{\mathbf{f}}} = \frac{\partial \mathbf{f}}{\partial t} + \dot{\tilde{\mathbf{x}}} \cdot \boldsymbol{\nabla} \mathbf{f} \tag{2.24}$$

With the above definition for temporal derivative, the ALE form of the balance laws can be framed as:

$$\begin{aligned} \dot{\tilde{\rho}} + \rho \boldsymbol{\nabla} \cdot \mathbf{v} &= 0 \\ \rho \dot{\tilde{\mathbf{v}}} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} - \rho \mathbf{b} &= 0 \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}^T \\ \rho \dot{\tilde{e}} - \boldsymbol{\sigma} : (\boldsymbol{\nabla} \mathbf{v}) + \boldsymbol{\nabla} \cdot \mathbf{q} - \rho s &= 0 \end{aligned} \tag{2.25}$$

Since spatial derivatives here are identical to its Eulerian counterpart, no modification is required for spatial discretisation when translating from Eulerian to ALE formulation. However, in addition to

[1] Arbitrary Lagrangian Eulerian

the modification specified for the temporal derivatives in (2.24) for ALE formulation, temporal derivatives, including the mesh velocity for moving meshes, should also satisfy DGCL[2], i.e. the procedure to compute geometric parameters such as mesh velocity should be independent of nature of mesh motion[25]. Any higher order temporal discretization for fixed mesh will at least be first order accurate for moving meshes, if DGCL is satisfied[26]. See [25] for various temporal discretization schemes that satisfy DGCL; and [27] for a comprehensive treatment of ALE description including the ALE forms of constitutive relations and their applications in fluid and solid mechanics.

As far as the arbitrary motion for the fluid mesh is concerned, many models mimicking the behaviour of solid matter are available. Such models are also referred to as pseudo-structure models. Based on the pseudo-structure compatibility condition discussed in previous section, the fluid mesh displacement for the non-interface nodes can be mathematically described as:

$$\mathbf{K}_{\Omega_f \setminus \Gamma} \tilde{\mathbf{x}}_{\Omega_f \setminus \Gamma} = \mathbf{K}_{\Gamma} \tilde{\mathbf{x}}_{\Gamma} \tag{2.26}$$

where $\mathbf{K}$ is the stiffness matrix of the pseudo-structure. See [23] for a summary of various pseudo-structure models available for the fluid mesh motion.

2.3.3. Temporal Coupling

After employing appropriate spatial and temporal discretization schemes, the non-linear partial differential equations discussed in previous sections can be transformed to a system of non-linear algebraic equations. For an incompressible laminar flow using ALE formulation of the balance laws, the system of algebraic equations for the fluid domain can be concisely described with respect to the unknown variables at current time step as:

$$F(\mathbf{v}^n, p^n, \dot{\tilde{\mathbf{x}}}^n, \dot{\mathbf{d}}_{\Gamma}^n) = 0 \tag{2.27}$$

Similarly, for the structure domain the Lagrangian form of the balance laws can be discretized into a system of algebraic equations as:

$$S(\mathbf{u}^n, \dot{\mathbf{u}}^n, \mathbf{s}_{\Gamma}^n) = 0 \tag{2.28}$$

And, the discretized form of the pseudo-structure model for the fluid mesh is described as:

$$M(\tilde{\mathbf{x}}^n, \mathbf{d}_{\Gamma}^n) = 0 \tag{2.29}$$

In the above equations, values at previous time step are not written for convenience. Thus, a FSI problem involves solving a coupled three field system of algebraic equations, which along with the equilibrium and compatibility conditions form a well defined problem. Dirichlet-Neumann type of temporal coupling involves the following steps in resolving the three field problem:

1. Make a prediction for the interface displacement $\mathbf{d}_{\Gamma}^n$
2. With predicted interface displacement ($\mathbf{d}_{\Gamma}^n$), obtain fluid mesh displacement ($\tilde{\mathbf{x}}^n$) from (2.29) and velocity ($\dot{\tilde{\mathbf{x}}}^n$) that satisfies DGCL.
3. With prescribed interface velocity ($\dot{\mathbf{d}}_{\Gamma}^n$) and fluid mesh velocity ($\dot{\tilde{\mathbf{x}}}^n$), solve for flow velocity ($\mathbf{v}^n$) and pressure (p^n) from (2.27).
4. From updated flow velocity ($\mathbf{v}^n$) and pressure (p^n), calculate Cauchy stress tensor using (2.17). Calculate interpolated traction ($\mathbf{s}_{\Gamma}^n$) on the interface of the structure domain by $\mathbf{s}_{\Gamma}^n = t(\boldsymbol{\sigma}_{\Gamma}^n \cdot \mathbf{n}_{\Gamma})$.
5. Using interpolated interface traction ($\mathbf{s}_{\Gamma}^n$), solve for structure displacement ($\mathbf{u}^n$) and velocity ($\dot{\mathbf{u}}^n$). Using interpolation ($u(\mathbf{u}_{\Gamma}^n)$), obtain the updated interface displacement ($\hat{\mathbf{d}}_{\Gamma}^n$) for the current time step.

For brevity, steps 1 – 4 can be condensed as:

$$\mathbf{s}_{\Gamma}^n = F_f(\mathbf{d}_{\Gamma}^n) \tag{2.30}$$

[2]Discrete Geometric Conservation Law

Also, step 5 can be condensed as:

$$\hat{\mathbf{d}}_\Gamma^n = F_s(\mathbf{s}_\Gamma^n) \tag{2.31}$$

Thus, the entire temporal coupling process can be condensed in the form of a fixed-point non-linear equation as:

$$\hat{\mathbf{d}}_\Gamma^n = F_s \circ F_f(\mathbf{d}_\Gamma^n) \tag{2.32}$$

When the above equation is satisfied, the partitioning error is removed from the system. Thus, temporal coupling can be understood as an operation to reduce partitioning error in the system. To realize black box partitioned coupling, the three field solvers involved in the fixed-point equation are treated as isolated entities. According to [28], temporal coupling process can be classified into loose and tight coupling. In loose coupling, for each time step of an FSI simulation, steps 1–5 described in D-N temporal coupling cycle are executed only once. For certain applications one cycle of D-N temporal coupling is not sufficient to completely eliminate partitioning error. This can causes incomplete physics coupling between the flow and structure domain. Tight coupling in contrast involves multiple D-N temporal coupling cycles until the partitioning error is completely eliminated, resulting in complete physics coupling between the flow and structure domain, whose solution at each time step will be equivalent to the one obtained from a monolithic FSI simulation. It goes without saying that tight coupling is much more expensive than loose coupling. In [29], loose and tight couplings are also referred to as explicit and implicit couplings respectively.

There are many types of loose coupling schemes. Earliest loose coupling schemes for FSI were conceived by Farhat *et al.* in [30]. They were originally developed for coupled transient aeroelastic problems. The basic loose coupling scheme is the CSS[3] scheme. It is identical to the D-N temporal coupling cycle with $\mathbf{d}_\Gamma^{n+1} = \hat{\mathbf{d}}_\Gamma^n$. If the fluid and structure behaviour exhibit significant difference in temporal scales, then [30] recommends employing the CSS scheme with sub-cycling in the flow domain. This allows the usage of much smaller time step values in the flow domain. For simulations in parallel environments, [30] advocates the use of CPS[4] scheme, where the flow and structure physics are significantly decoupled. Therefore, it is accurate only for very small time steps. Also, [30] proposed the ISS[5] scheme, which is the improved version of CSS scheme. In ISS, the flow solver and structure solver are marched forward in time with same time step, but at different instances of time. It has significantly higher accuracy than the CSS scheme. Diagrammatic versions of the above mentioned loosely coupled schemes are illustrated in figure 2.4.

As mentioned earlier, tight coupling is realized by multiple sub-iterations of D-N temporal coupling cycles for each time step on the fixed point equation. Such tight coupling procedure is referred to as fixed point iteration. The degree of non-linearity in any three field problem for FSI is a consequence of the behaviour of matter in a given application. For applications with significant non-linearity, fixed point iterations either converge slowly or simply fail[31]. For such scenarios, the fixed point equation can be reformulated into a root finding problem as:

$$R(\mathbf{d}_\Gamma^n) := F_s \circ F_f(\mathbf{d}_\Gamma^n) - \mathbf{d}_\Gamma^n = 0 \tag{2.33}$$

The above mentioned function for root finding problem is nothing but the interface residual, i.e. $R(\mathbf{d}_\Gamma) = \mathbf{r}_\Gamma$. Description of the interface residual is given in section 2.4. The solution of the above mentioned non linear equation can be obtained using Newton-Raphson iteration. Various tight coupling frameworks are available for fixed point iteration as well as Newton-Raphson iteration. Irrespective of either type of iterations, most of the tightly coupled frameworks employ multiple loosely coupled CSS cycles in a time step. Literature review of various tightly coupled frameworks employed in resolving strongly coupled FSI is given in sections 2.4 and 2.5.

2.4. Fixed Point Iterative Frameworks

As mentioned in the previous section, each time step of the partitioned coupling has multiple fixed-point iterations for tight coupling. Such fixed-point iteration is also referred to as sub-iteration[32]. Thus,

[3] Conventional Serial Staggered
[4] Conventional Parallel Staggered
[5] Improved Serial Staggered

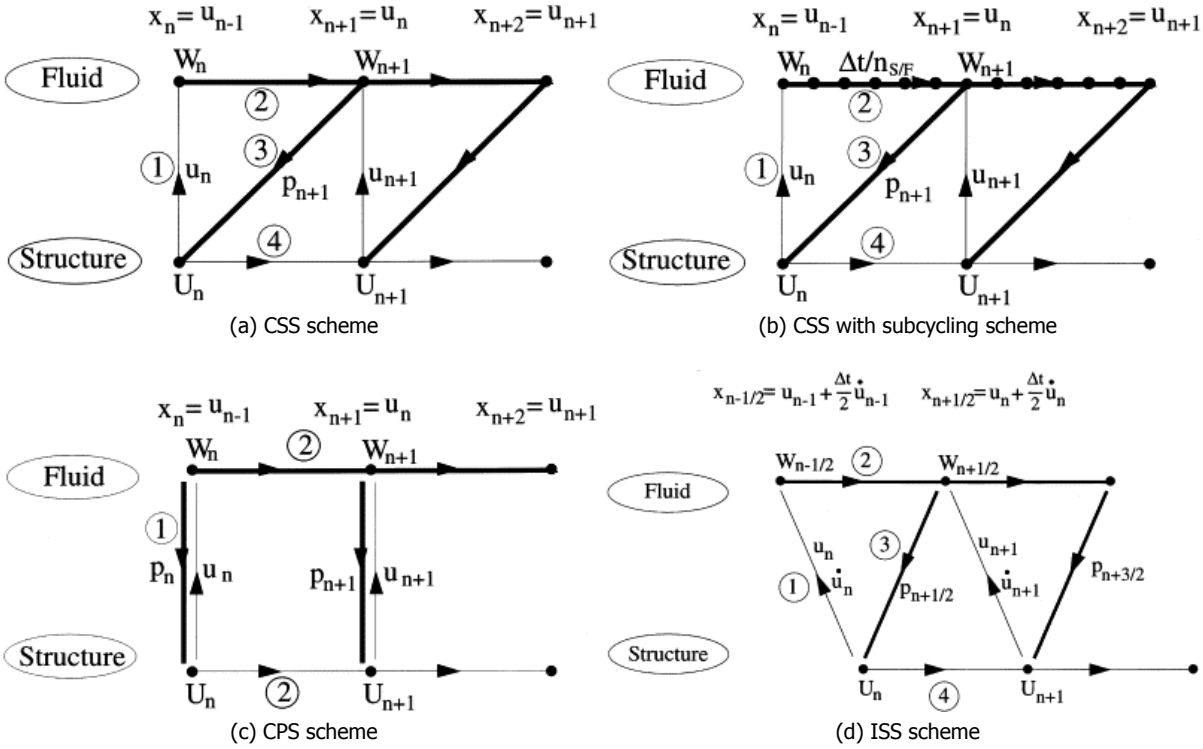


Figure 2.4: Temporal Coupling — Schematics of loosely coupled schemes[26]

for any time step in partitioned coupling, a sub-iteration for fixed point iteration is mathematically described as:

$$\hat{\mathbf{d}}_{\Gamma}^{i} = F_s \circ F_f(\mathbf{d}_{\Gamma}^{i}) \tag{2.34}$$

The current time step superscript n shall be ignored for concise relations in the remainder of this chapter. In the basic form of fixed point iteration, the displacement prediction for the succeeding sub-iteration is $\mathbf{d}_{\Gamma}^{i+1} = \hat{\mathbf{d}}_{\Gamma}^{i}$. The interface residual for a fixed point iteration is defined as:

$$\mathbf{r}_{\Gamma}^{i} := \hat{\mathbf{d}}_{\Gamma}^{i} - \mathbf{d}_{\Gamma}^{i} \tag{2.35}$$

Since partitioning error cannot be calculated for black box partitioned coupling, the interface residual can be thought as an indicator for the amount of partitioning error in the system. Thus, flow and structural physics are expected to be completely coupled when $\lim_{i\to\infty} \mathbf{r}^{i} = 0$. A better way to formulate a stopping criteria is by employing the euclidean length of a vector, also referred to as 2-norm. Thus, the stopping criteria for fixed point iteration is described as:

$$\|\mathbf{r}_{\Gamma}^{i}\| < \epsilon \tag{2.36}$$

where ϵ is the convergence/coupling tolerance, whose value is arbitrary. The lower the tolerance value, the lesser is the iterative error in the coupling framework. Once the converged displacement is obtained for a time step, it is used as the initial value for the sub-iterations in next time step, i.e. $\mathbf{d}_{n+1,\Gamma}^{0} = \hat{\mathbf{d}}_{n,\Gamma}^{*}$.

2.4.1. Relaxation methods

The fixed point iteration in its basic form is prone to divergence[32]. Richardson type of fixed point iteration with under-relaxation is recommended for such scenarios. In Richardson iteration, the displacement prediction for the next time step is described as:

$$\mathbf{d}_{\Gamma}^{i+1} = \omega_i \hat{\mathbf{d}}_{\Gamma}^{i} + (1 - \omega_i)\mathbf{d}_{\Gamma}^{i} \tag{2.37}$$

where ω_i is the relaxation parameter. $\omega_i > 1$ is over-relaxation. $\omega_i < 1$ is under-relaxation. $\omega_i = 1$ means no relaxation is employed. In sub-iterations with fixed relaxation, ω_i is a constant. Whereas sub-iterations with Aitken's relaxation employ a dynamic relaxation parameter. Aitken's relaxation was originally devised for efficient computational algorithms by Irons and Tuck in [33]. Aitken's dynamic relaxation parameter is defined as:

$$\omega_{i+1} := -\omega_i \frac{\mathbf{r}_\Gamma^i \cdot (\mathbf{r}_\Gamma^{i+1} - \mathbf{r}_\Gamma^i)}{\|\mathbf{r}_\Gamma^{i+1} - \mathbf{r}_\Gamma^i\|^2} \tag{2.38}$$

Since Aitken's relaxation parameter is recursively computed from its value at previous sub-iteration, a fixed relaxation parameter has to be employed for the initial sub-iteration. [33] recommends employing the final relaxation parameter from previous time step for the fixed relaxation parameter in the initial sub-iteration of the current time step. For Steepest descent relaxation, the dynamic relaxation parameter is defined as:

$$\omega_i := -\frac{\mathbf{r}_\Gamma^i \cdot \mathbf{r}_\Gamma^i}{\mathbf{r}_\Gamma^i \cdot \mathbf{J}_\Gamma^i \mathbf{r}_\Gamma^i} \tag{2.39}$$

where $\mathbf{J}_\Gamma$ is the interface Jacobian. It is defined as:

$$\mathbf{J}_\Gamma := \frac{\partial \mathbf{r}_\Gamma}{\partial \mathbf{d}_\Gamma} \tag{2.40}$$

Like Aitken's relaxation, initial sub-iteration with fixed relaxation is required before employing steepest descent relaxation. Exact computation of interface Jacobian is expensive[32], and various methods to approximate the above mentioned Jacobian will be treated in the next section.

2.5. Newton Raphson Iterative Frameworks

Unlike fixed point iteration, Newton Raphson iteration does not employ relaxation for obtaining the solution of a root finding problem. Therefore, using a correction $\Delta\mathbf{d}_\Gamma^i$, the interface displacment for the next sub-iteration is defined as:

$$\mathbf{d}_\Gamma^{i+1} := \mathbf{d}_\Gamma^i + \Delta\mathbf{d}_\Gamma^i \tag{2.41}$$

And, the displacement correction is obtained from the interface Jacobian and interface residual as:

$$\mathbf{J}_\Gamma^i \Delta\mathbf{d}_\Gamma^i = -\mathbf{r}_\Gamma^i \tag{2.42}$$

Once the converged displacement is obtained for a time step, it is used as the initial value for the sub-iterations in next time step, i.e. $\mathbf{d}_{n+1,\Gamma}^0 = \hat{\mathbf{d}}_{n,\Gamma}^*$. In contrast to the non-linear nature of the root finding problem in (2.33), the above equation is a linear problem for the solution of interface displacement correction. Thus, any iterative method can be employed for the solution of the system of linear equations. Depending on how the inverse of interface Jacobian is obtained, various Newton Raphson iterative frameworks for FSI are available.

2.5.1. Jacobian/Matrix free Newton Krylov method

In Jacobian free Newton Krylov method, any Krylov subspace method can be employed for obtaining the displacement correction. Thus for any sub-iteration i, the residual for the Krylov subspace method at j^{th} linear iteration is defined as:

$$\boldsymbol{r}_\Gamma^j := -\mathbf{r}_\Gamma^i - \mathbf{J}_\Gamma \Delta\mathbf{d}_\Gamma^j \tag{2.43}$$

Also in the GMRES[6] version of Krylov subspace method, the displacement correction at each linear iteration can be expanded into the span of its Krylov subspace basis as[34]:

$$\Delta\mathbf{d}_\Gamma^j = \Delta\mathbf{d}_\Gamma^0 + \sum_{t=0}^{j-1} \beta_t (\mathbf{J}_\Gamma)^t \boldsymbol{r}_\Gamma^0 \tag{2.44}$$

[6]Generalized Minimal Residual method

where β_j are the extrapolation coefficients that minimizes the linear residual, i.e.:

$$\boldsymbol{\beta}^j = \arg\min_{\beta_j} \|\boldsymbol{r}_\Gamma^j\| \tag{2.45}$$

In this framework, the product of interface Jacobian and linear residual can be approximated using finite differences as[35]:

$$\mathbf{J}_\Gamma \boldsymbol{r}_\Gamma^0 \approx \frac{R(\mathbf{d}_\Gamma^i + \delta \boldsymbol{r}_\Gamma^0) - \mathbf{r}_\Gamma^i}{\delta} \tag{2.46}$$

where δ is a small perturbation. The GMRES iterations are repeated until the linear residual approaches a very small value. With the computed interface displacement correction, the interface displacement for the next sub-iteration is obtained from (2.41). Since Jacobian is not computed explicitly in this framework, this method is referred to as Jacobian Free Newton Krylov. If no pre-conditioner is used for Krylov Subspace method, then this framework is also referred to as Matrix Free Newton Krylov. This framework is also referred to as FD-MFNK in [32]. Let F_s' and F_f' refer to the Jacobian of the output of the structure and flow solver with respect to the interface displacement. Then the MFNK framework in [32] describes the interface Jacobian and residual product as:

$$\mathbf{J}_\Gamma \boldsymbol{r}_\Gamma^0 = F_s' \circ F_f(\mathbf{d}_\Gamma^i) F_f'(\mathbf{d}_\Gamma^i) \boldsymbol{r}_\Gamma^0 - \boldsymbol{r}_\Gamma^0 \tag{2.47}$$

However, the computation of $F_f'(\mathbf{d}_\Gamma^i)$ involves quantities from different black box solvers, thereby making its calculation a challenging task. Therefore, for the MFNK framework in [32], it is approximated as:

$$F_f'(\mathbf{d}_\Gamma^i) \approx \frac{\partial F_f(\mathbf{d}_\Gamma^i)}{\partial \mathbf{v}_\Gamma^i} \frac{1}{\Delta t} \tag{2.48}$$

2.5.2. Interface Newton-Krylov/GMRES method

Michler *et al.* proposed a black box coupling framework for FSI that builds on the JFNK framework in [36]. Rather than expanding the interface displacement correction as a span of its Krylov subspace basis, a simple vector extrapolation is employed for the interface displacement correction as:

$$\Delta \mathbf{d}_\Gamma^j = \sum_{t=0}^{j} \psi_t (\hat{\mathbf{d}}_\Gamma^t - \mathbf{d}_\Gamma^i) \tag{2.49}$$

Also, the product of interface Jacobian and residual is not approximated here with a matrix free method for the determination of interface displacement correction. Instead, secant method is employed in lieu of Newton Raphson method, where the derivatives in the Jacobian matrix are approximated by first order finite difference discretization. The finite difference approximation of the interface Jacobian is described as:

$$\mathbf{J}_\Gamma^j \approx \frac{\mathbf{r}_\Gamma^j - \mathbf{r}_\Gamma^i}{\hat{\mathbf{d}}_\Gamma^j - \mathbf{d}_\Gamma^i} \tag{2.50}$$

From (2.43), (2.49) and (2.50), the linear residual for interface GMRES framework can be reformulated as:

$$\boldsymbol{r}_\Gamma^j = -\mathbf{r}_\Gamma^i - \sum_{t=0}^{j} \psi_t (\mathbf{r}_\Gamma^t - \mathbf{r}_\Gamma^i) \tag{2.51}$$

The extrapolation coefficients ($\boldsymbol{\psi}$) are again obtained such that it minimizes the linear residual as:

$$\boldsymbol{\psi}^j = \arg\min_{\psi_j} \|\boldsymbol{r}_\Gamma^j\| \tag{2.52}$$

With the updated extrapolation coefficients, the updated linear residual is obtained. The linear iterations are repeated until the norm of the linear residual converges to a linear tolerance value, i.e:

$$\|\boldsymbol{r}_\Gamma^j\| < \epsilon_0 \tag{2.53}$$

Using the extrapolation coefficients of the final linear iteration, the displacement correction is calculated using (2.49). With the updated displacement correction, the interface displacement for the next non-linear iteration is computed using (2.41). The sequence of multiple linear iterations followed by a single non linear iteration is repeated multiple times until the interface residual converges to tolerance ϵ. [36] recommends employing fixed or dynamic relaxation for the linear iterations to improve stability.

2.5.3. Interface Quasi Newton Method

Degroote *et al.* have recently proposed another black box coupling framework for FSI in [5], that builds on the reduced order model for the linear iterations of JFNK, described in [37]. Reduced order model is an extension of inexact Newton methods summarised in [35]. Its stated that the linear iterations need not be solved to a stricter linear tolerance, especially if such linear iterations are for initial sub-iterations of a time step. In [37], instead of employing lenient linear tolerance for linear iterations, a low fidelity mathematical model that models only the added fluid mass was employed for their version of inexact Newton JFNK framework. This low fidelity mathematical model was referred to as reduced order model. Owing to the simplicity of this reduced order model, any direct method can be employed for obtaining the displacement correction. Thus in contrast to JFNK and interface GMRES method, interface Quasi Newton methods have one linear iteration for each sub-iteration. Degroote *et al.* modified this reduced order model to support black box partitioned coupling in [5], which they termed it as IQN-ILS[7] framework. Suppose i is the current sub-iteration and k represent the set of sub-iterations from initial sub-iteration till the previous sub-iteration i.e. $k \in \{0, 1, ..., i-2, i-1\}$. For current sub-iteration, two vectors $\Delta\mathbf{r}_\Gamma^k$ and $\Delta\hat{\mathbf{d}}_\Gamma^k$ are defined as:

$$\begin{aligned} \Delta\mathbf{r}_\Gamma^k &:= \mathbf{r}_\Gamma^k - \mathbf{r}_\Gamma^i \\ \Delta\hat{\mathbf{d}}_\Gamma^k &:= \hat{\mathbf{d}}_\Gamma^k - \hat{\mathbf{d}}_\Gamma^i \end{aligned} \tag{2.54}$$

The matrices $\mathbf{V}^i$ and $\mathbf{W}^i$ for the current sub-iteration can be constructed by concatenation of vectors $\Delta\mathbf{r}_\Gamma^k$ and $\Delta\hat{\mathbf{d}}_\Gamma^k$ respectively, i.e.:

$$\begin{aligned} \mathbf{V}^i &:= [\Delta\mathbf{r}_\Gamma^{i-1} \quad \Delta\mathbf{r}_\Gamma^{i-2} \quad ... \quad \Delta\mathbf{r}_\Gamma^1 \quad \Delta\mathbf{r}_\Gamma^0] \\ \mathbf{W}^i &:= [\Delta\hat{\mathbf{d}}_\Gamma^{i-1} \quad \Delta\hat{\mathbf{d}}_\Gamma^{i-2} \quad ... \quad \Delta\hat{\mathbf{d}}_\Gamma^1 \quad \Delta\hat{\mathbf{d}}_\Gamma^0] \end{aligned} \tag{2.55}$$

V,W matrices require at least two values of $\mathbf{r}_\Gamma$ and $\hat{\mathbf{d}}_\Gamma$ for computation. Thus, the first two sub-iterations of a time step have to be executed using fixed point framework with relaxation before the employment of IQN-ILS framework. Suppose N is the dimension of the interface displacement. If more than p sub-iterations with IQN-ILS framework is required for convergence, then the earliest columns on the right hand side of the matrices are to be deleted. Thus the highest possible dimension for the $\mathbf{V}^i$ and $\mathbf{W}^i$ matrices is $N \times N$. By ensuring that $i \leq N$, if $\Delta\mathbf{r}_\Gamma^k$ corresponding to different sub-iterations are linearly independent, then these set of vectors is the basis for space $D \subseteq \mathbb{R}^N$. Thus any residual difference vector can be expressed as the span of the above mentioned basis i.e.:

$$\Delta\mathbf{r}_\Gamma \approx \sum_{k=0}^{i} \alpha^k(\Delta\mathbf{r}_\Gamma^k) = \mathbf{V}^i\boldsymbol{\alpha}^i \tag{2.56}$$

where $\boldsymbol{\alpha}^{i^T} := [\alpha^0 \ \alpha^1 \ ... \ \alpha^{i-1} \ \alpha^i]$ is the vector of extrapolation coefficients to be determined. For convergence, the desired interface residual has to be a vector of zeros, i.e. $\Delta\mathbf{r}_\Gamma = \mathbf{0} - \mathbf{r}_\Gamma^i$. Thus, the system of equations to determine $\boldsymbol{\alpha}^i$ is:

$$\mathbf{V}^i\boldsymbol{\alpha}^i = -\mathbf{r}_\Gamma^i \tag{2.57}$$

Since $i \leq p$, it forms an over-determined system of equations, and a unique solution is only possible for the extrapolation coefficients when $-\mathbf{r}_\Gamma^i$ lies in the column space of $\mathbf{V}^i$. If not, then an approximate

[7] Interface Quasi Newton with Inverse Jacobian from a Least Squares Model

solution can be obtained by an orthogonal projection of $-\mathbf{r}_\Gamma^i$ on the column space of $\mathbf{V}^i$. This is performed through least squares minimization. Various procedures are available to perform least squares minimisation. Ordinary least squares gives a closed solution for $\boldsymbol{\alpha}^i$ as:

$$\boldsymbol{\alpha}^i = -(\mathbf{V}^{iT}\mathbf{V}^i)^{-1}\mathbf{V}^{iT}\mathbf{r}_\Gamma^i \tag{2.58}$$

Other methods for least squares minimization are QR decomposition, Singular Value decomposition etc. The ordinary least squares method becomes unstable for higher number of columns in $\mathbf{V}^i$. Thus, the extrapolation coefficients were calculated in [5] using QR decomposition. In QR decomposition a rectangular matrix is decomposed into a product of orthogonal matrix ($\mathbf{Q}$) and upper triangular matrix ($\mathbf{R}$), i.e.:

$$\mathbf{V}_{N\times i}^i = \mathbf{Q}_{N\times i}^i \mathbf{R}_{i\times i}^i \tag{2.59}$$

Another advantage of QR decomposition is its inherent capability to identify the columns in $\mathbf{V}^i$ matrix that are linearly dependent. The columns in $\mathbf{V}^i$ matrix need not be linearly independent. This is possible if the root finding problem is non linear or if round-off errors are encountered[38]. The diagonal element corresponding to the linearly dependent column will be zero in the R matrix from QR decomposition. Therefore, the columns corresponding to linear dependency in $\mathbf{V}^i$ have to be removed before solving equation 2.57. This is referred to as QR-filtering. In [5], QR-filtering was performed by identifying rows in $\mathbf{R}$ whose diagonal element is zero; and eliminating the column corresponding to that row from $\mathbf{V}^i$. This QR-filtering is repeated until the resulting upper triangular matrix from QR-decomposition of $\mathbf{V}^i$ is non singular. On substituting (2.59) in (2.57):

$$\mathbf{Q}^i\mathbf{R}^i\boldsymbol{\alpha}^i = -\mathbf{r}_\Gamma^i \tag{2.60}$$

Inverse of an orthogonal matrix is its transpose. Thus:

$$\mathbf{R}^i\boldsymbol{\alpha}^i = -\mathbf{Q}^{iT}\mathbf{r}_\Gamma^i \tag{2.61}$$

Since $\mathbf{R}$ is an upper triangular matrix, $\boldsymbol{\alpha}$ can be obtained using back substitution. From (2.35):

$$\mathbf{d}_\Gamma = \hat{\mathbf{d}}_\Gamma - \mathbf{r}_\Gamma \tag{2.62}$$

Since spanning of vectors is a linear function, the above equation can be written as:

$$\Delta\mathbf{d}_\Gamma = \Delta\hat{\mathbf{d}}_\Gamma - \Delta\mathbf{r}_\Gamma \tag{2.63}$$

$\Delta\mathbf{d}_\Gamma$ is the parameter required for obtaining $\mathbf{d}_\Gamma^{i+1}$ from (2.41). As discussed earlier, $\Delta\mathbf{r}_\Gamma = -\mathbf{r}_\Gamma^i$ is desired. $\Delta\hat{\mathbf{d}}_\Gamma$ corresponding to $\Delta\mathbf{r}_\Gamma = -\mathbf{r}_\Gamma^i$ can be obtained using the computed extrapolation coefficients from (2.61) as:

$$\Delta\hat{\mathbf{d}}_\Gamma^i = \mathbf{W}^i\boldsymbol{\alpha}^i \tag{2.64}$$

On substituting (2.64) in (2.63), the displacement correction using IQN-ILS framework for the current-sub-iteration is described as:

$$\Delta\mathbf{d}_\Gamma^i = \mathbf{W}^i\boldsymbol{\alpha}^i + \mathbf{r}_\Gamma^i \tag{2.65}$$

This displacement correction is employed in (2.41) to obtain the interface displacement for the next sub-iteration, i.e. $\mathbf{d}_\Gamma^{i+1} = \mathbf{d}_\Gamma^i + \mathbf{W}^i\boldsymbol{\alpha}^i + \mathbf{r}_\Gamma^i$. Components of $\mathbf{r}_\Gamma^i$ that lie in the column space of $\mathbf{V}^i$ are dampened significantly using IQN-ILS in comparison to fixed point iteration. Components of $\mathbf{r}_\Gamma^i$ that lie outside the column space of $\mathbf{V}^i$ are dampened at the same rate as fixed point iterations. Another version of interface quasi Newton method was developed by Vierendeels *et al.* where the inverse of the approximated interface Jacobian is not required. This framework is referred to as IBQN-LS[8]. For more details about its implementation, see [39].

[8]Interface Block Quasi Newton with Jacobian from a Least Squares model

2.5.4. Interface Jacobian construction using data from multiple time steps

Suppose, a time series has $n+1$ samples, i.e. $\mathbf{t} = \{t_0, t_1, \dots, t_n\}$. If n is the latest time step and i is the currently active sub-iteration in the latest time step, then for k^{th} sub-iteration in m^{th} time step $\Delta\mathbf{r}^{\mathbf{k}}_{m,\Gamma}$ be defined as:

$$\Delta\mathbf{r}^{\mathbf{k}}_{m,\Gamma} := \mathbf{r}^{\mathbf{k}}_{m,\Gamma} - \mathbf{r}^{\mathbf{i}}_{n,\Gamma} \tag{2.66}$$

And, $\Delta\hat{\mathbf{d}}^{k}_{m,\Gamma}$ can be defined as:

$$\Delta\hat{\mathbf{d}}^{k}_{m,\Gamma} := \hat{\mathbf{d}}^{k}_{m,\Gamma} - \hat{\mathbf{d}}^{i}_{n,\Gamma} \tag{2.67}$$

If time step is smaller such that the interface displacements for those consecutive time steps can approximate a continuous curve, then $\Delta\mathbf{r}^{\mathbf{k}}_{m,\Gamma}$ and $\Delta\hat{\mathbf{d}}^{k}_{m,\Gamma}$ can be employed for the construction of inverse interface Jacobian. For i^{th} sub-iteration the $\mathbf{V}^i_n$ and $\mathbf{W}^i_n$ are defined as:

$$\begin{aligned}
\mathbf{V}^i_n &:= [\Delta\mathbf{r}^{i-1}_{n,\Gamma} \quad \Delta\mathbf{r}^{i-2}_{n,\Gamma} \quad \dots \quad \Delta\mathbf{r}^{k}_{n-1,\Gamma} \quad \Delta\mathbf{r}^{k-1}_{n-1,\Gamma} \quad \dots \quad \Delta\mathbf{r}^{k}_{m,\Gamma} \quad \Delta\mathbf{r}^{k-1}_{m,\Gamma} \quad \dots] \\
\mathbf{W}^i_n &:= [\Delta\hat{\mathbf{d}}^{i-1}_{n,\Gamma} \quad \Delta\hat{\mathbf{d}}^{i-2}_{n,\Gamma} \quad \dots \quad \Delta\hat{\mathbf{d}}^{k}_{n-1,\Gamma} \quad \Delta\hat{\mathbf{d}}^{k-1}_{n-1,\Gamma} \quad \dots \quad \Delta\hat{\mathbf{d}}^{k}_{m,\Gamma} \quad \Delta\hat{\mathbf{d}}^{k-1}_{m,\Gamma} \quad \dots]
\end{aligned} \tag{2.68}$$

Again, the number of columns in the above matrices should be less than the number of degrees of freedom on the interface. Other than the first time step, relaxation need not be employed for the first sub-iteration of a time step. Since the $\mathbf{V}^1_n$ and $\mathbf{W}^1_n$ matrices are partially constructed before the start of IQN-ILS framework in the current time step, it is expected that fewer sub-iterations are required for the convergence of interface displacement[6]. IQN-ILS framework with inverse Jacobian update from m previous time steps shall be termed as IQN-ILS(m). With higher number of columns in the $\mathbf{V}^i_n$ matrix, there is higher probability for the columns to be linearly dependent of one another, which is a potential disadvantage with construction of inverse Jacobian from multiple time steps. Similar technique can be employed for accelerating the convergence of linear and non-linear iterations in interface GMRES framework, which shall be termed as interface GMRES(m).

2.6. Robustness and Efficiency of temporal coupling frameworks

Based on the intensity of interaction between the physics models of different matter in a continuum, [28] classifies multi-physics coupling such as FSI into weak and strong coupling. For clarity, suppose the system of balance equations and its associated constitutive relations discussed in section 2.1 for a continuum involving structure and fluid domain be simplified into a semi-discrete form as[40]:

$$\dot{\mathbf{w}} = \mathbf{A}\mathbf{w} = \begin{pmatrix} \mathbf{A}_s & \mathbf{A}_{sf} \\ \mathbf{A}_{fs} & \mathbf{A}_f \end{pmatrix} \begin{pmatrix} \mathbf{w}_s \\ \mathbf{w}_f \end{pmatrix} \tag{2.69}$$

In the above equation, $\mathbf{w}_s$ and $\mathbf{w}_f$ represents the degrees of freedom of the flow and structure domain, $\mathbf{A}_s$ and $\mathbf{A}_f$ represents the spatial discretization of the structure and fluid domain, $\mathbf{A}_{sf}$ and $\mathbf{A}_{fs}$ represents the spatial coupling between degrees of freedom of different domains. Other than the non-zero entries for the interface degrees of freedom, the off block matrices in $\mathbf{A}$ are sparse in nature. According to [28], when off-diagonal block matrices are negligible with respect to the diagonal block matrices in $\mathbf{A}$, then the physics coupling is deemed to be weak. Otherwise, if off-diagonal matrices are quite significant, then the physics coupling is deemed to be strong. Unsteady aeroelastic problems[41] constitute a weakly coupled FSI, whereas blood flow in blood vessels[42] constitute a strongly coupled FSI.

For a matrix with significant off-diagonal entries, it cannot be a symmetric positive definite matrix, resulting in numerical instability issues when iterative methods are employed to obtain its solution[34]. Hence, obtaining the solution of a strongly coupled partitioned FSI is not a trivial task. Due to partial physics coupling in loosely coupled frameworks, they are highly prone to divergence for strongly coupled FSI[43]. Therefore, loosely coupled frameworks are typically employed for weakly coupled FSI. Also it was concluded in [43] that tightly coupled frameworks require more sub-iterations to resolve strongly coupled FSI in comparison to weakly coupled FSI. As long as smaller spatial and temporal discretizations are employed, accuracy of the solution obtained from any partitioned FSI simulation can be ensured. Owing to the issues related to strongly coupled FSI, stability is the limiting factor that affects the

robustness of any tightly coupled partitioned FSI frameworks. Thus, in addition to the tightly coupled frameworks discussed in sections 2.4 and 2.5, many other frameworks were developed with the view of addressing the numerical instability associated with a strongly coupled partitioned FSI simulation.

2.6.1. Paramters influencing stability of tightly coupled frameworks

The strength of the physics coupling is related to certain paramters. These parameters can be physical — fluid-structure mass ratio[43], Mach number (fluid compressibility)[44], non-dimensional structural stiffness[45] or numerical — non-dimensional time step[45].

In [45], partitioning error was used as a metric to analyse the behaviour of fixed point iterations for a stongly coupled FSI involving the 1D axisymmetric unsteady flow in a flexible tube of circular cross-section. After Fourier decomposition of the partitioning error, it was found that the fixed point iteration tend to amplify smooth error modes and decay rough error modes. Also, it was found that fixed point iterations become more unstable with smaller structural stiffness and time step.

[44] studies the effect of added fluid mass and compressibility on the stability of fixed point iterations as well as interface GMRES framework for a strongly coupled FSI involving a channel flow with a flexible wall at the base. Here, added fluid mass is an abstract mathematical quantity, that is a function of the principal eigenvalue of the Poincaré-Steklov operator of the flow solver in [44]. It was found that the ratio of added fluid mass to structural mass affects the stability of fixed point iterations. As long as that ratio is smaller than 1, fixed point iteration is stable. And, for time step approaching zero, the added mass for incompressible flow approaches to a constant, whereas for compressible flow it approaches zero. Owing to the presence of added fluid mass irrespective of the time step size, it was concluded that fixed point iteration for incompressible flow is conditionally stable. By employing fixed point iteration with under-relaxation, it can be made unconditionally stable for incompressible flow. In another article by Van Brummelen, it was concluded that interface GMRES can tolerate higher added fluid mass than fixed point iterations[46].

2.6.2. Performance of tightly coupled frameworks

Efficiency and robustness goes hand in hand to judge the desirability of any tightly coupled partitioned FSI framework for the industry. For problems of industrial scale involving large number of degrees of freedom, where very long simulation time within the black box flow and structure solvers is the norm, it is desirable for the coupling framework to have as few sub-iterations as possible. Therefore number of sub-iterations per time step, averaged over the entire duration of the FSI phenomena, is employed as the metric to benchmark the efficiency of various coupling frameworks.

In [32], fixed point iteration frameworks as well as the JFNK (FD MFNK and MFNK) coupling frameworks were benchmarked for the driven cavity with flexible bottom and the pressure wave in flexible tube test cases. For the driven case with stiffer base, Aitken's relaxation framework was found to be the most efficient. Whereas for the case with flexible base, FD MFNK and Aitken's relaxation were equally efficient. For the much strongly coupled pressure wave in flexible tube test case, the JFNK coupling frameworks required fewer sub-iterations than Aitken's relaxation framework.

In [6], Aitken's relaxation framework was benchmarked with interface GMRES and quasi Newton frameworks for the 2D cylinder with trailing flap and 3D flexible tube test case. For the 2D cylinder test case, the IQN-ILS(m) and IBQN-LS(m) frameworks with vector reuse were the most efficient, followed by Aitken's framework and interface GMRES(m) framework. Again, for the 3D flexible tube test case, the quasi Newton frameworks were the most efficient. However, interface GMRES(m) framework is much faster than the Atkin's relaxation framework. Thus, among the tightly coupled frameworks discussed in this chapter, the quasi Newton frameworks appear to be the most efficient.

2.7. Conclusions

An overview of the theoretical aspects involved in the modelling and execution of a numerical FSI simulation was provided in this chapter. As long as the matter can be approximated as a continuum, the balance laws of continuum mechanics along with appropriate constitutive relations can model the behaviour of solids as well as fluids. Direct computation of the aggregated fluid-structure balance laws was referred to as monolithic FSI simulation. Partitioned FSI simulation in contrast executes an FSI simulation through the spatial and temporal coupling of balance laws of segregated fluid and structure domains. A literature review on existing loosely and tightly coupled frameworks, including a brief

discussion on their effectiveness for weakly and strongly coupled FSI applications was also provided in this chapter. It was found that fixed point framework with Aitken's under-relaxation was mostly effective for relatively weak FSI coupling, whereas for strong FSI coupling, the quasi Newton frameworks were found to be more effective. Since, this research will be dealing with strongly coupled FSI applications, and the objective of this research is to evaluate the robustness of IQN-ILS framework, this framework will be employed for the FSI simulations to be performed in this research.

3

Partitioned FSI Testbed

For executing the author's research, a testbed involving the assembly of various software packages has to be formulated. However, before performing numerical experiments with the testbed, it has to be validated with existing literature to ensure its proper functioning. Therefore, this chapter treats the formulation and validation of this testbed. In section 3.1 the components of testbed are described. Section 3.2 discusses the literature to be employed for validating the testbed. Results of flow and structure solver validation are shown in sections 3.3 and 3.4. Finally, the validation of testbed in its entirety is executed in section 3.5.

3.1. Testbed Formulation

In line with the requirements of a black box partitioned FSI solver, the testbed consists of three components — flow solver, structure solver and the coupling library. Open source packages will be used for the above three components in order to facilitate the modification of their respective source codes.

3.1.1. Flow solver

For this research, `OpenFOAM v1906`[1] is employed as the flow solver. It is the largest open source CFD platform with multiple C++ libraries for different flow conditions and models. In addition to fluid dynamics, it also offers support for numerical simulation of acoustics, solid mechanics and electromagnetics[47]. The libraries of OpenFOAM are based on the work of Weller *et al.* in [48].

3.1.2. Structure solver

`CalculiX v2.15`[49] is chosen as the structure solver. It is an open source CSM solver based on the finite element discretization of continuum mechanics governing laws[12]. It is capable of resolving large non-linear deformations on the structure domain, which is vital for resolving strongly coupled FSI.

3.1.3. Coupling Library

Black box partitioned coupling requires the preservation of source codes of the above discussed flow and structure solver. For this reason, `preCICE`[2] is chosen as the coupling library for this research. Like `OpenFOAM` it is a platform with multiple libraries for different types of partitioned multi-physics simulations[50]. For partitioned FSI coupling, since `OpenFOAM` and `CalculiX` are officially supported at the time of this writing, the flow and structure solvers were chosen retrospectively based on this official support. A general schematic of functioning of `preCICE` is portrayed in figure 3.1. It can be seen that `preCICE` requires adapters for interfacing with the flow and structure solvers. The current adapter for `OpenFOAM` was originally developed for Conjugate Heat Transfer by Chourdakis in [51]. Risseeuw extended the functionality of this adapter to support FSI in [52]. The current adapter for `CalculiX` was originally developed for CHT by Yau in [53]. Rusch extended the functionality of this

[1] Open Source Field Operation and Manipulation
[2] Precise Code Interaction Coupling Environment

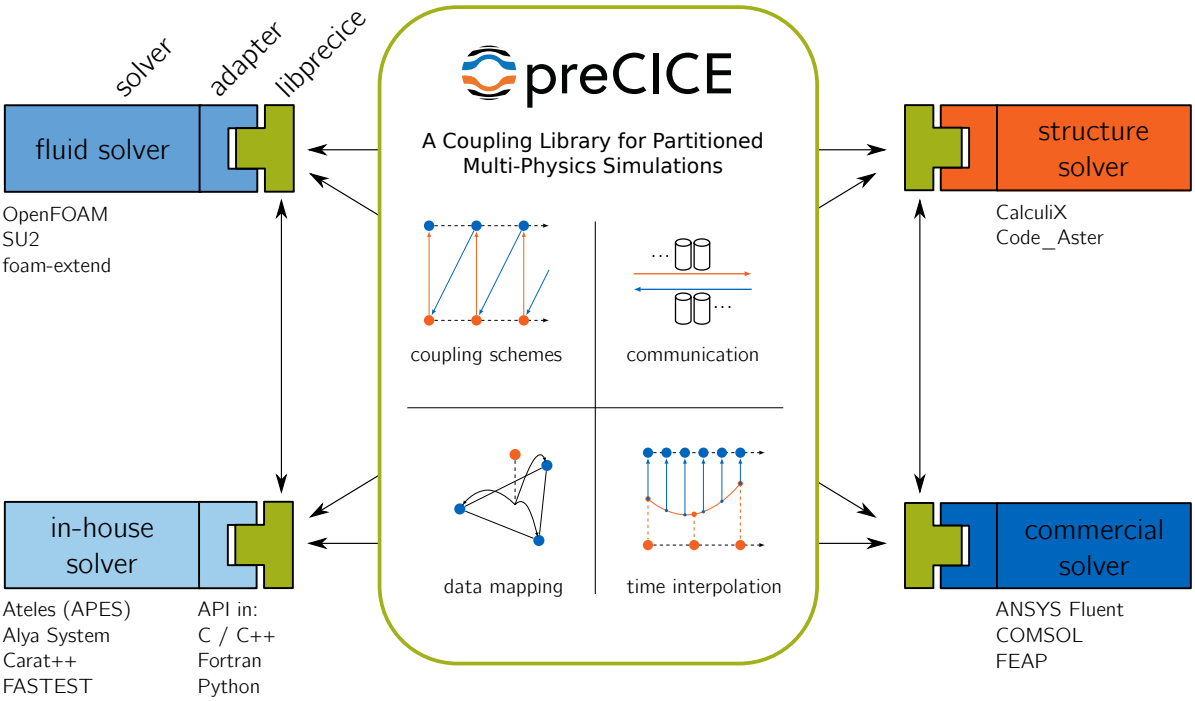


Figure 3.1: Schematic of `preCICE`[50]

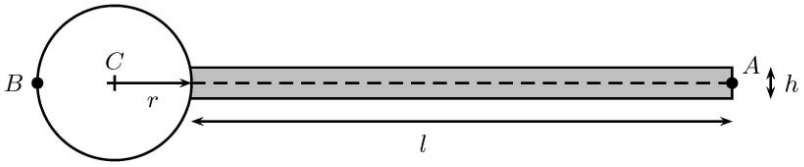


Figure 3.2: Cylinder with Trailing Flap — Geometry[55]

adapter to support FSI in [54]. By integrating the above two adapters within their respective physics solvers, a black box partitioned FSI coupling can be realised.

3.2. Reference Literature for Validation

To ensure that the partitioned FSI testbed formulated in previous section provides accurate results, it has to be validated with existing literature. It was decided to utilize the numerical FSI benchmark prescribed by Turek and Hron in [55] for validation. They employed their monolithic FSI solver described in [17] with finite element spatial discretization for both the flow and structure domain to perform numerical FSI simulations.

3.2.1. Test Case

The test case involves an incompressible laminar flow on a two dimensional geometry. The geometry is a cylinder with a trailing flap as shown in figure 3.2. For all intents, the cylindrical portion of the geometry can be assumed as a void, whereas the flap is the structural domain of the problem. The structural domain constitutes an elastic compressible solid. The flow domain surrounding the geometry is illustrated in figure 3.3. The spatial parameters of flow domain as well as the geometry are prescribed in table 3.1.

3.2.2. Boundary conditions

On the inlet of control volume a parabolic velocity profile is prescribed for the fluid as:

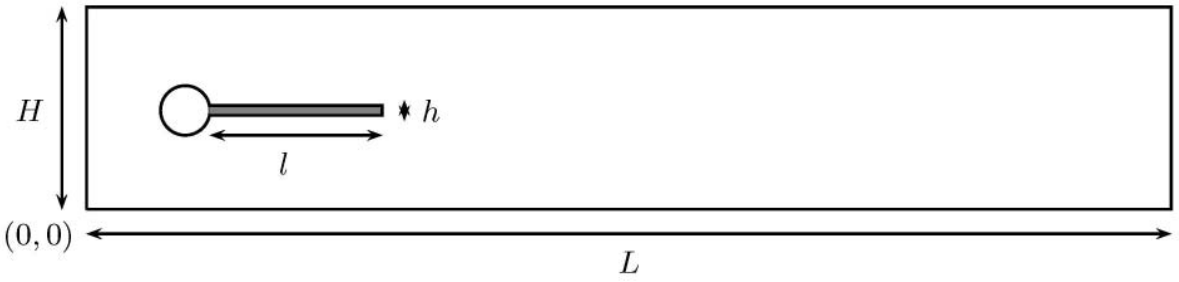


Figure 3.3: Cylinder with trailing Flap — Flow Domain[55]

Geometry Parameters		**Value** [m]
channel length	L	2.5
channel width	H	0.41
cylinder centre location	C	(0.2,0.2)
cylinder radius	r	0.05
flap length	l	0.35
flap thickness	h	0.02
initial flap tip location	A	(0.6,0.2)
cylinder root location	B	(0.2,0.2)

Table 3.1: Cylinder with Trailing Flap — Geometry Parameters[55]

$$v^f(0,y) = 1.5\overline{U}\frac{y(H-y)}{\left(\frac{H}{2}\right)^2} = 1.5\overline{U}\frac{4.0}{0.1681}y(0.41-y) \qquad [55] \tag{3.1}$$

For the outlet of control volume a constant pressure is prescribed for the fluid. Since the flow is incompressible, it is not necessary to prescribe an absolute value for the outlet pressure. Therefore, zero pressure is specified at the outlet. Outlet velocity and inlet pressure of the fluid are considered to be the extensions of the internal field, and therefore possess zero gradient boundary condition. For the walls of the geometry and control volume no slip velocity boundary condition is employed. Zero pressure gradient is employed for the walls.

In the structural domain, the root of the flap is constrained to be fixed, i.e. zero displacement and slope. To enable 2D analysis in `CalculiX`, the remaining nodes are constrained to deform in xy plane by specifying zero displacement in z direction.

3.2.3. Initial condition

[55] recommends employing a gradual acceleration to the flow at inlet for unsteady simulations such that:

$$v^f(t,0,y) = \begin{cases} v^f(0,y)\left(\frac{1-\cos\left(\frac{\pi}{2}t\right)}{2}\right) & \text{if } t < 2.0 \\ v^f(0,y) & \text{if } t \geq 2.0 \end{cases} \qquad [55] \tag{3.2}$$

Zero internal field is employed for the initial velocity, and pressure field. As for the structure domain is considered, zero internal field is employed for initial displacement and velocity field.

Parameters	CFD1	CFD2	CFD3
$\rho^f [10^3 \text{kg m}^{-3}]$	1	1	1
$\nu^f [10^{-3} \text{m}^2 \text{s}^{-1}]$	1	1	1
$\bar{U} [\text{m s}^{-1}]$	0.2	1	2
Re	20	100	200

Table 3.2: Cylinder with Trailing Flap — Flow simulation parameters[55]

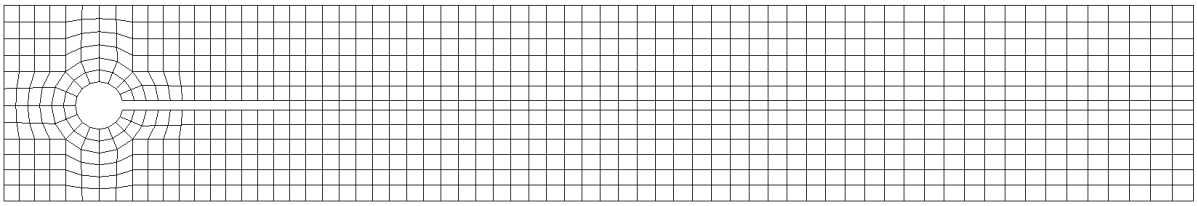


Figure 3.4: Cylinder with Trailing Flap — Sample flow mesh with 856 hexahedron cells

3.3. Flow Solver Validation

Three tests — CFD1, CFD2, CFD3 were proposed in [55] to validate the flow solver. CFD1 and CFD2 are steady state simulations, whereas CFD3 is a transient simulation. The material properties of the fluid as well as the mean inlet velocity for the above three cases are prescribed in table 3.2. Workflow of any CFD simulation can be split into three sections — Mesh generation, Flow solver computation and data postprocessing.

3.3.1. Mesh generation

An open source pre/post processing platform `Salome v9.3` was employed to generate the meshes for the flow domain. Seven grids with different number of hexahedron cells were generated to perform mesh independence study for all the CFD tests. A coarse faithful reproduction of the grids employed in this study is illustrated in figure 3.4. The number of cells in each mesh is tabulated in table 3.3. The meshes were generated such that most of the cells are uniformly refined.

3.3.2. Coupled pressure velocity equations

In incompressible flow, density of the fluid is constant. Therefore, density cannot be employed as a degree of freedom for the flow problem. Thus, in addition to the two flow velocity components, pressure has to be employed in order to make the system of equations fully determined. However, due to constant density, utilization of constant pressure field computed from isentropic relation leads to non physical solution. Due to the absence of an explicit way to calculate the pressure term in the momentum equation, the discretized system of flow governing equations resembles a strongly coupled implicit problem as shown in (3.3) for a steady state problem.

$$\mathbf{Au} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_b \\ \mathbf{0} \end{pmatrix} \quad [56] \tag{3.3}$$

Owing to the above set of equations constituting a saddle point problem, obtaining solution is not a trivial task[56]. Said difficulty can be circumvented by decomposing the system matrix using Schur's complement as shown in (3.4), and decoupling pressure from velocity.

Mesh	1	2	3	4	5	6	7
Cells	13504	30240	54016	84160	121536	165088	216064

Table 3.3: Cylinder with Trailing Flap — Number of flow domain cells

$$\mathbf{A} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{F} & 0 \\ \mathbf{B} & -\mathbf{B}\mathbf{F}^{-1}\mathbf{B}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{F}^{-1}\mathbf{B}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \mathbf{L}\mathbf{U} \quad [56] \tag{3.4}$$

Once decoupled, a prediction for velocity ($v^{\star}$)can be made from pressure and velocity at previous iteration, with which the predicted pressure ($p^{\star}$) is computed as shown in (3.5).

$$\begin{pmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{B} & -\mathbf{B}\mathbf{D}^{-1}\mathbf{B}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{v}^{\star} \\ \mathbf{p}^{\star} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_b \\ \mathbf{0} \end{pmatrix} \quad [56] \tag{3.5}$$

The velocity (v') and pressure corrections (p') can be computed from the predicted value, which is used to compute the corrected pressure and velocity as:

$$\begin{pmatrix} \mathbf{I} & \mathbf{D}^{-1}\mathbf{B}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{v}' \\ \mathbf{p}' \end{pmatrix} = \begin{pmatrix} \mathbf{v}^{\star} \\ \mathbf{p}^{\star} \end{pmatrix} \quad [56] \tag{3.6}$$

The predicted velocity satisfies the momentum equation, whereas the corrected velocity satisfies the continuity equation. This cycle of prediction and correction is repeated until the predicted velocity is equal to the corrected velocity($\mathbf{v}' = \mathbf{v}^{\star}$), i.e velocity satisfies both continuity and momentum equations. Since, computation of matrix inverse is an expensive affair, the inverse of matrix $\mathbf{F}$ associated with finite volume discretization of convective and viscous terms is approximated with the matrix $\mathbf{D^{-1}}$.

3.3.3. SIMPLE Family of Algorithms

The **SIMPLE**[3] algorithm is one of the methods to decouple pressure velocity equations developed by Patankar and Spalding in [57] for transient flow simulation. Thus each time step consists of multiple prediction and correction steps. Calculation of pressure correction requires knowledge of interpolated surface velocity corrections arising from the integration of Laplacian term in the momentum equation, a consequence of finite volume discretization. In the original SIMPLE method this correction was neglected. This approximation does not have any bearing on the final solution since the correction term will vanish on convergence. However, when this surface velocity correction is large during the non-final iterations, such approximation leads to slow convergence or divergence. Thus, under relaxation has to be employed for both the pressure and velocity field.

The consistent form of the SIMPLE algorithm, **SIMPLEC** was developed by Van Doormaal and Raithby in [58]. In SIMPLEC, the weighted mean of the interpolated surface velocity corrections is approximated by the cell centre velocity correction, thereby leading to increased convergence stability by recognizing the role of surface velocity correction terms. In fact, under relaxation is not necessary for the pressure field in SIMPLEC. But, each SIMPLEC iteration will be more expensive than a SIMPLE iteration.

Maliska and Raithby developed the **PRIME**[4] algorithm in [59]. As the name suggests, in each PRIME iteration momentum equation is solved explicitly whereas the pressure equation is solved implicitly. A consequence of this algorithm is the appearance of an additional term in the pressure equation, involving the difference of interpolated predicted and initial surface velocities, whose function is to diminish the role of interpolated surface velocity correction term[56]. Therefore, neglecting this minuscule difference will not significantly affect the convergence stability. However, owing to the explicit treatment of momentum equation, the time step is restricted by the CFL[5] number.

Issa developed the **PISO**[6] algorithm in [60]. It takes advantage of the implicit nature of SIMPLE algorithm as well as the robustness of PRIME algorithm. One PISO iteration is composed of a single SIMPLE iteration followed by one or many PRIME iterations[56]. With each additional PRIME iteration, the value of the corrected difference term discussed in PRIME algorithm decreases further. Thus, with more PRIME iterations, ignorance of the difference term will have marginal consequence on the stability of convergence. Irrespective of the above discussed algorithms, each time step requires multiple iterations of these algorithms. Although, each PISO iteration is computationally more expensive than SIMPLE algorithm, it might be the case that few PISO iterations are required per time step for a transient flow problem.

[3] Semi Implicit Method for Pressure Linked Equations
[4] Pressure Implicit Momentum Explicit
[5] Courant Friedrichs Lewy
[6] Pressure Implicit Split Operator

3.3.4. Solvers in OpenFOAM

Four OpenFOAM solvers suitable for resolving incompressible laminar flow are icoFoam, simpleFoam, pisoFoam and pimpleFoam. icoFoam is pisoFoam solver without turbulence modelling, and therefore its treatment is avoided. simpleFoam is a steady state flow solver based on SIMPLE/SIMPLEC algorithm, whereas pisoFoam and pimpleFoam are transient flow solvers based on PISO algorithm. The difference arises from the fact that pisoFoam uses single PISO iteration for each time step, whereas pimpleFoam uses multiple PISO iterations for each time step. In table 3.6 the maximum CFL number for different meshes are prescribed. Since, the CFL number for the transient CFD3 simulations is much higher than 1 for all the six meshes, it was decided to employ pimpleFoam solver for transient simulations. pisoFoam and pimpleFoam can also be employed for obtaining the steady state solution by marching the transient problem to steady state solution. However, it requires multiple linear equation iterations for each time step, and if one is not interested in the evolution of the flow, then it is much faster to obtain the steady state solution by solving the boundary value problem without the temporal terms by any one of the robust linear equation solvers[61]. Therefore, the consistent version of simpleFoam, activated by enabling the consistent option, is employed for steady state CFD1 and CFD2 simulations. Also, the consistent version of SIMPLE was employed for the SIMPLE part of the pimpleFoam solver in CFD3 simulation.

Irrespective of the algorithm employed, the tolerance for the underlying preconditioned Krylov subspace linear equation solvers are assigned as 10^{-10}. Also, the pressure and velocity tolerance for the final solution was set as 10^{-8}. The difference between the two tolerances is that the former controls the iterative error of thee linear equation solver, whereas the latter controls the iterative error of the pressure-velocity coupling. Since the maximum orthogonality for all the meshes ranged between 35° and 45°, two non orthogonal corrections were employed for each pressure velocity coupling iteration. And, for each PISO iteration, one PRIME iteration is employed after a SIMPLE iteration, thereby leading to two pressure corrections in each PISO iteration. Finally, under relaxation factor of 0.9 was employed for the velocity field if the linear equation solver fails to converge without under relaxation.

Second order Gaussian quadrature integration with linear interpolation is employed for the gradient terms on cell centres as well as the convective divergence term. For the gradients terms on the surface centres required for the computation of the Laplacian term as well as the pressure term, Rhie-Chow interpolation with non orthogonal correction has been employed to avoid the checker-boarding issue in collocated meshes[56]. Again, Gaussian quadrature integration is employed for the Laplacian term. Finally, for the transient CFD3 simulation, the second order backward differentiation formula (BDF-2) was employed for the time marching of the unsteady problem. As observed in [55], the transient simulation was performed for 10s with time step of 0.005s to obtain a constant amplitude periodic solution.

3.3.5. Numerical results for CFD simulations

The simulations were executed on a Linux workstation with four physical and four logical processors. The duration of the simulation for all the CFD tests are enclosed in table 3.4. It can be seen that the transient CFD3 simulation becomes prohibitively expensive with increased number of mesh cells. Owing to huge computational resource requirement, it was decided to not perform CFD3 test for Mesh 7.

Once the simulations are performed, to benchmark with the reference solution in [55], lift and drag force exerted on the entire geometry by the flow is calculated using the inbuilt post-processing tool in OpenFOAM. The lift and drag forces for the CFD1 and CFD2 tests are tabulated in table 3.5. For both CFD1 and CFD2 tests, drag and lift values monotonously approach a stationary solution with increase in mesh refinement, thereby guaranteeing a mesh independent solution. From here on, the solution of the mesh with highest mesh refinement will be defined as the most accurate solution. The most accurate solution is within 1% of the reference lift and drag values, whose accuracy can be further improved with higher mesh refinement. Therefore, it can be concluded that the flow solver has been validated for steady state simulations.

Mean, amplitude and frequency of the periodic forces are required for benchmarking with the reference data of transient CFD3 simulation. Mean is calculated from the average of maximum and minimum force in the last period. Amplitude is calculated from the difference between mean and either of maximum or minimum force in the final period. Frequency is calculated from the reciprocal of the final time period with respect to mean value. The numerical results for CFD3 test are enclosed in table 3.6.

Mesh	**CFD1**[s]	**CFD2**[s]	**CFD3**[s]
1	9	33	1543
2	33	115	6435
3	178	378	28460
4	620	1189	95522
5	1613	2336	222151
6	3462	4424	425273
7	6814	6626	–

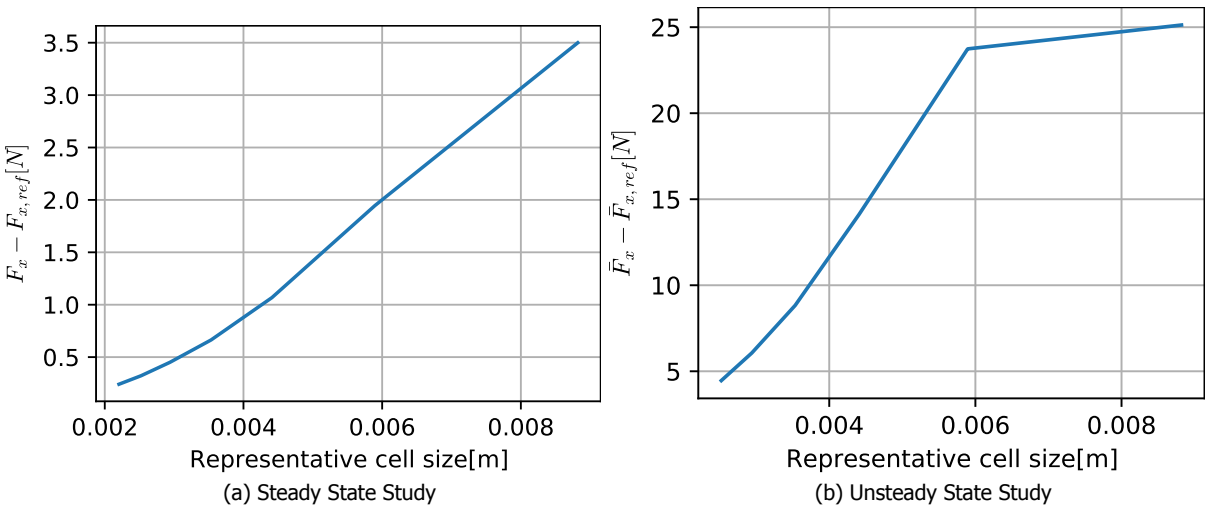
Table 3.4: Cylinder with Trailing Flap — Duration of CFD simulations

Mesh	**CFD1**		**CFD2**	
	Drag[N]	**Lift**[N]	**Drag**[N]	**Lift**[N]
1	14.2069	1.0966	140.1994	10.1370
2	14.2474	1.1065	138.6452	10.2618
3	14.2640	1.1106	137.7676	10.3286
4	14.2719	1.1127	137.3638	10.3649
5	14.2769	1.1140	137.1505	10.3937
6	14.2799	1.1148	137.0221	10.4120
7	14.2822	1.1155	136.9413	10.4281
Reference	14.29	1.119	136.7	10.53
Deviation[%]	0.0546	0.3146	0.1765	-0.9680

Table 3.5: Cylinder with Trailing Flap — Numerical results of CFD1 and CFD2 tests

Mesh	**Drag**			**Lift**			**CFL No.**
	Mean[N]	**Amp.**[N]	**Freq.**[s^{-1}]	**Mean**[N]	**Amp.**[N]	**Freq.**[s^{-1}]	
1	464.5753	3.3826	4.3120	-38.2553	351.9974	4.3130	4.9166
2	463.1878	5.5173	4.3808	-19.9316	466.6350	4.3817	7.6982
3	453.5805	5.6218	4.3969	-15.3234	462.2827	4.3969	10.3256
4	448.2779	5.6255	4.4048	-14.1977	454.4851	4.4044	12.9693
5	445.5134	5.6084	4.4089	-13.0372	450.3650	4.4089	15.5965
6	443.9086	5.6138	4.4117	-12.3057	447.9352	4.4116	18.233
Ref.	439.45	5.6183	4.3956	-11.893	437.81	4.3956	–
Dev.[%]	1.0146	-0.0801	0.3665	3.4701	2.3127	0.3640	–

Table 3.6: Cylinder with Trailing Flap — Numerical results of CFD3 test



(a) Steady State Study (b) Unsteady State Study

Figure 3.5: Cylinder with Trailing Flap — Mesh refinement study using drag and mean drag force for steady and unsteady simulations from CFD2 and CFD3 benchmarks respectively

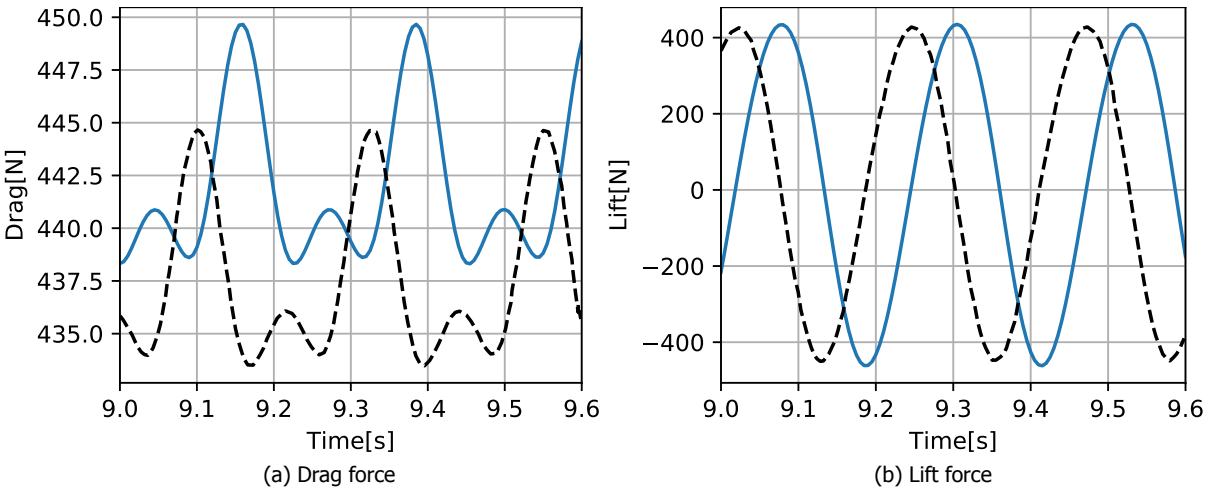
The solution for Mesh 1 seems to be an outlier in comparison to the CFD3 results for the remaining meshes. This can be attributed to poor mesh resolution to simulate vortex shedding. For the remaining meshes, all the parameters of lift and drag forces appear to approach a stationary solution, indicating that mesh independent solution can be established with further mesh refinement. The lift mean and amplitude exhibit deviation of greater than 1% from reference values. By observing the spatial convergence of lift mean and amplitude for different meshes, this deviation can be attributed to the unavailability of CFD3 results for Mesh 7. Whereas, drag mean is around 1% of the reference solution, whose accuracy can be further improved with higher mesh refinement. And, the frequency of unsteady lift and drag forces is within 1% of the reference data.

Mesh refinement studies have been done to verify if the spatial discretization error is in agreement with theoretical prediction. For the central discretization scheme, the discretization error is proportional to the square of the representative cell size. Let the error in the computed quantity be defined as $err(quantity) := computed\ value - reference\ value$. For a 2D mesh, the representative cell size can be approximated as $\Delta x \approx Mesh\ Area/\sqrt{J}$, where J is the number of cells in the fluid mesh. If measurements of a quantity are available for meshes of different refinements, then the order of discretization can be determined as:

$$O(err) := \frac{\log err_2 - \log err_1}{\log \Delta x_2 - \log \Delta x_1} \tag{3.7}$$

Drag and mean drag force from CFD2 and CFD3 test shall be employed for obtaining the order of discretization error in steady and unsteady simulations. The plots of this analysis are portrayed in figure 3.5. Similar discretization order has been observed for the six finest meshes in steady state simulation. The averaged discretization order is 2.019. For the unsteady study, similar discretization order has been observed for the four finest meshes. The O(err) value computed using the first two meshes was 0.141 which is quite different from its counterparts. This could mean that the refinement of Meshes 1 and 2 are not good enough to accurately model the vortex shedding. Therefore, the averaged discretization order computed was 1.991 using O(err) from five finest meshes. Mesh refinement study for both the steady and unsteady simulation indicated that the spatial discretization order is 2, which is in line with the expected value for the second order spatial discretization schemes mentioned in section 3.3.4.

A qualitative comparison between the most accurate transient solution and the reference data is illustrated in figure 3.6. Both the lift and drag has a phase offset with respect to the reference data, indicating that fluctuations in the flow field were triggered at different instances. For accurate modelling of the onset of vortex shedding, the spatial discretization should be very fine. Therefore, there exists a phase difference between the observed and reference lift and drag. However, the measured



(a) Drag force (b) Lift force

Figure 3.6: Cylinder with Trailing Flap — Drag and Lift Evolution from 9s to 9.6s in CFD3 test. —— : Mesh 6; – –: Reference data. Reference data digitally reconstructed from [55].

Parameters	CSM1	CSM2	CSM3
$\rho^s[10^3 \text{kg m}^{-3}]$	1	1	1
$g[\text{m s}^{-2}]$	2	2	2
$E^s[10^6 \text{kg m}^{-1} \text{s}^2]$	1.4	5.6	1.4
ν^s	0.4	0.4	0.4

Table 3.7: Cylinder with Trailing Flap — Structural simulation parameters[55]

unsteady lift and drag are qualitatively identical to that of reference ones. Owing to the qualitative and quantitative agreement of the most accurate solution with the reference solution, the flow solver is deemed to be validated for transient flow simulations. Also, this batch of flow meshes will be employed for validation of the testbed in 3.5. An earlier batch of meshes with very high aspect ratio cells in the far-stream were initially employed for flow solver validation. See appendix A for more details about the initial batch of flow meshes and the reason for its exclusion.

3.4. Structure Solver Validation

Three tests — CSM1, CSM2, CSM3 were proposed in [55] to validate the structure solver. CSM1 and CSM2 are static loading simulations, whereas CSM3 is a dynamic loading simulation. In static analysis, the rate of application of force is so small that the time dependent inertial term in the governing equation can be neglected, resulting in the equilibrium of the structure. For dynamic analysis, the force is thought to be applied instantaneously upon the commencement of simulation, leading to additional contribution from inertial term,resulting in dynamic response of the structure. For the partial structural tests, only the action of gravitational force on the beam in isolation is considered. The structure material is assumed to be a St. Venant-Kirchoff material, i.e isotropic linear elastic tensor with two independent parameters — Young's modulus [E^s] and Poisson's ratio [ν^s]. Parameters of the simulation are enclosed in table 3.7.

3.4.1. Mesh Generation

Again, `Salome` was employed to generate the mesh for the structural domain consisting of the flap behind the cylinder. Linear hexbricks with $2 \times 2 \times 2$ integration points were employed for discretizing the spatial domain to perform finite element simulations. A representative structural mesh with its elements and nodes is portrayed in figure 3.7. Seven grids with different number of quadratic hexbrick elements were generated to perform mesh independence study. Number of elements and nodes in

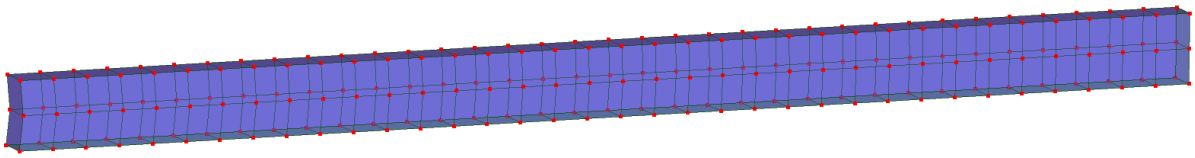


Figure 3.7: Cylinder with Trailing Flap — Structural mesh with 70 linear hexahedrons

Mesh	1	2	3	4	5	6	7
Elements	70	280	1120	4480	17920	71680	286720
Nodes	216	710	2538	9554	37026	145730	578178

Table 3.8: Cylinder with Trailing Flap — Number of structural domain elements and nodes

each mesh is tabulated in table 3.8.

3.4.2. Finite Element Solver

`CalculiX` is a finite element analysis solver based on the finite element method. Earliest known formal definition of FEM can be traced back to the work of Clough in [62]. FEM was initially developed as a computational framework to numerically obtain the solution of boundary value problems in static analysis. Primarily formulated for tackling problems in structural engineering analysis, FEM was later incorporated in other engineering disciplines like electromagnetism, heat transfer and fluid dynamics. Weak form of the conservation laws of continuum mechanics is discretized into elements, where each element is an entity of its own, and unknown kinematic quantities are sampled discretely using shape functions on the nodes of an element. Discretized conservation laws along with its appropriate constitutive additives for such individual elements, relating kinematic and kinetic quantities, are grouped into a single system of equations as:

$$\mathbf{K}\mathbf{u} + \mathbf{M}\ddot{\mathbf{u}} = \mathbf{F} \qquad [12] \tag{3.8}$$

where $\mathbf{K} := \sum_e \mathbf{L}_e^{\mathrm{T}} \mathbf{K}_e \mathbf{L}_e$; $\mathbf{M} := \sum_e \mathbf{L}_e^{\mathrm{T}} \mathbf{M}_e \mathbf{L}_e$ and $\mathbf{F} := \sum_e \mathbf{L}_e^{\mathrm{T}} \mathbf{F}_e \mathbf{L}_e$ are the aggregated stiffness matrix, mass matrix and force vector respectively[12]. Damping term is not required for the CSM analyses, and therefore is not considered here. The unknown kinematic quantity at nodes, the displacement vector $\mathbf{u}$ is obtained by employing a linear equation solver. In `CalculiX`, `SPOOLES`[7] is employed by default to obtain solution of linear equations. `SPOOLES` is an object oriented sparse matrix library developed by Ashcraft and Grimes in [63] that resolves system of linear equations directly by the factorization of the system matrix. Integral quantities in $\mathbf{K}_e, \mathbf{M}_e$ and $\mathbf{F}_e$ are obtained discretely through the weighted sum of values at $2 \times 2 \times 2$ integration points using Gaussian quadrature scheme.

3.4.3. Solver Setup

Analysis for linear hexahedral mesh with $2 \times 2 \times 2$ integration points is performed by using the `C3D8` option for mesh element type. For large deformations, quadratic terms in strain tensor cannot be ignored, resulting in the appearance of non-linear terms in (3.8). The solution of non linear equations is obtained by employing Newton Raphson iterations in `SPOOLES` solver. This is activated in `CalculiX` by employing `NLGEOM` parameter in the run file. Static analysis is triggered in `Calculix` using the `STATIC` parameter. For dynamic analysis, temporal discretization is also required. In `CalculiX`, this is accomplished through the α-method, a derivative of the Newmark implicit temporal discretization. Dynamic analysis with a time step of 0.005s is activated in `CalculiX` by employing the `DYNAMIC` parameter.

3.4.4. Numerical results for CSM simulations

The duration of the simulation for all the CSM tests are enclosed in table 3.9. Unlike the flow simulations, only 1 processor was employed for CSM tests. Owing to the nature of dynamic loading simulation, the

[7] Sparse Object Oriented Linear Equations Solver

Mesh	**CSM1**[s]	**CSM2**[s]	**CSM3**[s]
1	0.033	0.024	16.9840
2	0.095	0.068	65.7220
3	0.301	0.239	274.965
4	1.263	0.992	1157.42
5	5.981	4.900	5682.72
6	31.37	24.44	26537.7
7	169.8	131.9	151114

Table 3.9: Cylinder with Trailing Flap — Duration of CSM simulations

run time for dynamic analyses is approximately 10^3 times longer than its static counterparts.

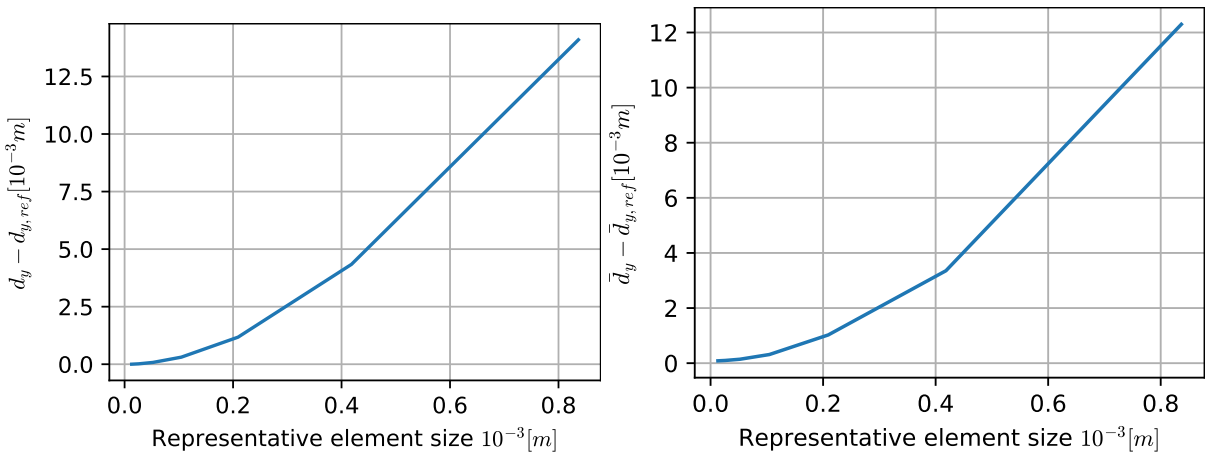
Displacement calculated at point A on the tip of flap will be compared with reference data in [55] for all the CSM tests. Numerical results for CSM1 and CSM2 tests are enclosed in table 3.10. Irrespective of number of mesh elements, displacements for both CSM1 and CSM2 tests become stationary at higher mesh refinement. Also, the most accurate result is within 1% of the reference values of CSM1 and CSM2 tests. Thus, the structural solver has been validated for static analyses.

Numerical results for the CSM3 test encompassing the mean, amplitude and frequency of displacement are tabulated in table 3.11 for the final cycle of the displacements. Again, the solution becomes stationary at higher mesh refinements. Like in the case of static analyses, the most accurate solution for CSM3 test is within 1% of the reference solution. Similar to the procedure in section 3.3.5, a mesh refinement study can be performed to determine the order of spatial discretization. The mesh refinement study for steady and unsteady simulations shall be performed with respect to the vertical displacement and mean vertical displacement from CSM2 and CSM3 benchmarks respectively. The plots of this analysis are portrayed in figure 3.8. Other than the O(err) corresponding to the last two meshes, the measured discretization orders are similar. For the steady state simulation, the averaged discretization order was found to be 2.17. In the unsteady state solution, the O(err) corresponding to the last three meshes were quite low. This means that for the temporal resolution of 0.005s might be too coarse for the structural Meshes 5 and above, serving as a bottleneck for the minimization of spatial discretization error. Perhaps, smaller time step for the above mentioned meshes can result in higher O(err). In view of the above, the averaged discretization order was computed using Meshes 1–4. It was found to be 1.764. On observing the averaged O(err) values, the discretization order of the C3D8 FEM brick discretization can be roughly concluded to be 2.

Qualitative comparison of the measured horizontal and vertical displacement with their reference counterparts in 3.9 reveal a perfect agreement without any phase offset. It is because the mesh resolution for Mesh 7 is very fine for the dimensions of the flap, thereby it is capable of accurately depicting the evolution of displacement field. Owing to qualitative and quantitative agreement of the most accurate solution with the reference data for CSM3 test, the structural solver is deemed to be validated for dynamic loading simulations. Also, this batch of structural meshes will be employed for validation of the testbed in 3.5. An earlier batch of meshes with quadratic hexbricks of 20 nodes were initially employed for flow solver validation. See appendix B for more details about the initial batch of structural meshes and the reason for its exclusion.

3.5. Testbed Validation

Three tests — FSI1, FSI2 and FSI3 were proposed in [55] to compare various FSI frameworks. The above three tests shall be replicated in the testbed described in 3.1, whose results shall be benchmarked for the purpose of validating the testbed. Both the flow and structural parameters for the above three benchmarks are mentioned in table 3.12. FSI1 simulation has a steady state solution, whereas FSI2 and FSI3 simulations have a periodic solution. Owing to a very low flow velocity at the inlet for FSI1 benchmark, the mass flow rate is not high enough to trigger the periodic motion of flap. Whereas, the mass flow rate of the fluid for FSI2 and FSI3 benchmarks are significant enough to sustain the periodic



(a) Steady State Study (b) Unsteady State Study

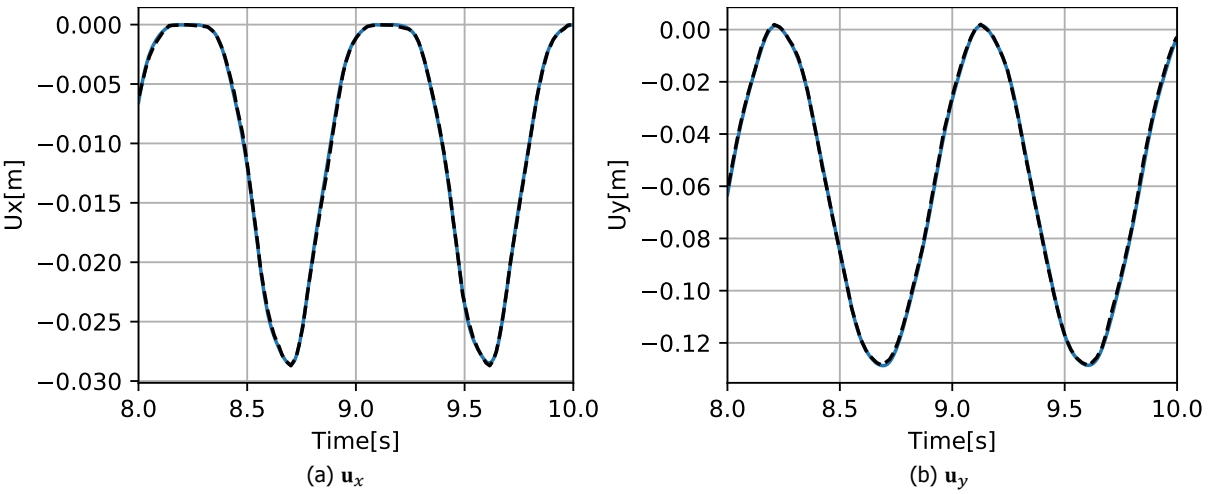
Figure 3.8: Cylinder with Trailing Flap — Mesh refinement study using horizontal and mean horizontal displacement for steady and unsteady simulations using CSM2 and CSM3 benchmarks respectively

Mesh	**CSM1**		**CSM2**	
	$\mathbf{u}_x$[mm]	$\mathbf{u}_y$[mm]	$\mathbf{u}_x$[mm]	$\mathbf{u}_y$[mm]
1	-4.4309	-52.0084	-0.2840	-13.2081
2	-6.2692	-61.7612	-0.4068	-15.8048
3	-6.9331	-64.9246	-0.4517	-16.6555
4	-7.1207	-65.7929	-0.4644	-16.8899
5	-7.1703	-66.0216	-0.4678	-16.9518
6	-7.1833	-66.0821	-0.4687	-16.9681
7	-7.1868	-66.0983	-0.4689	-16.9725
Reference	-7.187	-66.1	-0.469	-16.97
Deviation[%]	0.0026	0.0025	0.0161	0.0146

Table 3.10: Cylinder with Trailing Flap — Numerical results of CSM1 and CSM2 tests

Mesh	$\mathbf{u}_x$			$\mathbf{u}_y$		
	Mean[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]	**Mean**[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]
1	-9.0662	9.0661	1.2316	-51.3101	51.7670	1.2315
2	-12.7012	12.7014	1.1315	-60.2519	61.0584	1.1317
3	-13.8263	13.8263	1.1049	-62.5821	64.0770	1.1034
4	-14.1295	14.1291	1.0979	-63.2929	64.8053	1.0963
5	-14.2070	14.2067	1.0960	-63.4639	65.0168	1.0944
6	-14.2285	14.2280	1.0956	-63.5083	65.0687	1.0940
7	-14.2324	14.2321	1.0954	-63.5220	65.0878	1.0938
Ref.	-14.305	14.305	1.0995	-63.607	65.16	1.0995
Dev.[%]	0.5073	0.5099	0.37	-0.1337	-0.1108	-0.5163

Table 3.11: Cylinder with Trailing Flap — Numerical results of CSM3 test



(a) $\mathbf{u}_x$ (b) $\mathbf{u}_y$

Figure 3.9: Cylinder with Trailing Flap — $\mathbf{u}_x$ and $\mathbf{u}_y$ Evolution from 8s to 10s in CSM3 test. —— : Mesh 7; – – : Reference data. Reference data digitally reconstructed from [55].

Parameters	FSI1	FSI2	FSI3
$\rho^f[10^3 \text{kg m}^{-3}]$	1	1	1
$\nu^f[10^{-3} \text{m}^2 \text{s}^{-1}]$	1	1	1
$\bar{U}[\text{m s}^{-1}]$	0.2	1	2
$\rho^s[10^3 \text{kg m}^{-3}]$	1	10	1
$E^s[10^6 \text{kg m}^{-1} \text{s}^2]$	1.4	1.4	5.6
ν^s	0.4	0.4	0.4

Table 3.12: Cylinder with Trailing Flap — FSI simulation parameters[55]

motion of flap.

3.5.1. Mesh Generation

Partitioned coupling by its very nature, does not require generation of new mesh for simulating fluid structure interaction. Existing meshes that were generated during flow and structure solver validation in sections 3.3.1 and 3.4.1 shall be made use of for performing the FSI benchmarks. The nomenclature for such mesh collections shall be coined as "Mesh <x>+<y>" for the remainder of this report, where <x> corresponds to flow mesh number and <y> corresponds to structural mesh number.

3.5.2. Testbed Setup

As discussed in section 3.1.3, the multi-physics coupling library `preCICE` makes use of adapters to interface with physics solvers. Adapter configuration files for both the fluid and structure solver serve the purpose of communicating the spatial information of fluid and solid mesh boundaries neighbouring the interface as well as physical quantities required for the execution of partitioned coupling to `preCICE`. As discussed in section 2.3.1, the physical quantities required for fluid structure interaction are traction and displacement on fluid and solid boundaries adjoining the interface respectively. For the FSI benchmarks, the outer walls of flap in contact with fluid constitute the interface. As of this writing, the `OpenFOAM` adapter is capable of only coupling the force vector at interface to `preCICE`. Therefore force computed by the fluid solver shall be used in lieu of traction for performing partitioned coupling.

Another configuration file exists for the sole purpose of configuring the coupling library `preCICE` itself, which ultimately tunes the parameters required for a successful spatial and temporal coupling. The temporal coupling frameworks discussed in chapter 2 are in essence tightly coupled versions of the

Benchmark	ϵ'_{disp}	ϵ'_{force}	**Duration**[s]	**Time Step**[s]
FSI1	10^{-6}	10^{-4}	20	10^{-2}
FSI2	10^{-6}	10^{-2}	35	10^{-3}
FSI3	10^{-6}	10^{-2}	20	5×10^{-4}

Table 3.13: Cylinder with Trailing Flap — FSI simulation settings

CSS scheme, which is sequential in nature. In sequential coupling, force vector is mapped from face centres of the fluid interface to the nodes at the structure interface, and displacement vector is mapped from nodes at the structure interface to face corners of the fluid interface. Global RBF[8] interpolation with thin spline basis functions is employed for force mapping, whereas compact RBF interpolation with thin spline basis functions is employed for displacement mapping. Owing to the nature of mapped quantities involved here, the force mapping is conservative, whereas the displacement mapping is consistent. Whenever, displacement mapping fails due to the RBF linear system not converging, the displacement mapping is retried with compact-polynomial-c0 basis functions.

Based on analyses from existing literature in section 2.6.2, the IQN-ILS framework was chosen for performing temporal coupling. Instead of the absolute convergence criterion in equation 2.36, it was recommended in [64] to employ relative convergence criterion for both displacement and force vector as stopping criterion for the sub-iterations. The relative stopping criterion for displacement and force residual are mathematically described as:

$$\|\hat{\mathbf{d}}_\Gamma^i - \mathbf{d}_\Gamma^i\| < \epsilon'_{disp}\|\hat{\mathbf{d}}_\Gamma^i\| \tag{3.9}$$

$$\|\hat{\mathbf{F}}_\Gamma^i - \mathbf{F}_\Gamma^i\| < \epsilon'_{force}\|\hat{\mathbf{F}}_\Gamma^i\| \tag{3.10}$$

where $\mathbf{F}_\Gamma^i$ in the sequential coupling discussed in section 2.3.3 is described as $\mathbf{F}_\Gamma^i = F_f(\hat{\mathbf{d}}_\Gamma^i)$. For FSI1 benchmark, the relative tolerances for displacement and force were 10^{-6} and 10^{-4} respectively. Whereas relative tolerances of 10^{-6} and 10^{-2} were utilized for displacement and force residuals respectively in FSI2 and FSI3 benchmarks. Since the testing conditions for FSI1 benchmark produces steady state solution, the values for total physical time and individual time step is up to the user's discretion. Total physical time of 20s was chosen in order to ensure that there is sufficient time window for the simulation to produce steady state solution. Since the added mass effect is not significant enough for FSI1 benchmark, and the transient phase of the solution is of no interest to the user in FSI1 benchmark, one can get away with a relatively large time step for steady state solution. In view of this, the time step value for FSI1 benchmark is chosen as 10^{-2}s. In accordance with the settings in [55], the total physical time and time step for FSI2 and FSI3 benchmarks are prescribed in table 3.13.

A modified version of the original QR filter discussed in section 2.5.3, developed by Haelterman *et al.* in [38], is utilized for filtering out data that causes singularity in all the FSI benchmarks. Based on the recommendation in the user manual for `preCICE`[65], a limit of 10^{-2} was chosen for the QR2 filter. Again, in line with the recommendations in [65], the coupling is allowed to construct **V,W** matrices from a maximum of 100 previous sub-iterations or 10 previous time steps, whichever is satisfied first, in order to accelerate the convergence of the sub-iterations.

Finally, some modifications are required in the workflow of participating physics solvers in order to execute Fluid Structure Interaction.The ALE formulation of the balance laws discussed in section 2.3.2 is activated through the `dynamicMeshDict` functionality in `pimpleFoam` solver. As of this writing, the `OpenFOAM` adapter supports only `displacementLaplacian` mesh motion algorithm. Thus, it is enabled for FSI benchmarks. This results in an additional variable for the flow solver called `cellDisplacement`. This, in conjunction with the variables from coupled pressure velocity equations discussed in section 3.3.2 form a system of equations, which are solved using the same preconditioned Krylov subpace solver used in CFD solver validation. For successful spatial coupling from `preCICE` to `OpenFOAM`, the compatibility conditions discussed in section 2.3.1 have to be enforced in conjunction with DGCL. The viscous fluid compatibility condition is enforced by using `movingWallVelocity` patch for

[8]Radial Basis Function

Mesh	$\mathbf{u}_x$[mm]	$\mathbf{u}_y$[mm]	**drag**[N]	**lift**[N]
1 + 1	0.02227	0.69902	14.2085	0.80145
2 + 2	0.02236	0.78103	14.2483	0.77489
3 + 3	0.02244	0.80608	14.2656	0.75502
4 + 4	0.02247	0.81344	14.2734	0.76336
Ref.	0.0227	0.8209	14.295	0.7638
Dev.[%]	1.0097	0.9093	0.1513	0.0577

Table 3.14: Cylinder with Trailing Flap — FSI1 benchmark results

the boundary value of the flow velocity at the interface. The pseudo-structure compatibility condition for the fluid mesh is enforced through `fixedValue` patch for mesh displacements at face corners of the fluid interface. This causes the `OpenFOAM` solver to exclusively adopt values for interface mesh displacement from the mapped values in `OpenFOAM` adapter.

Only minor modification is required in the workflow of structure solver to execute Fluid Structure Interaction. For successful spatial coupling from `preCICE` to `CalculiX`, the equilibrium condition discussed in section 2.3.1 has to be enforced. It is done by making the `CalculiX` solver to exclusively adopt the mapped interface force vector from `CalculiX` adapter in the guise of concentrated loading boundary condition at the interface nodes. Other than the modifications to the workflow of the participating physics solvers, there is no need to tweak the physics solver settings that were employed in CFD and CSM benchmarks.

3.5.3. Numerical results for FSI simulations

The coupled FSI simulations were performed on a 32 processor node of a compute server with a maximum run time of 72 hours for each simulation. Due to the above computing constraints, it was only possible to perform FSI1 benchmark for four meshes, and FSI2/FSI3 benchmark for three meshes. Lift and Drag forces computed on the cylinder and flap from `OpenFOAM` in conjunction with the displacement computed on the tip of flap at point A from `CalculiX` shall be compared with their reference counterparts in [55].

The solution for FSI1 benchmark after 20 seconds is tabulated for the four sets of meshes in table 3.14. The most accurate solution for all the quantities are within acceptable bounds of the reference data. Also, for all the physical quantities, spatial convergence towards the reference data is observed. Thus, more accurate solution can be expected with a finer spatial discretization. Hence, the testbed can be deemed to pass the FSI1 benchmark.

Since FSI2 and FSI3 benchmarks produce periodic solutions, the mean, amplitude and frequency of the physical quantities are compared with their reference counterparts in [55]. Similar to the unsteady cases in CFD and CSM validation, the above three statistical quantities were computed for the final data cycle. The results of FSI2 benchmark are tabulated in table 3.15. With the exception of lift mean and amplitude, other quantities are around 5% of the reference data. Although the relative error for lift mean is very high (above 100%), it can be explained by the fact that the magnitude of mean lift is close to zero, leading to huge relative error for a negligible amount of absolute error. Also, the inability to perform simulations with meshes of finer spatial discretization hinders accuracy in this benchmark.

The results of FSI3 benchmark are tabulated in table 3.16. The frequency of displacements and forces are within 5% of the reference results. However, it is not the case with amplitude and mean values. In fact, they suffer much more in accuracy in comparison to their FSI2 counterparts. It seems that the meshes of low resolution are over predicting the lift amplitude, because the predicted amplitude decreases with increase in mesh resolution.

The qualitative comparison of computed forces and displacements with their reference counterparts for FSI2 and FSI3 benchmarks are portrayed in figures 3.10 and 3.11. From the plots it can be seen that the reference plots for FSI2 and FSI3 benchmarks were obtained for the time interval $[34, 35]s$ and $[19.5, 20]s$ respectively. Unfortunately, the FSI2 and FSI3 benchmarks for Mesh 3+3 configuration ran out of computation time before it can generate results for all the time steps in the above men-

Mesh	Axis	u			F		
		Mean[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]	**Mean**[N]	**Amp.**[N]	**Freq.**[s^{-1}]
1 + 1	X	-13.651	12.061	4.10	220.502	78.149	4.14
	Y	1.260	76.64	2.1	-0.247	322.09	2.05
2 + 2	X	-14.947	12.450	3.92	217.799	78.094	3.95
	Y	1.249	81.55	2.0	-0.528	262.99	1.96
3 + 3	X	-15.367	12.743	3.86	217.072	77.540	3.90
	Y	1.233	82.73	1.9	-0.378	251.32	1.94
Ref.	X	-14.58	12.44	3.8	208.83	73.75	3.8
	Y	1.23	80.6	2	0.88	234.2	2
Dev.[%]	X	-5.397	-2.434	-1.517	-3.947	-5.139	-2.628
	Y	-0.211	-2.644	3.104	142.9	-7.309	3.085

Table 3.15: Cylinder with Trailing Flap — FSI2 benchmark results

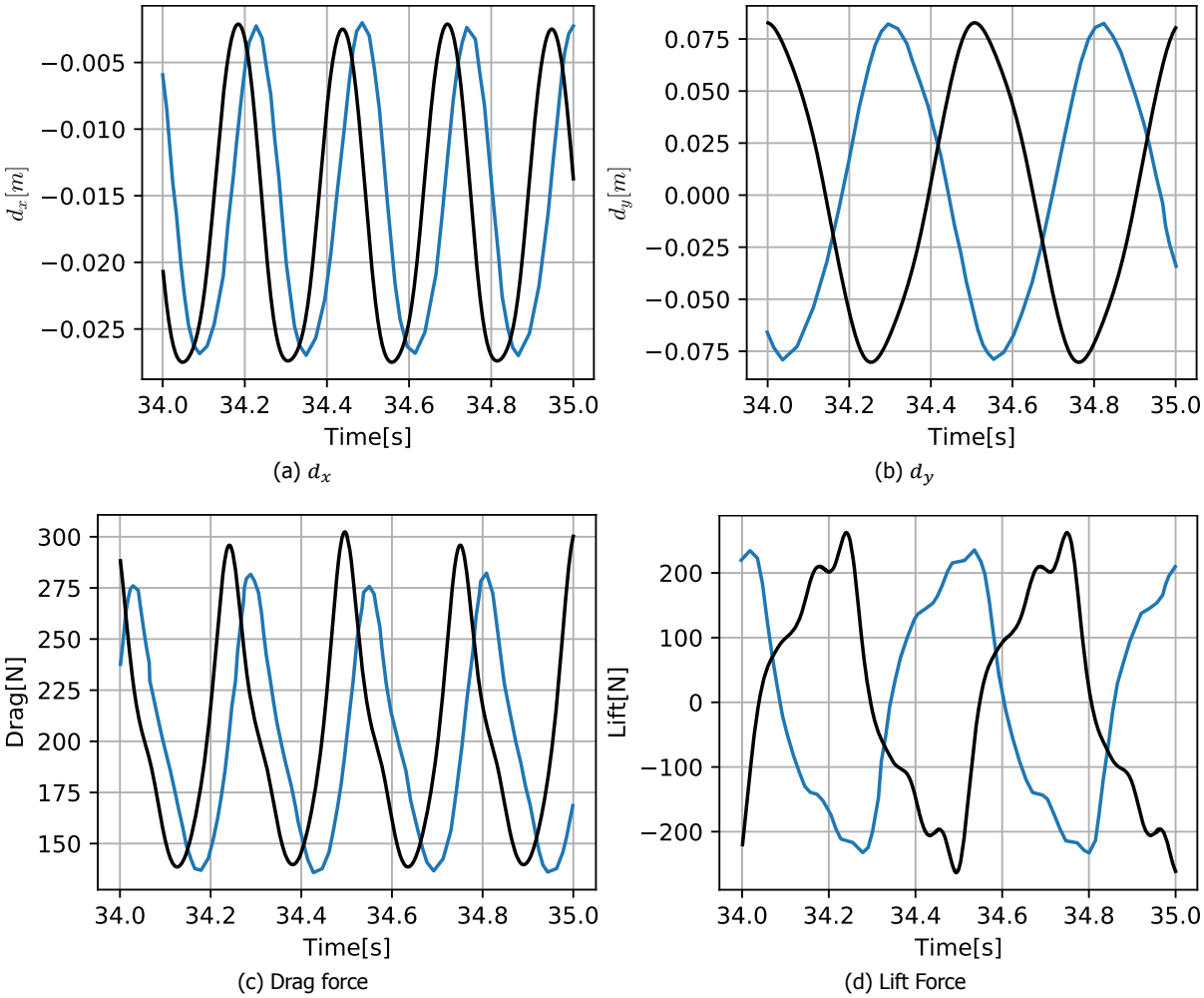
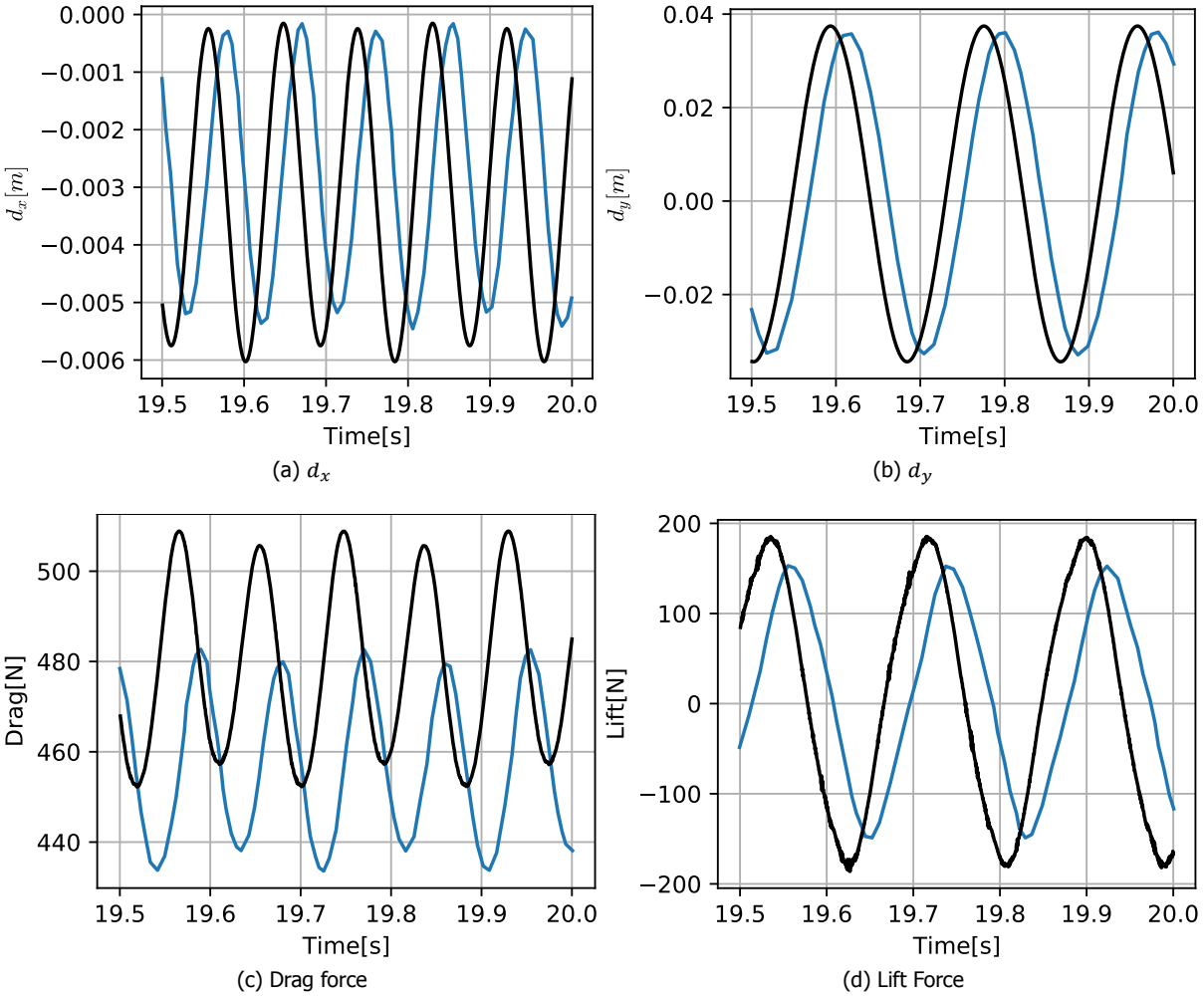


Figure 3.10: Cylinder with Trailing Flap — Comparison of displacements and forces from FSI2 benchmark between Mesh 2+2 configuration and reference data for time interval 34-35[s]. ——: Reference data; ——: Mesh 2+2 data. Reference data digitally reconstructed from [55]

Mesh	**Axis**	**U**			**F**		
		Mean[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]	**Mean**[N]	**Amp.**[N]	**Freq.**[s^{-1}]
1 + 1	X	-2.343	2.197	11.27	472.01	21.654	11.55
	Y	1.525	31.542	5.55	1.284	245.895	5.55
2 + 2	X	-2.998	2.754	10.93	481.46	24.185	11.15
	Y	1.530	35.923	5.49	1.289	182.992	5.49
3 + 3	X	-3.007	2.743	11.08	473.96	24.367	10.60
	Y	1.506	36.080	5.47	3.017	167.134	5.46
Ref.	X	-2.69	2.53	10.9	457.3	22.66	10.9
	Y	1.48	34.38	5.3	2.22	149.78	5.3
Dev.[%]	X	-11.790	-8.432	-1.682	-3.643	-7.531	2.716
	Y	-1.790	-4.944	-3.202	-35.907	-11.587	-2.989

Table 3.16: Cylinder with Trailing Flap — FSI3 benchmark results



(a) d_x (b) d_y

(c) Drag force (d) Lift Force

Figure 3.11: Cylinder with Trailing Flap — Comparison of displacements and forces from FSI3 benchmark between Mesh 2+2 configuration and reference data for time interval 19.5-20[s]. ——: Reference data; ——: Mesh 2+2 data. Reference data digitally reconstructed from [55]

tioned intervals. Therefore, the qualitative comparison was performed using results from the Mesh 2+2 configuration. The shape of the time series for the computed and reference data look similar with a phase difference between them. And on further observation of the computed lift force for the FSI3 benchmark, it seems to be affected by noise. Although the effect of this noise on the computation of lift force is trivial, it might be the case that the noise is larger for other periods, which can have an impact on the evolution of solution. Finally, based on the quantitative and qualitative analysis it can be inferred that if high resolution mesh configurations can be employed for benchmarking, then the deviation for forces is expected to reduce. Thus, it can be concluded that the test-bench is validated for FSI2 benchmark. As for the FSI3 benchmark is concerned, it can be deemed to be perfectly validated when the issue of noise can be resolved.

3.6. Conclusions

A brief summary of the physics solvers and the multi-physics coupling library, whose integration results in the testbed for performing numerical experiments pertaining to partitioned coupling, was given in section 3.1. However, the testbed has to be validated in order to ensure its proper functioning, before carrying out any research. To this end, the numerical benchmark involving a fluid structure interaction between an incompressible flow and elastic solid, proposed by Turek and Hron in [55], was chosen for validating the author's testbed in section 3.2. This numerical benchmark involves the simulation of channel flow around a cylinder with trailing flap, and is split into three components — CFD, CSM and FSI benchmarks. The flow solver `OpenFOAM` was validated with CFD tests in section 3.3, whereas the structure solver was validated with CSM tests in section 3.4. The results of the FSI benchmarking are enclosed in section 3.5. Some hindrances were faced by the author when trying to validate the testbed. Owing to the huge computational cost involved in executing a partitioned FSI coupling, the validation was possible only with a limited number of meshes in comparison to CFD and CSM benchmarks. The FSI1 benchmark results produced promising results for Mesh 4+4 configuration. However, the FSI benchmarks with periodic solutions, especially the FSI3 benchmark, were plagued with issues. It might be due to the poor resolution of the meshes in regards to the treatment of shedding vortices. At least for FSI2 benchmark, it is predicted that the simulated solution will be closer to reference solution when high fidelity meshes can be employed. Thus, the testbed can be said to be validated for FSI2 benchmark. For the testbed to be validated for FSI3 benchmark, the noise problem has to be resolved. Perhaps, it can be resolved by tweaking the various numerical parameters for temporal coupling until the noise issue has been fixed. To this end, the objective of this research shall be revised from **evaluation of robustness of IQN-ILS framework** to **Investigation and addressal of noise observed in force data from FSI3 benchmark**. With this is mind, an exhaustive study of the behaviour of noise for different types of temporal coupling settings will be performed in the next chapter.

4

Investigation of Numerical Noise in FSI3 Benchmark Forces

This chapter investigates the effect of temporal coupling settings on the numerical noise observed in the forces computed from the FSI3 benchmark in order to understand the sources that generate such noise and implement the knowledge obtained from this investigation in the noise filtering methods discussed in chapter 5. Among the FSI benchmarks with periodic results, the FSI3 benchmark has higher mass flow rate and lower flap density in comparison to the corresponding parameters for FSI2 benchmark. As discussed in section 2.6.1, higher added mass of fluid in combination with lower mass of the flap results in stronger coupling for FSI3 benchmark in comparison to FSI2 benchmark. Strongly coupled FSI problems are more prone to stability issues, which might explain the prevalence of noise for the force data from FSI3 benchmark.

Before initiating the noise investigation study, it is beneficial to identify a mathematical function that indicates the degree of smoothness/roughness of a signal. To this end, a mathematical framework based on total variation was used to perform this task. This framework is described in detail in section 4.1. The hypothesis for the reason behind noise formation in force date is given in section 4.2. The effect of coupling tolerances as well as IQN-ILS framework settings on noise formation are discussed in sections 4.3 and 4.4 respectively. Coupled simulations with displacement extrapolation were investigated in section 4.5 to check if reduced number of sub-iterations results in noisier force signal. In section 4.6, the role of proportion of degrees of freedom between the fluid and solid interface on the level of noise is discussed. And, coupled simulations were executed with Aitken's and fixed under-relaxation frameworks in section 4.7 to check whether the noisy force signal is specific to IQN-ILS framework or not. Finally, with the knowledge obtained from the above discussed sections, coupled simulation with the best settings for IQN-ILS framework shall be executed in section 4.8.

4.1. Total Variation Method

Consider a time dependent periodic signal $\mathbf{g}$ with $r+1$ terms, i.e:

$$\mathbf{g} := [g_0 \quad g_1 \quad g_2 \quad \dots \quad g_{r-1} \quad g_r] \tag{4.1}$$

Let the first instance of maximum and minimum values occur at l^{th} and m^{th} time step respectively. Let the vector of these maximum and minimum values in addition to the initial and final value of the periodic signal be named as $\mathbf{g}_{ext}$, i.e.:

$$\mathbf{g}_{ext} := [g_0 \quad g_l \quad g_m \quad g_{l+w} \quad g_{m+w} \quad g_{l+2w} \quad g_{m+2w} \quad \dots \quad g_r] \tag{4.2}$$

The actual total variation is defined as the sum of absolutes of differences of $\mathbf{g}$, i.e.:

$$\sum |\mathbf{\Delta g}| := [|g_1 - g_0| + |g_2 - g_1| + \quad \dots \quad + |g_r - g_{r-1}|] \tag{4.3}$$

It is equivalent to measuring the vertical distance traversed by a signal. Also, the expected total variation is defined as the sum of absolutes of differences of $\mathbf{g}_{ext}$, i.e.:

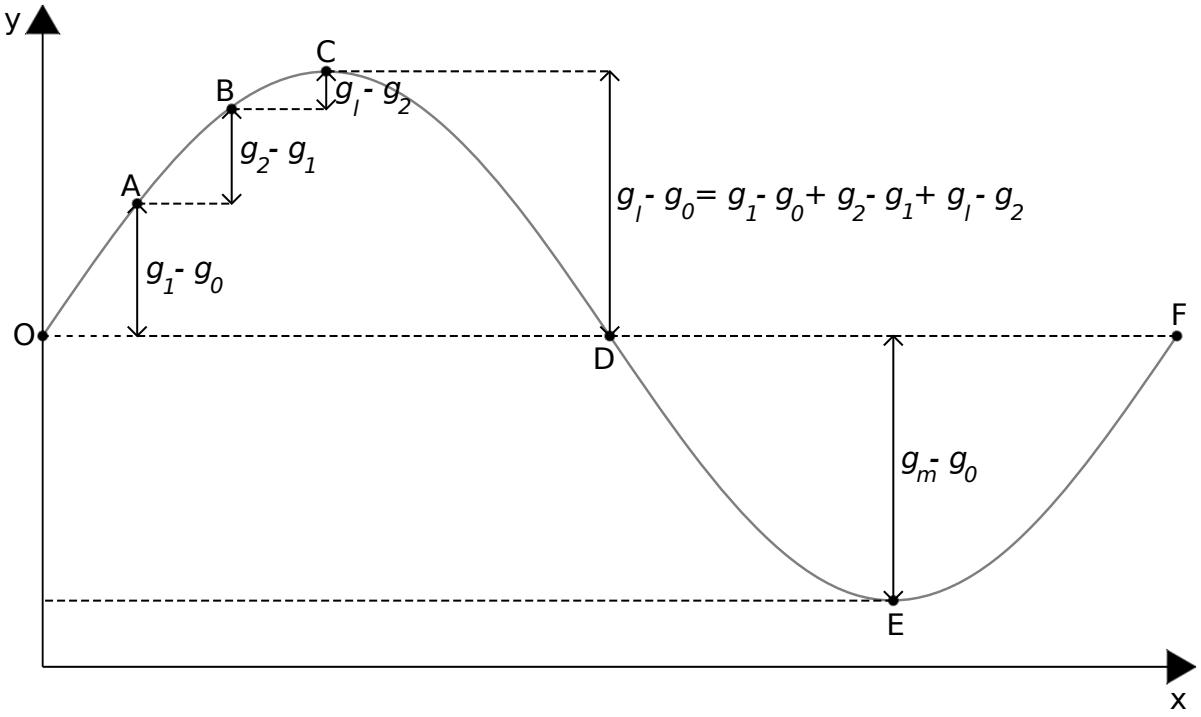


Figure 4.1: Total Variation Method — Illustration of actual total variation and expected total variation

$$\sum |\mathbf{\Delta g}_{ext}| := [|g_l - g_0| + |g_m - g_l| + |g_{l+w} - g_m| + \quad \dots \quad + |g_{m+Kw} - g_{l+Kw}| + |g_r - g_{m+Kw}|] \quad (4.4)$$

where K is number of periods in the signal. The expected total variation is an alternative interpretation of measuring the vertical distance traversed by the signal. Therefore, for a smooth signal where it traverses monotonously between the extreme values, the actual total variation and expected total variation will be the same, as illustrated in figure 4.1 for the initial quarter period of a sine wave, where $g_{m-1} < g_m \ \forall \ m \in [t_0, t_l]$. For a signal afflicted with noise, where the noise causes the signal to vary non monotonously between extreme values, then the actual total variation will be higher than the expected total variation. Thus, the difference of actual total variation and expected total variation can serve as an indicator for the amount of noise in the signal. This function can be defined as:

$$\Theta(\mathbf{g}) := \sum |\mathbf{\Delta g}| - \sum |\mathbf{\Delta g}_{ext}| \quad (4.5)$$

This function shall be named as smoothness function. If the signal has no noise as illustrated in figure 4.1, then the smoothness function for that signal will be zero. If the signal is noisy, then the smoothness function will have a positive value. And, the value of smoothness function is proportional to the amount of noise in the signal. This function can detect noise that is wiggly in nature, i.e. the local slope change by the noise results in non-monotonous curve. For the noise of the staircase type, i.e. the local slope change by the noise still results in monotonous curve, then the smoothness function cannot detect it. From the magnified version of a lift plot from a random FSI3 test in figure 4.3b , it can be seen that the typical signature of noise consists of wiggles, which can be detected by the smoothness function. Thus, the smoothness function will be a suitable mathematical function for determining the smoothness of force data from coupled FSI simulations.

4.1.1. Sensitivity to noise magnitude

Consider force values at two consecutive time steps in a noisy signal as F_1 and F_2. These forces can be decomposed into its physical value and noise as:

$$F_1 := {F_1}^{\dagger} + {F_1}^{\ddagger}$$
$$F_2 := {F_2}^{\dagger} + {F_2}^{\ddagger} \tag{4.6}$$

where $F^{\dagger}$ is the physical value and $F^{\ddagger}$ is the noise. Thus the variation for that time interval can be defined as:

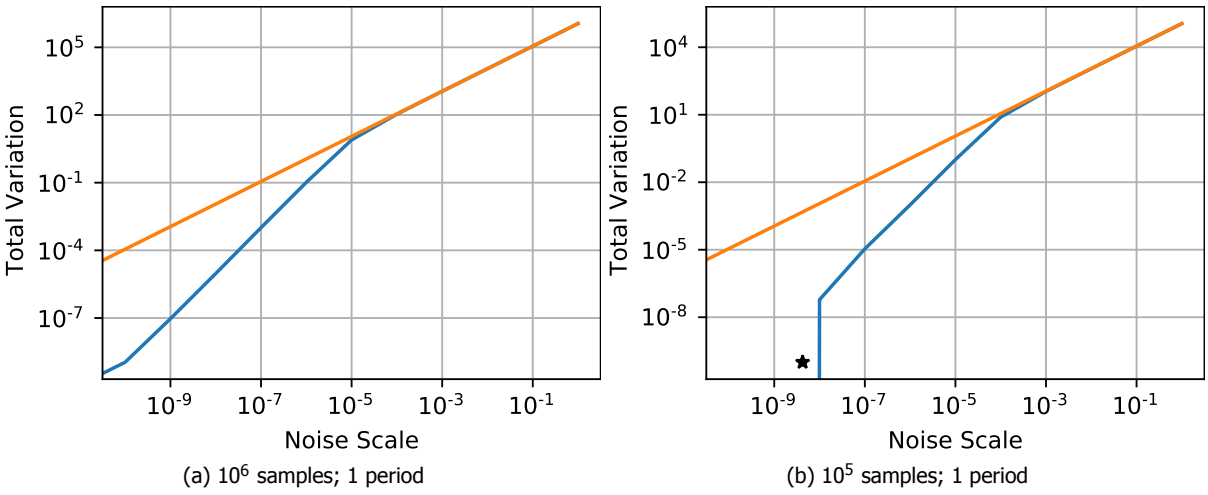
$$\Delta F := \Delta F^{\dagger} + \Delta F^{\ddagger} \tag{4.7}$$

Since the smoothness function is a relative parameter, in order for it to generate a meaningful value, the variation due to noise i.e. $\Delta F^{\ddagger}$ should at least be of similar order of magnitude as the variation due to actual physical value i.e. $\Delta F^{\ddagger}$. The noise variation in turn is dependent on magnitude of noise. Thus, the smallest noise that can be detected by the total variation method has to be estimated in order to ensure that the ignored noise is trivial enough for smoothness analysis. To this end, total variation method was performed on a single cycle of sine wave. A sine series of 10^6 samples has been created. Sine wave was chosen for this analysis because the force plots observed in section 3.5.3 are sinusoidal in nature. For a fixed time period, higher number of samples result in smaller variation for a time step. Also, from figure 4.1, by measuring the vertical distance of curve segments OA,AB,BC it can be seen that the variation becomes smaller as the curve approaches maxima or minima. Therefore, for total variation method to function, it is expected that the minimum variation of noise (${\Delta F^{\ddagger}}_{min}$) is of similar order of magnitude as minimum variation of actual value (${\Delta F^{\dagger}}_{min}$), which occurs at signal minima or maxima. A vector with 10^6 random values that are of order of magnitude of O(1) was generated. This vector was then scaled by amplitudes ranging between 10^{-10} and 10^{10} in order to generate noise vectors of different magnitudes. In addition to the original sine series of 10^6 samples, other series of 10^3, 10^4, 10^5 terms were obtained as subsets of the original series with larger spacing.

Let $\mathbf{g}^{\dagger}$ be the sine series and $\mathbf{g}^{\ddagger}$ be the scaled noise series with amplitude ξ. Noise afflicted sine wave is generated by adding $\mathbf{g}^{\dagger}$ and $\mathbf{g}^{\ddagger}$. The smoothness function values as well as the total variation due to noise are plotted for the noise afflicted sine wave with different number of samples in figures 4.2a – 4.2d. As expected, for a fixed period the total variation method can detect smaller noise as the number of samples in a signal increases. For the signal with 10^6 samples, the total variation method was able to compute a meaningful smoothness function value for noise of all orders of magnitude involved in the analysis. Total variation method on $\mathbf{g}_{10^5}$, $\mathbf{g}_{10^4}$ and $\mathbf{g}_{10^3}$ is sensitive from ξ_{min} of 10^{-8}, 10^{-6} and 10^{-4}. Also, in the above mentioned figures, the total variation due to noise and smoothness function value start to become similar when the magnitude of noise vector increases. This is expected, since for noise with large ξ, $\sum|\mathbf{\Delta g}^{\ddagger}| >> \sum|\mathbf{\Delta g}^{\dagger}|$, resulting in $\Theta(\mathbf{g}) \approx \sum|\mathbf{\Delta g}^{\ddagger}|$. Since, the sensitivity of the total variation method was examined for discrete values of ξ at 10^{-10}, 10^{-9}, 10^{-8} etc.; it could be the case that the actual value of ξ_{min} lies between the discrete values. Therefore an optimization study was performed for $\mathbf{g}_{10^5}$, $\mathbf{g}_{10^4}$ and $\mathbf{g}_{10^3}$ to determine ξ_{min} that corresponds to $\Theta(\mathbf{g}) > 10^{-10}$. These values are tabulated in the first three rows of table 4.1. These values were also highlighted in the figures 4.2b – 4.2d.

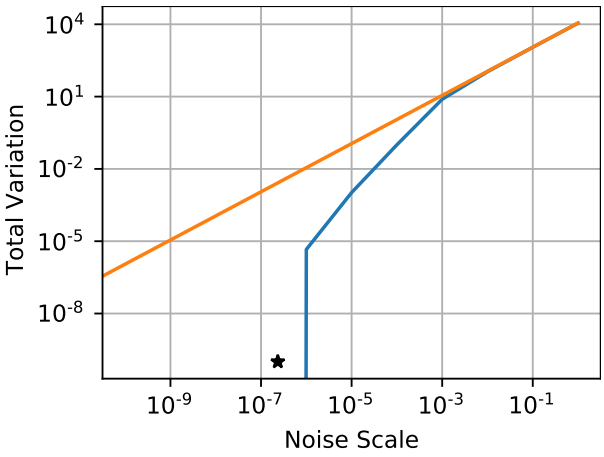
4.1.2. Noise threshold analysis for FSI3 benchmark force data

In order to apply the findings from this section to smoothness analysis of force data, the smoothness function was again evaluated for noise afflicted sine wave with 10^4 samples, but with different number of cycles. From the force results for FSI3 simulation of Mesh 2+2 configuration in table 3.16, with time step being 5×10^{-4}s, and drag and lift frequency approximately being 11 and 5.5 respectively; the number of cycles required to mimic drag and lift signal for 10^4 samples (t=5s) are 55 and 28 respectively. Therefore, two additional noise afflicted sine wave was constructed with 55 and 28 cycles. The steps followed for smoothness function evaluation of single period noise afflicted sine wave was repeated for the above two mentioned signals. The smoothness function and total noise variation plot for these signals are enclosed in figures 4.2e and 4.2f. It can be seen that ξ_{min} increases with higher number of cycles in a signal. This is expected, since increasing the number of cycles for a fixed time window results in larger variation of physical value for a time step, thereby requiring noise variation of larger magnitude to be detected by total varaition method. From table 4.1, it can be seen that total variation method is sensitive to noise that is at least 2.3204×10^{-4} times drag amplitude and 9.2837×10^{-5} times lift amplitude for smoothness analyses of drag and lift data respectively from FSI3 benchmark of mesh

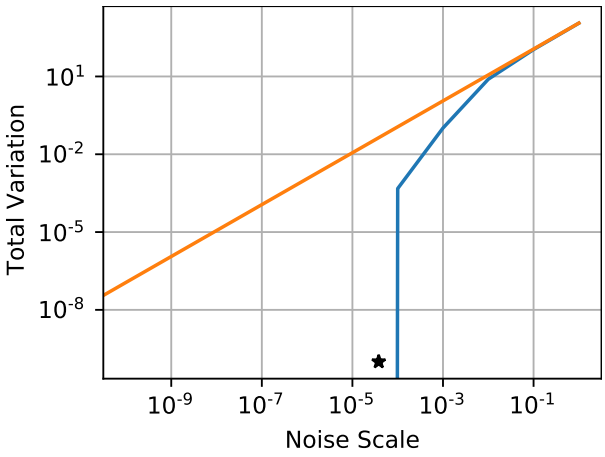


(a) 10^6 samples; 1 period

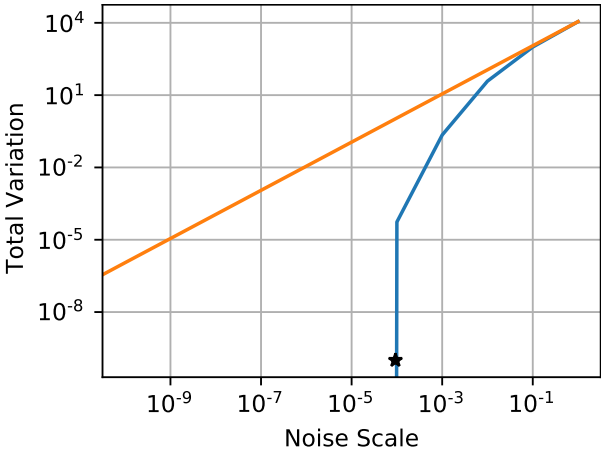
(b) 10^5 samples; 1 period



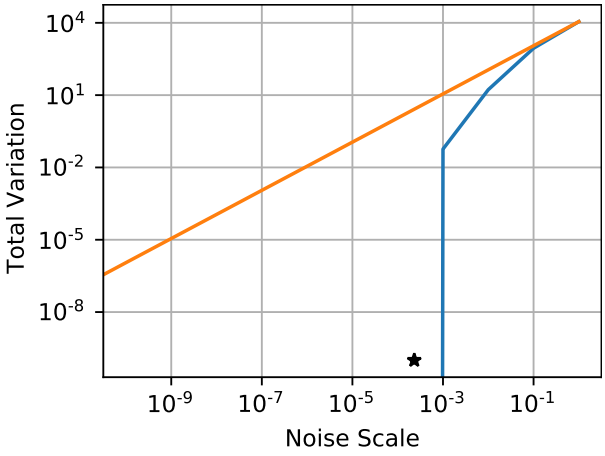
(c) 10^4 samples; 1 period



(d) 10^3 samples; 1 period



(e) 10^4 samples; 28 periods



(f) 10^4 samples; 55 periods

Figure 4.2: Total Variation Method — Smoothness function value and Total Variation of noise for a sine wave afflicted with noise of different scales/amplitude. — : Smoothness function value; — : Total variation of noise vector; ⋆ : Minimum noise amplitude sensitive to TVM

Signal	Minimum Noise Amplitude
10^5 samples; 1 period	4.2250×10^{-9}
10^4 samples; 1 period	2.3487×10^{-7}
10^3 samples; 1 period	3.8106×10^{-5}
10^4 samples; 28 periods	9.2837×10^{-5}
10^4 samples; 55 periods	2.3204×10^{-4}

Table 4.1: Total Variation Method — Minimum noise sensitive to TVM for sine series of different number of samples with one period. Series with 10^4 samples is also tested with 28 and 55 periods to determine the sensitivity of TVM to lift and drag force from FSI3 benchmark

2+2 configuration. From table 3.16, the drag and lift amplitude were approximately 24N and 183N respectively. Therefore, using total variation method, it is possible to perform smoothness comparisons with drag and lift noise being as low as 5.569×10^{-3}N and 0.017N respectively for any time step.

4.2. Hypothesis for noise formation

As mentioned in previous section, the zig-zag pattern of noise is its characteristic signature. Thus, the frequency of this zig-zag corresponds to the time step of the simulation, indicating that the noise is temporal in nature. The study of mechanisms involved in spatial and temporal coupling is a good starting point to understand how noise can creep into force. Interface force is computed in flow solver as a function of pressure and wall shear stress at the interface. From (2.17), the shear stress vector for Newtonian fluid at interface of cell i is:

$$\boldsymbol{\tau}_{\Gamma(i)} = \mu(\boldsymbol{\nabla}\mathbf{v}_i + \boldsymbol{\nabla}\mathbf{v}_i^T) \cdot \mathbf{n}_{\Gamma(i)} \tag{4.8}$$

where $\mathbf{n}_{\Gamma(i)}$ is the normal vector of the face of cell i corresponding to the interface. The wall shear stress can be obtained as:

$$\tau_{\Gamma(i),w} = \boldsymbol{\tau}_{\Gamma(i)} \cdot \mathbf{t}_{\Gamma(i)} \tag{4.9}$$

where $\mathbf{t}_{\Gamma(i)}$ is the tangential vector of the face of cell i. The force at face of cell i is then computed as:

$$\begin{aligned} \mathbf{F}_{\Gamma(i)} &= p_i \mathbf{dA}_{\Gamma(i)} + \tau_{\Gamma(i),w} \mathbf{dA}_{\Gamma(i)} \\ \mathbf{F}_{\Gamma(i)} &= p_i \mathbf{dA}_{\Gamma(i)} + \mu(\boldsymbol{\nabla}\mathbf{v}_i + \boldsymbol{\nabla}\mathbf{v}_i^T) \cdot \mathbf{n}_{\Gamma(i)} \cdot \mathbf{t}_{\Gamma(i)} \mathbf{dA}_{\Gamma(i)} \end{aligned} \tag{4.10}$$

Thus it can be seen that the force at the wall is a function of velocity gradients at the interface. Noise, observed in the force is temporal in nature. (4.10) does not seem to be a explicit function of time. But the velocity gradient of interface cell i is a function of velocity at its faces. Since, one of the faces is the fluid-structure interface, the velocity gradient is a function of interface velocity, i.e.:

$$\boldsymbol{\nabla}\mathbf{v}_i = \mathbf{F}(\mathbf{v}_{\Gamma(i)}) \tag{4.11}$$

From the kinematic condition at the interface in (2.20):

$$\mathbf{v}_{\Gamma(i)} = \dot{\mathbf{d}}_{\Gamma(i)} \tag{4.12}$$

Substituting (4.11) and (4.12) in (4.10):

$$\mathbf{F}_{\Gamma(i)} = p_i \mathbf{dA}_{\Gamma(i)} + \mu(\mathbf{F}(\dot{\mathbf{d}}_{\Gamma(i)})) \cdot \mathbf{n}_{\Gamma(i)} \cdot \mathbf{ta}_{\Gamma(i)} \mathbf{dA}_{\Gamma(i)} \tag{4.13}$$

Now, interface force can be seen as a function of derivative of interface displacement. Thus, if noise occurs in interface force, the only source for it is the derivative of interface displacement, assuming the flow solver is perfectly capable of computing the other quantities in (4.10) without noise. And, if

the interface displacement has noise, then the derivative of interface displacement has noise. Let, the interface displacement be decomposed as:

$$\mathbf{d}_{m,\Gamma} = \mathbf{d}^{\dagger}_{m,\Gamma} + \mathbf{d}^{\ddagger}_{m,\Gamma} \tag{4.14}$$

On applying BDF-2 temporal discretization to interface displacement:

$$\dot{\mathbf{d}}_{m,\Gamma} = \frac{\frac{3}{2}\mathbf{d}^{\dagger}_{m,\Gamma} - 2\mathbf{d}^{\dagger}_{m-1,\Gamma} + \frac{1}{2}\mathbf{d}^{\dagger}_{m-2,\Gamma}}{\Delta t} + \frac{\frac{3}{2}\mathbf{d}^{\ddagger}_{m,\Gamma} - 2\mathbf{d}^{\ddagger}_{m-1,\Gamma} + \frac{1}{2}\mathbf{d}^{\ddagger}_{m-2,\Gamma}}{\Delta t} \tag{4.15}$$

For FSI benchmark $\Delta t = 0.0005$. Thus:

$$\dot{\mathbf{d}}_{m,\Gamma} = 2\times 10^{3}\left(\frac{3}{2}\mathbf{d}^{\dagger}_{m,\Gamma} - 2\mathbf{d}^{\dagger}_{m-1,\Gamma} + \frac{1}{2}\mathbf{d}^{\dagger}_{m-2,\Gamma}\right) + 2\times 10^{3}\left(\frac{3}{2}\mathbf{d}^{\ddagger}_{m,\Gamma} - 2\mathbf{d}^{\ddagger}_{m-1,\Gamma} + \frac{1}{2}\mathbf{d}^{\ddagger}_{m-2,\Gamma}\right) \tag{4.16}$$

From the above equation, it can be seen that the first derivative of noise $\mathbf{d}^{\ddagger}_{m,\Gamma}$ is scaled by three orders of magnitude, which is a significant value. This implies that minuscule error in interface displacement on differentiation manifests as significant noise in force data. It is the reason, why the noise was only observed in interface force in figure 3.11. As for the noise in interface displacement is concerned, it can be caused by round-off error, modelling error, iterative error, discretization error etc., i.e.:

$$\mathbf{d}^{\ddagger}_{m,\Gamma} = \mathbf{d}^{\ddagger}_{m,\Gamma,round-off} + \mathbf{d}^{\ddagger}_{m,\Gamma,model} + \mathbf{d}^{\ddagger}_{m,\Gamma,discretization} + \mathbf{d}^{\ddagger}_{m,\Gamma,iterative} + \mathbf{d}^{\ddagger}_{m,\Gamma,misc} \tag{4.17}$$

4.2.1. Round-Off Error

Round-off error is caused as a consequence of machine precision. Machine precision is the smallest number that can be represented by a computer. This causes the accuracy of the solution to be limited by the machine precision. This error only comes into picture, when the other sources of numerical error have been addressed.

4.2.2. Modelling Error

Modelling error occurs when the model approximates certain aspects of the physical phenomenon. Non linear version of the structural governing equations were employed in the `CalculiX` solver. And, the St.Venant Kirchoff's model for the constitutive relation is an appropriate model for modelling isotropic elastic material. Thus, it can be concluded that modelling error is negligible in $\mathbf{d}^{\ddagger}_{m,\Gamma}$.

4.2.3. Discretization error

Discretization error is a combination of spatial and temporal discretization error within the structure solver and spatial coupling error from RBF interpolation when the interface displacement is coupled from the structure domain to fluid domain. However, the error arising from the RBF interpolation is ultimately a function of spatial discretization error. Therefore error generated from RBF interpolation can be categorized under $\mathbf{d}^{\ddagger}_{m,\Gamma,spatial-discretization}$. Within the structure solver, the spatial discretization error in interface displacement is independent of time step, if the time step is sufficiently small. The mesh refinement study in section 3.4.4 showed that the spatial discretization is of second order. And, the alpha method employed for temporal discretization is second order accurate. Therefore the derivative of discretization error is:

$$\dot{\mathbf{d}}^{\ddagger}_{m,\Gamma,discretization} = \frac{O(\Delta t^2)}{\Delta t} + \frac{O(\Delta x^2)}{\Delta t} \tag{4.18}$$

$\mathbf{d}^{\ddagger}_{m,\Gamma,temporal-discretization}$ decreases with smaller time step. However, by reducing the time step, the error from spatial discretization and other sources are amplified. If the temporal discretization error is much smaller than the spatial discretization error and iterative error to begin with, than further reduction in Δt would only amplify the noise in interface force.

$\epsilon'_{\mathbf{d}}$	$\epsilon'_{\mathbf{F}}$	$\bar{k}$	$\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{d}} \rVert}$	$\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{F}} \rVert}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
10^{-6}	10^{-2}	4.9	1.893×10^{-7}	3.312×10^{-3}	568.651	15344.922
10^{-4}	10^{-4}	8.9	1.551×10^{-8}	4.328×10^{-5}	79.417	1696.299

Table 4.2: Cylinder with Trailing Flap — Mean number of sub-iterations, Mean interface residuals and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 1+1 configuration with different coupling tolerances for time interval 15 - 20 [s]

4.2.4. Iterative error

Iterative errors are associated with solutions obtained using iterative methods. In partitioned FSI, iterative methods are employed at two levels. At the first level, iterative methods like fixed point and Newton-Raphson iterations are used to treat the partitioning error. At the second level, iterative methods within the flow and structure solver treat the flow field and structure field error respectively. In applications related to development of partitioned coupling methods, the iterative error in fluid and structural solvers are kept low in comparison to the iterative error of the partitioning framework. Therefore, in this research $\mathbf{d}^{\ddagger}_{m,\Gamma,iterative}$ is dominated by the partitioning error.

Miscellaneous error

Any sources of error other than the four mentioned sources shall be categorized as $\mathbf{d}^{\ddagger}_{m,\Gamma,misc}$.

4.2.5. Synthesis

From the discussion so far in this section, it can be seen that for small temporal and spatial discretization in the structure solver, the dominant error in $\mathbf{d}^{\ddagger}_{m,\Gamma}$ is expected to be the iterative error. And, for this research the dominant error in iterative error is the partitioning error. However, the miscellaneous error cannot be predicted a priori. Therefore, the remainder of this chapter will focus on finding the optimal parameters for IQN-ILS algorithm that can produce the least partitioning and miscellaneous error, thereby minimizing the occurrence of noise in interface force.

4.3. Effect of coupling tolerances

Tweaking the coupling tolerances of temporal coupling is the fundamental way to control the iterative error. If convergence is possible, then a stricter coupling tolerance will reduce the partitioning error. Since `preCICE` also supports CPS scheme, one can assign values for both ϵ'_{disp} and ϵ'_{force}. Since, the implicit temporal coupling is sequential, ϵ'_{force} is not expected to affect the partitioning error. Only ϵ'_{disp} will play a main role in affecting partitioning error. The amount of noise with respect to ϵ'_{disp} shall be measured.

4.3.1. Mesh 1+1 configuration

FSI 3 benchmark was repeated for Mesh 1+1 with $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$, which is compared with the original results obtained using $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-2}$. Results for $t = [15, 20]s$ of this comparison are enclosed in table 4.2. It can be seen that the simulation with higher $\epsilon'_{\mathbf{d}}$ has much lower noise. This is because $\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{d}} \rVert}$ is an order of magnitude smaller for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ in comparison to $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-2}$. The lesser $\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{d}} \rVert}$ is possible because of stricter $\epsilon'_{\mathbf{F}}$ in $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$. Thus, it can be seen that a stricter $\epsilon'_{\mathbf{F}}$ can indirectly influence $\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{d}} \rVert}$ by forcing the temporal coupling framework to perform more sub-iterations. However, the test for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ required nearly double the amount of sub-iterations in comparison to $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-2}$. Therefore for Mesh 1+1 configuration, the cost for obtaining $\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{d}} \rVert}$ that is one order of magnitude lower is twice the number of sub-iterations. Still, converging to a smaller interface residual resulted in lesser noise, which is in line with the hypothesis.

4.3.2. Mesh 2+2 configuration

FSI 3 benchmark was repeated for Mesh 1+1 with $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$, which is compared with the original results obtained using $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-3}$. Results for the $t = [11.445, 16.445]s$ of this comparison are enclosed in table 4.3. In this case, the comparison is for an earlier time interval

$\epsilon'_{\mathbf{d}}$	$\epsilon'_{\mathbf{F}}$	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{y,\Gamma})$
10^{-6}	10^{-3}	6.8	8.705×10^{-8}	4.124×10^{-4}	164.547	13687.118
10^{-4}	10^{-4}	12.5	3.447×10^{-7}	1.118×10^{-4}	12100.711	223963.376

Table 4.3: Cylinder with Trailing Flap — Mean number of sub-iterations, Mean interface residuals and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2+2 configuration with different coupling tolerances for time interval 11.445 - 16.445 [s]

because the FSI3 benchmark with $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ ran out of computation time. The results observed for Mesh 2+2 configuration are in contrast to the results for Mesh 1+1 configuration. Noise observed for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ is much larger than noise observed for $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-3}$. $\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$ is much higher for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ in comparison to $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-3}$. This is because $\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|} = 1.118 \times 10^{-4}$ for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ is greater than the mandated $\epsilon'_{\mathbf{F}} = 10^{-4}$. This implies that the sub-iterations did not converge for some time steps. Even if the sub-iterations have not converged, $\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|} = 3.447 \times 10^{-7}$, which is still lower than the $\epsilon'_{\mathbf{d}}$ for the simulation with $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-3}$ that produced less noise.

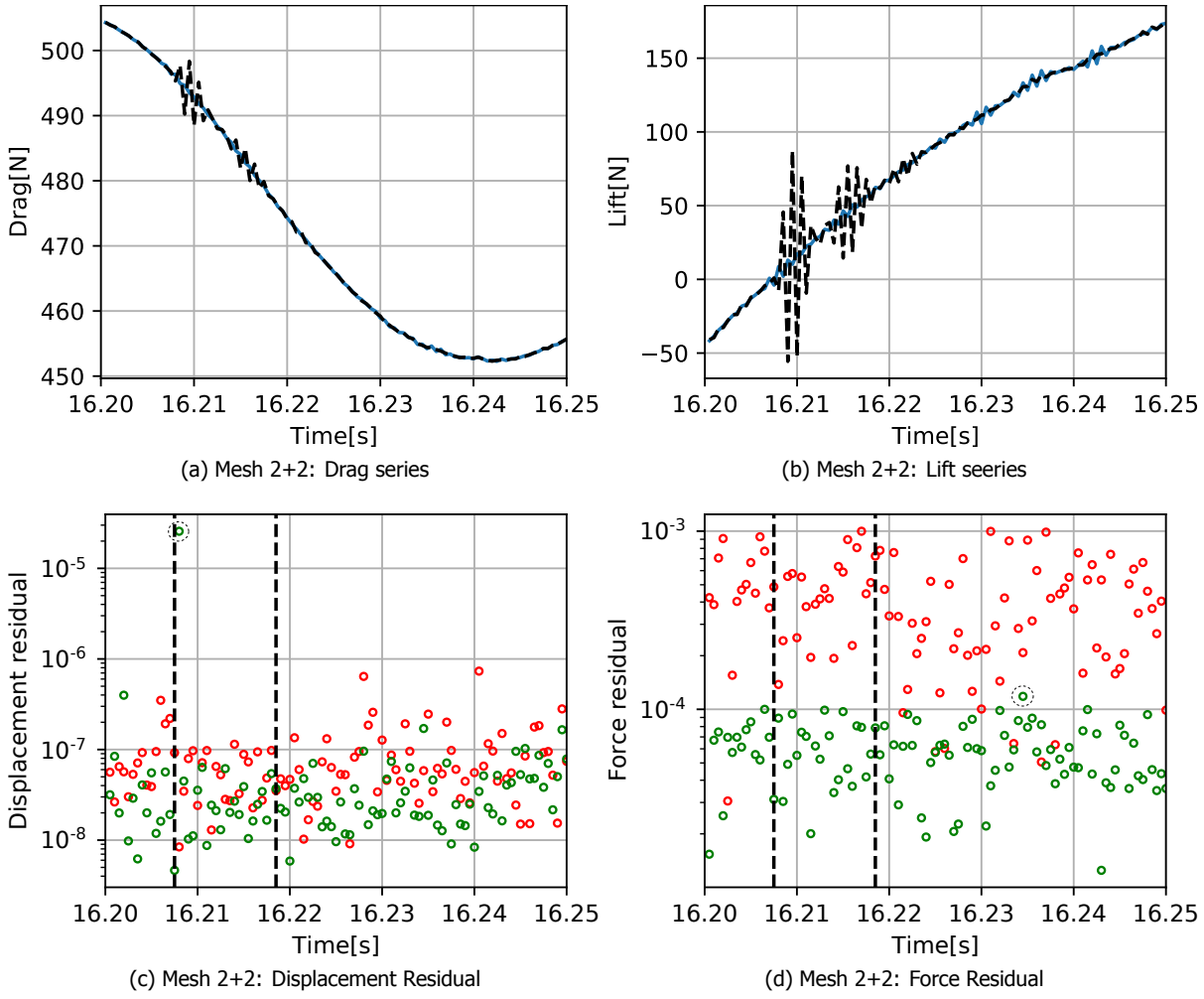
This warranted a detail investigation. The forces for both the simulations are plotted for $t = [16.2, 16.25]s$ in figures 4.3a and 4.3b. The drag and lift data for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ is plagued with huge numerical artefacts for $t = [16.2075, 16.2185]s$. To understand more about these artefacts the scattered plots for $\mathbf{r}_{\Gamma,\mathbf{d}}$ and $\mathbf{r}_{\Gamma,\mathbf{F}}$ for $t = [16.2, 16.25]s$ are shown in figures 4.3c and 4.3c. Within these plots, the interval corresponding to $t = [16.2075, 16.2185]s$ are bounded by shaded black lines. Both the scatter plots indicate that the $\mathbf{r}_{\Gamma,\mathbf{d}}$ and $\mathbf{r}_{\Gamma,\mathbf{F}}$ are generally lower for $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$ in comparison to $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-3}$. However, for the time step just after $t = 16.2075[s]$, $\mathbf{r}_{\Gamma,\mathbf{d}}$ is of the order of 10^{-5}, which is highlighted in the plot. Since it is lower than $\epsilon'_{\mathbf{d}}$ for the simulation with $\epsilon'_{\mathbf{d}} = 10^{-4}$, $\epsilon'_{\mathbf{F}} = 10^{-4}$, the temporal coupling deemed that time step to be converged and proceeded to the next time step. This is a special event, where a stricter $\epsilon'_{\mathbf{F}}$ did not force the temporal coupling to perform additional sub-iterations. Consequentially, it can be clearly seen from the force plots that for time step at $t = 16.2075[s]$, where $\mathbf{r}_{\Gamma,\mathbf{d}}$ is in the order of 10^{-5}, noise is triggered in both drag and lift data. This is because higher $\mathbf{r}_{\Gamma,\mathbf{d}}$ amounts to higher partitioning error, and the partitioning error further gets amplified under temporal derivative in the guise of iterative error. Thus, this observation is in line with the hypothesis.

4.3.3. Synthesis

Based on the results for investigation of coupling tolerances on Mesh 1+1 and Mesh 2+2 configurations, it can be concluded that smaller $\mathbf{r}_{\Gamma,\mathbf{d}}$ results in lesser noise in the force data. And, no relationship can be observed between $\mathbf{r}_{\Gamma,\mathbf{F}}$ and noise in the force data. Thus, manipulation of $\epsilon'_{\mathbf{d}}$ is the reliable way to control the amount of noise in force data. Therefore, stricter $\epsilon'_{\mathbf{d}}$ will result in lesser noise in the force data. And, it is recommended to use a higher value for $\epsilon'_{\mathbf{F}}$ to avoid the wasteful expenditure of precious sub-iterations. Use of stricter $\epsilon'_{\mathbf{F}}$ is also dangerous because there is a higher probability for the columns to be identical in $\mathbf{V}^i$ matrix of IQN-ILS framework, causing them to be eliminated frequently by the QR filter, resulting in non convergence of IQN-ILS framework. Therefore, the optimal coupling tolerances for IQN-ILS framework are $\epsilon'_{\mathbf{d}} = 10^{-6}$, $\epsilon'_{\mathbf{F}} = 10^{-3}$.

4.4. IQN-ILS framework settings

In [65], many parameters that influence the performance of IQN-ILS algorithm were specified. It was concluded in previous section that the QR filtering, which is one of the mentioned parameters in the above citation, can play a role in influencing the rate of convergence of sub-iterations, thereby dictating the level of noise in the solution. In addition to QR-filter the other parameters that can play a role in noise generation are maximum allowable columns in **V,W** matrices and maximum number of time steps allowed for construction of **V,W** matrices. Therefore, the effect of the above three IQN-ILS settings on noise in the force will be analyzed in this section.



(a) Mesh 2+2: Drag series

(b) Mesh 2+2: Lift seeries

(c) Mesh 2+2: Displacement Residual

(d) Mesh 2+2: Force Residual

Figure 4.3: Cylinder with Trailing Flap — FSI3 benchmark force and residual comparison of Mesh 2+2 configuration with different coupling tolerances. —/◦ : $\epsilon'_{\mathbf{d}} = 10^{-6}, \epsilon'_{\mathbf{F}} = 10^{-3}$; – –/◦ : $\epsilon'_{\mathbf{d}} = 10^{-6}, \epsilon'_{\mathbf{F}} = 10^{-3}$

QR filter	$\bar{k}$	Deleted columns	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
QR2/10^{-2}	6.84	3.5047	8.859×10^{-8}	4.137×10^{-4}	180.606	13892.357
QR1/10^{-6}	5.8	2.9151	2.002×10^{-7}	4.536×10^{-4}	102.681	5353.251

Table 4.4: Cylinder with Trailing Flap — Mean Number of sub-iterations, deleted columns, residuals per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2+2 configuration with QR-1 and QR-2 filtering for time interval 15 - 20 [s]

4.4.1. Type of QR-filter

As of this writing `preCICE` supports two types of QR filter — QR1 and QR2. The reason for employing QR filter in IQN-ILS framework is discussed in section 2.5.3. As described earlier, QR1 filter is based on eliminating columns in **V,W** matrices that causes singularity in upper triangular matrix (R matrix) from the QR decomposition of **V** matrix. QR2 filter is a novel QR filtering technique developed by Haelterman *et al.* in [38]. [65] recommends using QR-2 filter with limit of 10^{-2} or QR-1 filter with limit of 10^{-6}. Thus FSI3 simulations using the above two filters shall be compared. Typically, QR-filtering is expected to affect only the number of sub-iterations required for convergence. It was shown in section 4.3.2 that non converged sub-iterations can introduce numerical effects. Therefore to do a fair comparison, the coupling tolerances $\epsilon_{rel,disp} = 10^{-6}, \epsilon_{rel,force} = 10^{-3}$ were chosen in order to ensure that all the sub-iterations in the time window of comparison have converged. The results of this comparison are tabulated in table 4.4 for $t = [15, 20]s$. It can be seen that fewer sub-iterations are required for convergence in the simulation with QR1-filtering. The lower number of sub-iterations for QR-1 filter can be attributed to lesser number of columns being deleted in **V,W** matrices. Also, less noise is observed for the simulation with QR-1 filter. This is despite $\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$ and $\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$ being similar for both the simulations. It means that the partitioning errors for both the simulations are similar. Therefore, the additional noise generated by QR-2 filter cannot be from $\mathbf{d}^{\ddagger}_{m,\Gamma,iterative}$. Thus, this additional noise from QR-2 filter can be classified under $\mathbf{d}^{\ddagger}_{m,\Gamma,misc}$. In view of the above results, it was decided to employ QR-1 filter for rest of the analyses in this chapter.

4.4.2. Maximum number of columns in **V,W** matrices

In section 2.5.3, a hard limit was prescribed for number of previous sub-iterations, with which the **V,W** matrices are constructed, i.e. the maximum number of columns in the above mentioned matrices should be lesser than the number of degrees of freedom on fluid structure interface. The Mesh 2+2 configuration has 580 degrees of freedom on the structure interface. Thus, the number of sub-iterations reused for IQN-ILS temporal coupling in previous analyses — 100, is sufficiently below the hard limit of 580. It was decided to check if the maximum limit on the number of columns in **V,W** matrices has an influence on the noise in force data. FSI3 benchmark was repeated for Mesh 2+2 configuration, where the maximum number of columns in **V,W** matrices is restricted to 50. The results of this comparison are tabulated in table 4.5. There is only marginal difference in the amount of noise observed for lift and drag data. Also, the average number of sub-iterations and averaged residuals are marginally different. This marginal difference can be attributed to the maximum number of previous time steps allowed for construction of interface Jacobian. Average number of sub-iterations per time step is 5.8 for both the cases. Thus, with 10 time steps being reused, the number of columns in **V,W** matrices due to reuse from previous time steps is on average 58. For the test with 100 permitted columns, the **V,W** matrices will have 64 columns(58 + 6). Whereas for the test with 50 permitted columns, only the columns associated with the 50 latest sub-iterations out of 64 previous sub-iterations shall be recorded in the **V,W** matrices. If the **V,W** matrices are different, then the computed extrapolation coefficients will be different, resulting in the computed forces for the two cases being different. However, as seen in the results, this phenomenon only causes a marginal change to the existing noise in lift and drag data. Therefore, it can be concluded that the parameter governing the maximum number of permitted columns in **V,W** matrices has a marginal influence on the amount of noise in force data. Thus this parameter can have any value, as long as it is lower than the hard limit discussed earlier.

4.4.3. Maximum number of time steps reused in **V,W** matrices

In section 2.5.4, it was discussed that the number of sub-iterations required to attain convergence can be reduced by reusing data from previous time steps in the construction of **V,W** matrices. As

Max columns	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
100	5.8	2.002×10^{-7}	4.536×10^{-4}	102.681	5353.251
50	5.819	1.981×10^{-7}	4.527×10^{-4}	89.681	5656.754

Table 4.5: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2+2 configuration with 100 and 50 max number of columns in **V,W** matrices for time interval 15 - 20 [s]

Max time steps/columns	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
5/25	6.1	1.629×10^{-7}	4.528×10^{-4}	88.931	4568.524
10/50*	5.807	1.954×10^{-7}	4.519×10^{-4}	85.117	5308.252
20/100	5.81	2.225×10^{-7}	4.678×10^{-4}	173.206	9896.387
30/150	5.991	2.656×10^{-7}	4.768×10^{-4}	273.477	15264.047

Table 4.6: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2+2 configuration with 5, 10, 20 and 30 max number of reused time steps in **V,W** matrices for time interval 15 - 20 [s]

shown in (2.68), the oldest and newest vectors are on the rightmost and leftmost sides of the **V,W** matrices respectively. A limit can be specified for the maximum number of time steps employed for the construction of **V,W** matrices. If this limit is large, then it might be the case that the extrapolation in IQN-ILS is more biased towards past flow field, resulting in noise creation in force data. All the FSI3 tests discussed so far had data reused from at most 10 time steps for the construction of **V,W** matrices. It was not certain if this is the optimal value for the maximum number of time steps reused. Therefore, FSI3 benchmark was repeated for Mesh 2+2 configuration with maximum number of reused time steps — 5, 20 and 30. If data has to be reused from higher number of previous time steps, then the maximum number of permitted columns discussed in section 4.4.2 should be scaled proportionately as well. The results for the test involved in this comparison are tabulated in table 4.6. Data from 25, 50, 100 and 150 latest sub-iterations out of 36, 64, 122 and 186 averaged previous sub-iterations was reused for the simulation involving reuse of data from 10, 20 and 30 previous time steps respectively. This shows that a significant proportion of the available data from time step reuse criterion has been employed for all the cases, thus ensuring a fair comparison.

From the results, it can be seen that the number of time steps reused has a significant influence on the noise in force data when more than 10 time steps are reused. The noise magnitude in the signal seems to be directly proportional to the number of time steps reused (above 10) for the construction of **V,W** matrices. The average number of sub-iterations per time step appear to be similar for the cases involving 10 and 20 reused time steps. Whereas, it is higher for the case involving 5 and 30 reused time steps. Thus, for this particular FSI problem, higher number of reused time steps causes the vector extrapolation within IQN-ILS framework to be more biased to older solution, which might be the reason for larger noise. Thus, 10 seems to be the optimal value for number of time steps reused in the construction of **V,W** matrices, which shall be employed for further analyses in the remainder of this report.

4.4.4. Synthesis

Each IQN-ILS parameter investigated in this section, exhibited different levels of noise in force for different values of those parameters. However, the additional noise generated by tweaking with these parameters did not seem to affect the partitioning error. Therefore, the generated noise that was observed for these investigations can be classified under $\mathbf{d}^{\ddagger}_{m,\Gamma,misc}$.

*Data different from the second row in table 4.5 owing to the flow solver tolerance being 10^{-10}. Results in table 4.5 were estimated with the flow solver tolerance being 10^{-8}.

Extrapolation Order	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
constant	5.8	2.002×10^{-7}	4.536×10^{-4}	102.681	5353.251
2nd order	4.535	1.121×10^{-7}	4.428×10^{-4}	291.918	19100.721

Table 4.7: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2+2 configuration with and without displacement extrapolation for time interval 15 - 20 [s]

4.5. Effect of displacement extrapolation

FSI simulations that were discussed so far employed constant extrapolation from previous time step in order to obtain the initial interface displacement for the first sub-iteration of next time step, i.e. $\mathbf{d}^0_{m,\Gamma} = \mathbf{d}^{k_{final}}_{m-1,\Gamma}$. Initial value extrapolation can be employed to improve the prediction of initial interface displacement for any time step[66]. The first order extrapolation is defined as:

$$\mathbf{d}^0_{m,\Gamma} = 2\mathbf{d}^{k_{final}}_{m-1,\Gamma} - \mathbf{d}^{k_{final}}_{m-2,\Gamma} \tag{4.19}$$

whereas second order extrapolation is defined as:

$$\mathbf{d}^0_{m,\Gamma} = 2.5\mathbf{d}^{k_{final}}_{m-1,\Gamma} - 2\mathbf{d}^{k_{final}}_{m-2,\Gamma} + 0.5\mathbf{d}^{k_{final}}_{m-3,\Gamma} \tag{4.20}$$

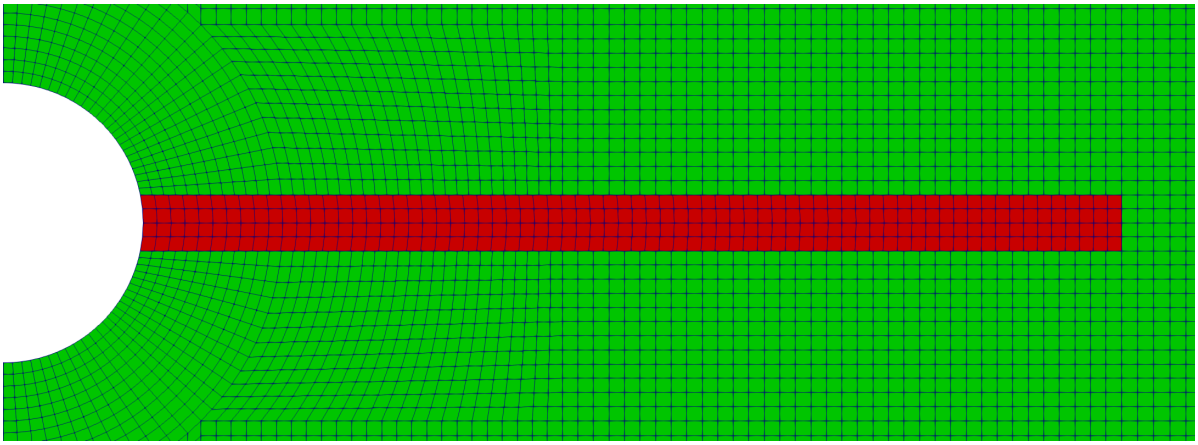
Since a better prediction for initial interface displacement can be made using extrapolation, it can result in fewer sub-iterations for each time step. And, by requiring fewer sub-iterations, the coupling residuals has more chances to converge for a fixed maximum number of sub-iterations. Thus, FSI3 simulation for Mesh 2+2 configuration was performed with and without displacement extrapolation in order to investigate the effectiveness of initial value extrapolation. Second order displacement extrapolation was employed for the simulation with initial value extrapolation. The results of this analysis for $t = [15, 20]s$ are shown in table 4.7. The employment of displacement extrapolation has indeed reduced the required number of sub-iterations. However, the noise seems to be amplified when displacement extrapolation is used, in spite of the final coupling tolerances being similar for both the simulations. This can be due to the IQN-ILS framework reusing information from multiple previous time steps (10 time steps in this case) to build columns for **V,W** matrices. Due to displacement extrapolation, the computed column corresponding to the initial sub-iteration of a time step might be incoherent in comparison to the columns from previous time steps. In spite of reduction in required number of sub-iterations, displacement extrapolation will not be employed for further studies in this section owing to noise amplification, which is in odds with goals of this research.

4.6. Effect of relative number of interface degrees of freedom

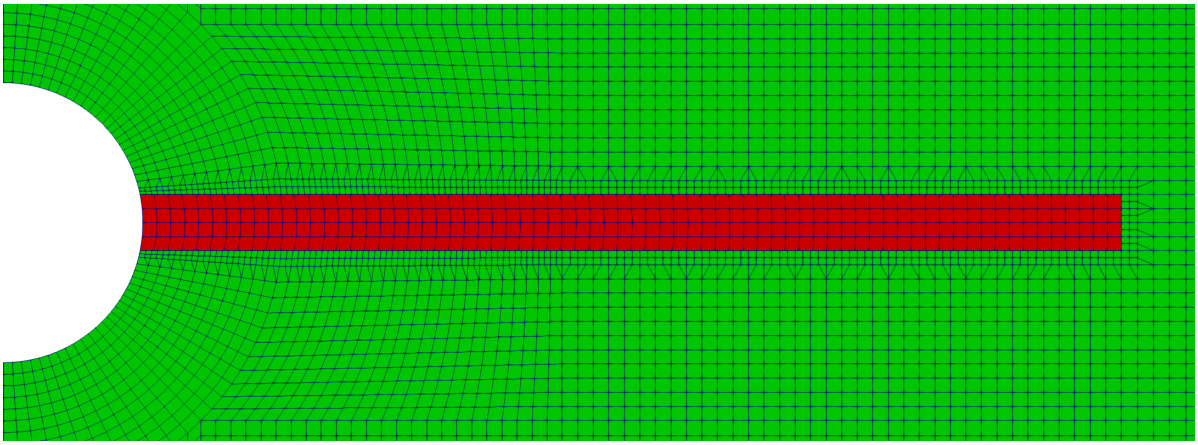
In [67], it was recommended to execute Quasi Newton acceleration on coarser grid for a robust coupling. In this research, the root finding problem was defined for interface displacement. Therefore, the **V,W** matrices in IQN-ILS framework were constructed using interface displacements. This means that for a robust coupling, the number of degrees of freedom on the structure interface should be lesser than the number of degrees of freedom on the fluid interface. Therefore, in this section the effect of relative number of interface degrees of freedom on noise in force data shall be evaluated.

4.6.1. Interface degrees of freedom

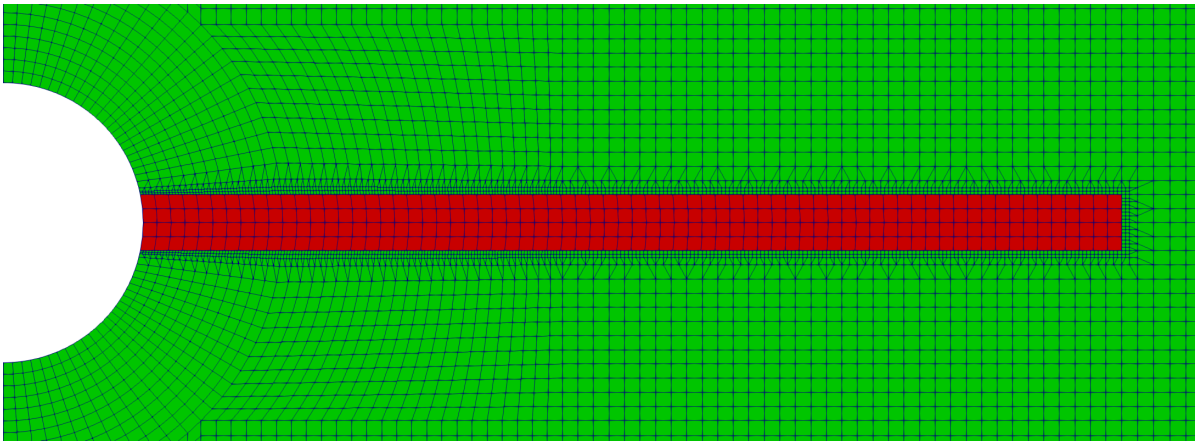
For this analysis, it is important to know what constitutes an interface degree of freedom. The face centres on the fluid interface correspond to locations of the interface degrees of freedom on the fluid mesh, whereas the nodes on the solid interface correspond to locations of the interface degrees of freedom on the solid mesh with respect to force coupling. Also, the face vertices on the fluid interface correspond to locations of the degrees of freedom on the fluid mesh, whereas the nodes on the solid interface again correspond to locations of the interface degrees of freedom on the solid mesh with respect to displacement coupling. For two dimensional analysis, each location mentioned above will have two dimensions of force and displacement for force and displacement coupling respectively. Since the number of face centres is lower than number of face vertices, number of face centres on the fluid



(a) Mesh 2+2



(b) Mesh 2ref1+2



(c) Mesh 2ref2+2

Figure 4.4: Cylinder with Trailing Flap — Illustration of Mesh 2+2 configurations with different levels of local refinement near flap interface

interface will serve as the critical parameter for evaluating the robustness of IQN-ILS algorithm. Keeping the above conventions in mind for interface degrees of freedom; the Mesh 2 configuration of the solid mesh has 580 degrees of freedom on its interface, and the Mesh 2 configuration of the fluid mesh has 296 degrees of freedom on its interface.

4.6.2. Local refinement in fluid Mesh 2

For a robust sequential IQN-ILS coupling involving the construction of **V,W** matrices using interface displacement vector, the number of degrees of freedom on the fluid interface is expected to be more than the number of degrees of freedom on the solid interface. It is not the case for the fluid Mesh2 in conjunction with solid Mesh2. Perhaps, the noise observed in FSI3 simulation for Mesh 2+2 configuration can be due to the IQN-ILS coupling not being robust. Thus, two additional fluid meshes — Mesh 2ref1 and Mesh2ref2 were created using local refinement for the cells on the interface of fluid Mesh2. For Mesh 2ref1, each interface cell on the original configuration was split into two; whereas for Mesh 2ref2, each interface cell on the original configuration was split into four. Therefore, after local refinement, Mesh 2ref1 and Mesh2ref2 have 592 and 1184 interface degrees of freedom respectively. Mesh 2ref1+2 has similar number of degrees of freedom on fluid and solid interface, whereas Mesh 2ref2+2 will have more than twice the number of degrees of freedom on the fluid interface in comparison to the solid interface. The three mesh configurations in this analysis — Mesh 2+2, Mesh 2ref1+2, Mesh2ref2+2 are illustrated in figure 4.4.

4.6.3. Numerical results

Since the fluid meshes involved in the configurations — Mesh 2+2, Mesh 2ref1+2, Mesh 2ref2+2 have different spatial discretization near the interface, it is prudent to present the displacement and force data for the above three configurations. This data is tabulated in table 4.8. Lift and Drag amplitude for the mesh configuration without local refinement (Mesh 2+2) is lower in comparison to the mesh configurations with local refinement (Mesh 2ref1+2, Mesh2ref2+2). The other quantities appear to be similar for all the mesh configurations. Since the spatial discretization in the base Mesh 2 configuration is not fine to begin with, the observed quantities for all the configurations deviate from the reference literature data. The drag and lift plot for the final second of the simulation are plotted for all the mesh configurations in figure 4.5. From the plots, it can be seen that the three signals are not in phase. The phase shift is due to the minor differences in frequency among the signals accumulating over a large duration. A strange phenomenon was observed for the lift plot of Mesh 2 + 2 configuration (see highlighted portion in the lift plot), where it exhibits kinks on the advancing side of extreme values in the signal; which is not the case for Mesh 2ref1+2 and Mesh 2ref2+2 configurations. This could be the reason for the lift amplitude being small for Mesh 2+2 configuration.

The total variation of the lift and drag force as well as averaged final residuals for the final 5s of the simulation are tabulated in table 4.9. It can be seen that the Mesh 2ref2+2 configuration has least noise for drag and lift data. This result is in line with the recommendation in [67]. However, marginally higher noise is observed in the lift data for Mesh 2ref1+2 configuration in comparison to the Mesh 2+2 configuration, which is in contrast to the recommendation. A clear reason for the lowest observed noise in Mesh 2ref2+2 configuration could not be explained. However, the mesh 2ref2 + 2 configuration requires more sub-iterations for convergence, resulting in longer computation times. Since, the observed noise in the force data is low for Mesh 2ref2+2 configuration, it was decided to perform further analyses using this configuration.

4.6.4. Synthesis

Least noise was observed in force data for Mesh 2ref2 + 2 configuration. However, the amount of partitioning error appears to be the same for all the mesh configurations. Therefore, the additional noise in force observed for the Mesh 2+2 and Mesh2ref1 + 2 configuration can be classified under $\mathbf{d}^{\ddagger}_{m,\Gamma,misc}$.

Mesh	**Axis**	**u**			**F**		
		Mean[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]	**Mean**[N]	**Amp.**[N]	**Freq.**[s^{-1}]
2 + 2	X	-2.998	2.754	11.13	481.54	24.243	11.39
	Y	1.530	35.923	5.49	1.540	183.521	5.49
2ref1 + 2	X	-3.113	2.855	11.07	481.85	27.588	11.30
	Y	1.522	36.527	5.47	1.902	189.020	5.47
2ref2 + 2	X	-3.100	2.841	11.06	482.00	27.991	11.26
	Y	1.527	36.449	5.46	1.631	189.108	5.46
Ref.	X	-2.69	2.53	10.9	457.3	22.66	10.9
	Y	1.48	34.38	5.3	2.22	149.78	5.3

Table 4.8: Cylinder with Trailing Flap — FSI3 benchmark results for Mesh 2+2, Mesh 2ref1+2, Mesh2ref2+2 configurations

Mesh	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
2 + 2	5.807	1.954×10^{-7}	4.519×10^{-4}	85.117	5308.252
2ref1 + 2	5.714	3.347×10^{-7}	4.821×10^{-4}	84.977	6359.040
2ref2 + 2	6.148	4.692×10^{-7}	4.592×10^{-4}	54.790	3149.060

Table 4.9: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2+2, Mesh 2ref1+2 and Mesh2ref2+2 configurations for time interval 15 - 20 [s]

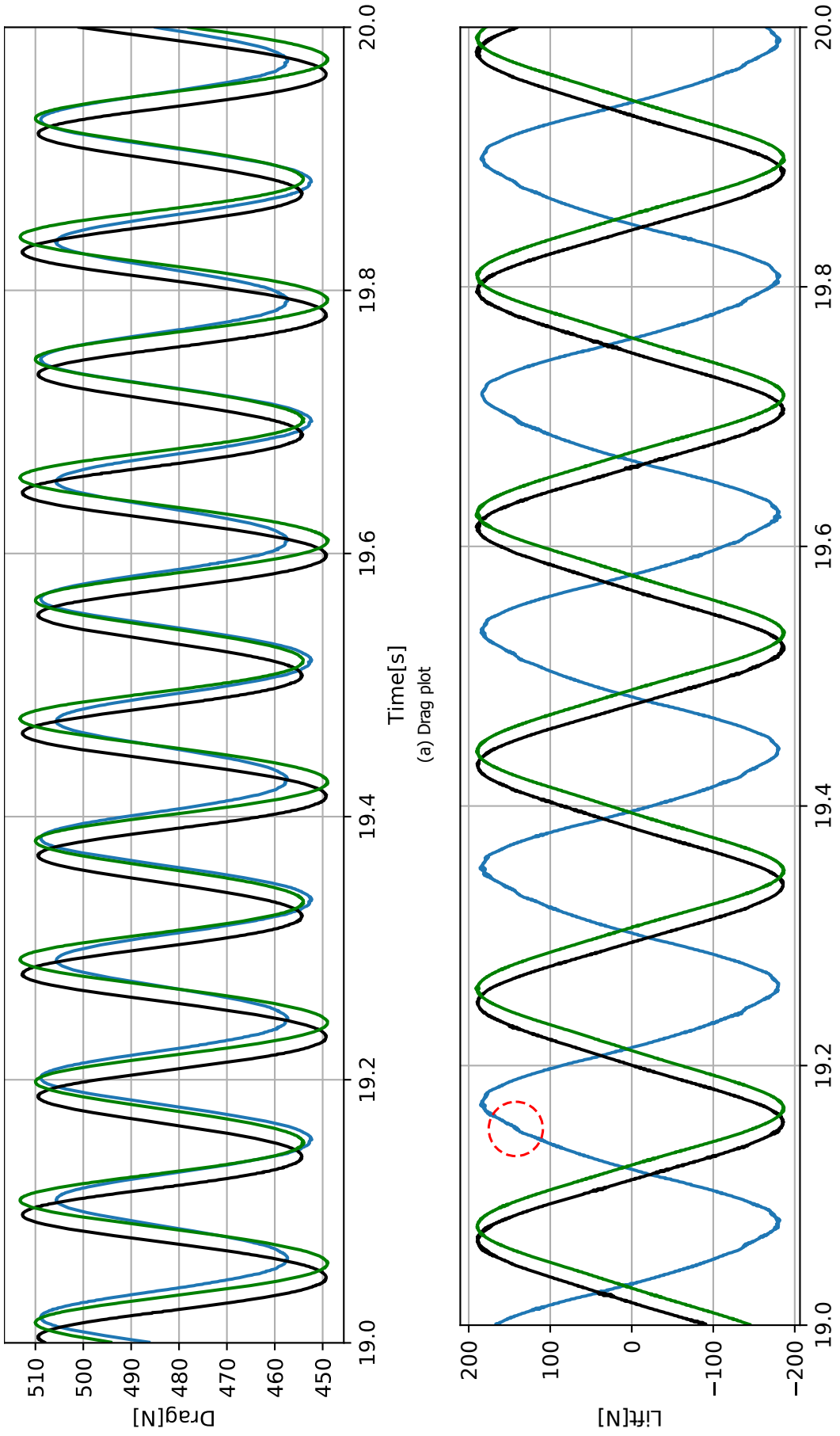


Figure 4.5: Cylinder with Trailing Flap — Drag and Lift plot for FSI3 benchmark of Mesh 2+2, Mesh 2ref1+2 and Mesh2ref2+2 configurations. —: Mesh 2+2; —: Mesh 2ref1+2; —: Mesh 2ref2+2

4.7. Comparison of IQN-ILS with Aitken's and Fixed point frameworks

Although, the IQN-ILS method falls under the family of Newton Raphson Iterative frameworks, it can also be interpreted as fixed point iterative method. The basic fixed point iteration method is equivalent to newton raphson iteration if:

$$\mathbf{J}^{-1}_{k,\Gamma,fixed} = -\mathbf{I} \tag{4.21}$$

For fixed point iteration with constant/dynamic(Aitken's) under-relaxation, the equivalency condition is:

$$\mathbf{J}^{-1}_{k,\Gamma,under-relaxed} = -\omega^k \mathbf{I} \tag{4.22}$$

where $\omega^k = \omega^{k-1}$ for constant under-relaxation and $\omega^k = -\omega_{k-1}\frac{\mathbf{r}_\Gamma^{k-1}\cdot(\mathbf{r}_\Gamma^k - \mathbf{r}_\Gamma^{k-1})}{\|\mathbf{r}_\Gamma^k - \mathbf{r}_\Gamma^{k-1}\|^2}$ for Aitken's under-relaxation. Whereas for IQN-ILS framework, the inverse of the interface Jacobian is approximated as [38]:

$$\begin{aligned}\mathbf{J}^{-1}_{k,\Gamma,IQN} &= \mathbf{W}^k(\mathbf{V}^{k^T}\mathbf{V}^k)^{-1}\mathbf{V}^{k^T} - \mathbf{I} \\ \mathbf{J}^{-1}_{k,\Gamma,IQN} &= \mathbf{W}^k(\mathbf{V}^{k^T}\mathbf{V}^k)^{-1}\mathbf{V}^{k^T} + \mathbf{J}^{-1}_{k,\Gamma,fixed}\end{aligned} \tag{4.23}$$

By comparing (4.21) and (4.23), it can be seen that the inverse Jacobian for IQN-ILS framework is a sum of the inverse Jacobian for fixed point iteration and $\mathbf{W}^k(\mathbf{V}^{k^T}\mathbf{V}^k)^{-1}\mathbf{V}^{k^T}$. Therefore the components of interface residual that lie in the column space of $\mathbf{W}^k(\mathbf{V}^{k^T}\mathbf{V}^k)^{-1}\mathbf{V}^{k^T}$ experience additional rate of convergence. And, for components that lie outside the column space, IQN-ILS method is equivalent to fixed point iteration. If the **V,W** matrices become a square matrix after N sub-iterations, then all the components of interface residual will lie in the column space of $\mathbf{W}^k(\mathbf{V}^{k^T}\mathbf{V}^k)^{-1}\mathbf{V}^{k^T}$, thereby resulting in increased rate of convergence for all components. Since the above three mentioned frameworks are a form of Newton-Raphson Iteration in one way or another, it is interesting to see if Aitken's and Fixed under-relaxation frameworks can generate noise in force or not. To this end, FSI3 benchmark for Mesh 2ref2+2 configuration is repeated with fixed under-relaxation and Aitken's under-relaxation. As mentioned earlier, since fixed and Aitken's under-relaxation requires large number of sub-iterations for convergence, the above two simulations shall be performed for a small time window.

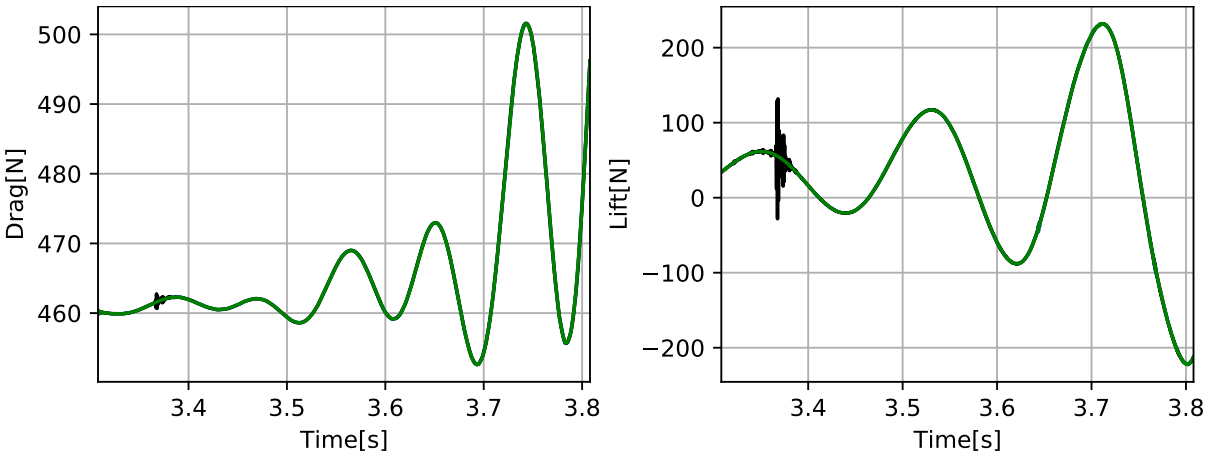
4.7.1. Numerical results

All the tests involved in this comparison are simulated for the time interval 3.308s – 3.808s. Mesh 2ref2+2 configuration was employed for this analysis. Since earliest instance of noise is observed in this time interval for IQN-ILS framework, it was chosen for comparing IQN-ILS framework with Aitken's and fixed under-relaxation framework. For all the cases, simulation was restarted from flow and structure field at 3.308s. Theses field were obtained using IQN-ILS framework until the above mentioned time. In case of fixed under-relaxation framework, 0.05 was chosen as the relaxation factor, since it was the highest possible value for which the simulation does not fail for the above mentioned time interval. To maintain similarity in comparison, the initial relaxation value was also chosen as 0.05 for the IQN-ILS and Aitken's under-relaxation frameworks.

The results of the noise analysis are enclosed in table 4.10. As expected IQN-ILS framework requires least number of sub-iterations for convergence, whereas fixed under-relaxation framework requires the highest number of sub-iterations. Also, the fixed under-relaxation framework has the least amount of noise. This implies that the displacement prediction routines within IQN-ILS and Aitken's under-relaxation play a role in generating noise. This noise especially seems to be more pronounced for the Aitken's under-relaxation framework. The averaged force residual for fixed under-relaxation framework is of one order of magnitude lower in comparison to other frameworks. This could be the consequence of constant under-relaxation within the sub-iterations of fixed under-relaxation framework, where the variation in force is performed monotonously over large number of sub-iterations.

Coupling Framework	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
IQN-ILS	5.148	5.259×10^{-7}	3.811×10^{-4}	3.336	116.672
Aitken's under-relaxation	31.338	5.850×10^{-7}	3.293×10^{-4}	17.019	985.477
Fixed under-relaxation	134.153	9.750×10^{-7}	5.274×10^{-5}	0.915	21.695

Table 4.10: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2ref2+2 configuration with IQN-ILS, Aitken's and Fixed under-relaxation temporal coupling frameworks for time interval 3.308 - 3.808 [s]



(a) Drag Time Series (b) Lift Time Series

Figure 4.6: Cylinder with Trailing Flap — Force comparison for Mesh 2ref+2 configurations with different temporal coupling frameworks. — : IQN-ILS; — : IQN-ILS-Optimized; — : Fixed under-relaxation

4.7.2. Synthesis

Of all the three frameworks compared in this analysis, fixed point under-relaxation with constant relaxation factor, had the least noise. In fact, it can be assumed to be almost noise free. The partitioning error for all the three frameworks are of similar magnitude. Therefore, the additional noise observed for IQN-ILS and Aitken's under-relaxation can be classified under $\mathbf{d}^{\ddagger}_{m,\Gamma,misc}$.

4.8. FSI3 simulation with optimal settings

Of all the analyses performed so far in this chapter using IQN-ILS temporal coupling framework for Mesh 2+2 configuration, the simulation on Mesh 2ref+2 configuration with QR-1 filter, displacement tolerance of 10^{-6} and force tolerance of 10^{-3} exhibited least noise. In section 4.3, it was shown that the noise starts diminishing with stricter displacement tolerance. Thus, simulations were tried with displacement tolerance below 10^{-6}. However, it was observed that once the displacement residual is of the order of 10^{-6}, it stagnates, causing the sub-iterations to not converge, irrespective of maximum permitted sub-iterations in a time step. Displacement residual stagnating around 10^{-6} might be due to the filtering tolerance (10^{-6}) of QR-1 filter discussed in section 2.5.3 being of similar order of magnitude; causing the columns in **V,W** matrices, that can predict more accurate interface displacement to be eliminated.

Another reason for displacement residual stagnating could be that the components in interface residual that lie outside the column space of interface Jacobian matrix are the only significant components remaining, which are not treated by the IQN-ILS framework. Perhaps, the remaining components in the interface residual can be treated using fixed point iteration with constant under-relaxation. This is referred to as Anderson temporal coupling framework, which is not available in `preCICE` as of this writing.

In view of the above, FSI3 simulation was attempted with a QR-1 filtering tolerance of 10^{-8} with displacement tolerance of 10^{-7} and force tolerance of 10^{-3} to investigate the former theory. IQN-ILS simulation with these settings shall be referred to as optimized IQN-ILS. When the simulation was

Coupling Framework	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
IQN-ILS	5.329	5.279×10^{-7}	3.928×10^{-4}	2.173	110.170
IQN-ILS optimized	10.449	7.485×10^{-8}	1.345×10^{-4}	0.766	44.936
Fixed under-relaxation	136.418	9.745×10^{-7}	5.381×10^{-5}	0.073	19.375

Table 4.11: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2ref+2 configuration with IQN-ILS, Optimized IQN-ILS and Fixed under-relaxation temporal coupling frameworks for time interval 3.389 - 3.808 [s]

Coupling Framework	$\bar{k}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{d}}\|}$	$\overline{\|\mathbf{r}_{\Gamma,\mathbf{F}}\|}$	$\Theta(\Delta\mathbf{F}_{x,\Gamma})$	$\Theta(\Delta\mathbf{F}_{\mathbf{y},\Gamma})$
IQN-ILS	6.21	4.685×10^{-7}	4.768×10^{-4}	16.085	962.329
IQN-ILS optimized	7.86	6.097×10^{-8}	2.633×10^{-4}	10.288	362.101

Table 4.12: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2ref+2 configuration with IQN-ILS and Optimized IQN-ILS for time interval 13 - 14.5 [s]

started from rest, the optimized IQN-ILS simulation crashed. This can be due to the lenient QR-1 filtering tolerance causing singularity in the **V,W** matrices for the ramp up phase of the boundary conditions (0-2s), when sub-iterations are not expected to converge. When the optimized IQN-ILS simulation was restarted from flow and structure fields at 3.308 s, it did not crash. Force data for the initial 1000 time steps from optimized IQN-ILS simulation, is compared with the IQN-ILS and fixed under-relaxation results from section 4.7.1 in figure 4.6. Within first 150 time steps, significant noise was observed for the optimized IQN-ILS simulation. This was due to the sub-iterations not converging for those instances. From 3.389s onwards, the sub-iterations of optimized IQN-ILS simulation have started converging. Thus, in the next section the numerical analysis of IQN-ILS framework with optimized settings shall be performed for results from 3.389s.

4.8.1. Numerical results

The results of optimized IQN-ILS simulation in comparison with IQN-ILS and fixed point under-relaxation frameworks for interval 3.389 - 3.808 [s] are tabulated in table 4.11. On average, simulation with optimized IQN-ILS settings requires twice the number of sub-iterations in comparison to the simulation with standard IQN-ILS settings for convergence. And, the forces obtained using optimized IQN-ILS settings exhibit less noise in comparison to standard IQN-ILS settings. Still, the noise observed for optimized IQN-ILS settings is high in comparison to noise for the simulation with fixed under-relaxation framework. This is in spite the averaged displacement residual for optimized IQN-ILS simulation being an order of magnitude lower than that of simulation with fixed point under-relaxation. This finding is a strong evidence for one of the conclusions made in section 4.7 that the least sq.

Due to higher number of sub-iterations required for convergence, the simulation with optimized IQN-ILS settings managed to run till 14.5755 s before running out of allotted computation time. Thus comparison of simulation with optimized and standard IQN-ILS settings shall be made for the time interval 13 - 14.5 [s] in order to analyse noise in the periodic region of the force data. The results for this comparison are tabulated in table 4.12. As expected, the simulation with optimized IQN-ILS settings has less noise in comparison to the one with standard IQN-ILS settings. Also, for this time interval, the average number of sub-iterations required for convergence is only 27% larger for optimized IQN-ILS in comparison to standard IQN-ILS. Therefore, to obtain a force data with less noise, it is recommended to use the optimized settings for IQN-ILS simulation, provided that the initial flow and structure fields are from the periodic region of the fluid-structure interaction. These initial fields can be obtained using standard IQN-ILS settings. Do note that the optimized iQN-ILS settings are applicable only for the FSI3 benchmark of the cylinder with trailing flap test case using the Mesh 2ref2+2 configuration.

4.9. Conclusions

The influence of various temporal coupling settings on the noise in force data has been documented in this chapter. For relativistic comparison of noise levels, a mathematical tool called total variation method was employed to determine the smoothness function defined in section 4.1. Signal with lower smoothness function is deemed to have lesser noise. A hypothesis was proposed in section 4.2, wherein the error in interface displacement, if any, will amplify by three orders of magnitude for $\Delta t = 0.0005s$ on differentiation. Since, force acting on the flap is a function of first derivative of interface displacement, minuscule error in interface displacement will amplify into significant error in force, which manifests as noise. The findings in section 4.3 support the above mentioned hypothesis indirectly by showing that smaller interface residual leads to smaller iterative error, resulting in lower noise in force data. It is beneficial to have stricter displacement tolerance for lesser noise, subject to the sub-iterations converging to the said tolerance. Several other settings of the temporal coupling framework were investigated to check their influence on noise in sections 4.4 - 4.8. From these analyses, it was concluded that IQN-ILS(10) framework on Mesh 2ref2 + 2 configuration with QR-1 filtering tolerance of 10^{-8} exhibited the least noise in force data. But, the optimized IQN-ILS framework still generated more noise in force data in comparison to fixed point iteration with constant under-relaxation. This marks the end of the first half of the revised objective for this research, **To investigate sources of noise in force data from FSI3 benchmark**. However, the number of sub-iterations required for fixed point framework are couple of orders of magnitude larger than IQN-ILS framework. Therefore, it is beneficial to find ways to reduce or remove the error in interface displacement before it is being coupled with the flow solver, which is nothing but the other half of this research objective. It shall be treated in the next chapter.

5

General Numerical Noise Filtering Framework

It was concluded in the previous chapter that the force data from FSI3 benchmark with the optimized IQN-ILS framework still possessed higher noise in comparison to that of fixed under-relaxation framework. Thus, this chapter will discuss about strategies to predict the occurrence of noise/artefacts in force data a priori before the force has been computed and implement a fix that can eliminate or dampen this noise using a mathematical framework.

5.1. Noise onset prediction in Force Data

It was shown in section 4.2 that the force acting on the flap is a function of the first derivative of interface displacement. Therefore, for predicting the onset of noise in force data, we can rely upon minuscule noise in displacement data. However, the interface displacement residual does not paint a complete picture of the error in interface displacement, since it is just an image of the error in interface displacement. Thus, displacement data of individual interface nodes are also required to predict the onset of noise in force data. For this study, force and displacement data obtained using IQN-ILS and fixed under-relaxation temporal coupling frameworks shall be analysed. For the IQN-ILS framework the coupling tolerances were $\epsilon_{rel,disp} = 10^{-4}, \epsilon_{rel,force} = 10^{-4}$. Also, displacement extrapolation was enabled for the analysis. These settings were chosen in order to trigger noise of larger magnitude, resulting in a clear distinction between data from IQN-ILS and fixed under-relaxation frameworks. These comparisons shall be done for the time interval – 3.308 - 3.808 [s].

5.1.1. Numerical acceleration

Drag and lift data has been plotted for the simulations with IQN-ILS and fixed under-relaxation frameworks in figures 5.1a and 5.1b respectively. As expected the force data for IQN-ILS framework has noise, with the drag having marginal noise and the lift having significant noise. This is also indicated by the smoothness function values for the forces from two frameworks in table 5.1. However, when observing the displacement data for the node located at (0.284,0.19)m, both appear to have a smooth profile. This is also indicated by the smoothness function values of the displacement data being zero for both IQN-ILS and fixed under-relaxation frameworks. The same phenomenon was observed for the

Coupling Framework	$\bar{k}$	$\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{d}} \rVert}$	$\overline{\lVert \mathbf{r}_{\Gamma,\mathbf{F}} \rVert}$	$\Theta(\mathbf{\Delta F}_{x,\Gamma})$	$\Theta(\mathbf{\Delta F}_{\mathbf{y},\Gamma})$
IQN-ILS	6.163	3.012×10^{-7}	5.864×10^{-5}	12.447	1166.77
Fixed under-relaxation	134.153	9.750×10^{-7}	5.274×10^{-5}	0.915	21.695

Table 5.1: Cylinder with Trailing Flap — Mean number of sub-iterations, coupling tolerances per time step; and Smoothness of Drag/Lift data for FSI3 simulations on Mesh 2ref+2 configuration with IQN-ILS and fixed under-relaxation framework for time interval 3.308 - 3.808 [s]

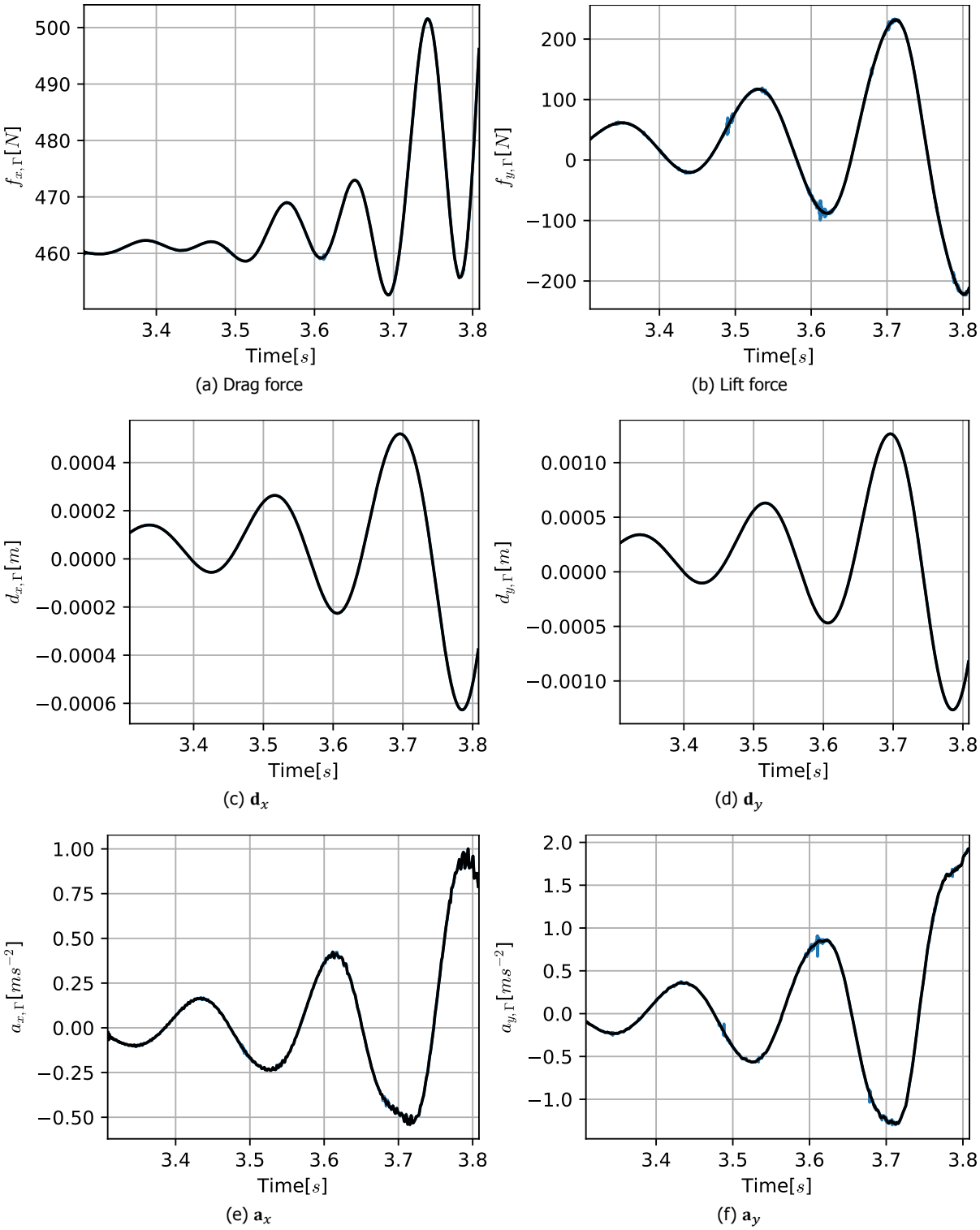


Figure 5.1: Cylinder with Trailing Flap — Force, displacement and acceleration comparison for Mesh 2ref+2 configurations with different temporal coupling frameworks. Displacement and Acceleration were computed for the node located at (0.284,0.19)m. —— : IQN-ILS; —— : Fixed under-relaxation

Coupling Framework	$\Theta(\mathbf{\Delta d}_{x,\Gamma})$	$\Theta(\mathbf{\Delta d}_{y,\Gamma})$	$\Theta(\mathbf{\Delta a}_{x,\Gamma})$	$\Theta(\mathbf{\Delta a}_{y,\Gamma})$
IQN-ILS	0	0	2.490	5.536
Fixed under-relaxation	0	0	1.587	0.467

Table 5.2: Cylinder with Trailing Flap — Smoothness of displacement and acceleration data for FSI3 simulations on Mesh 2ref+2 configuration with IQN-ILS and fixed under-relaxation framework for time interval 3.308 - 3.808 [s] at the node located at (0.284,0.19)m

remaining 289 nodes on the structure interface. Thus, it is not possible to make any prediction on the onset of noise using just the displacement at interface nodes on the structure owing to its minuscule magnitude. In section 4.2 it was mentioned that noise in displacement data can get magnified due to differentiation with respect to time when time step is smaller than 1s. Second derivative of interface displacement will make it easier to distinguish $\mathbf{d}^{\ddagger}_{m,\Gamma}$ (error) from $\mathbf{d}^{\ddagger}_{m,\Gamma}$ (error-free). Therefore, numerical acceleration obtained through second numerical differentiation of displacement data can be useful in predicting the onset of noise in force data.

Acceleration at the structure can be calculated numerically using finite differences of the interface displacement with respect to time. For the initial time step, acceleration is obtained using second order forward difference of displacement, i.e.:

$$a^{0}_{x\vee y,\Gamma} := \frac{d^{2}_{x\vee y,\Gamma} - 2d^{1}_{x\vee y,\Gamma} + d^{0}_{x\vee y,\Gamma}}{\Delta t^{2}} \tag{5.1}$$

Acceleration for intermediate time steps is obtained using the second order central difference of displacement, i.e.:

$$a^{m}_{x\vee y,\Gamma} := \frac{d^{m+1}_{x\vee y,\Gamma} - 2d^{m}_{x\vee y,\Gamma} + d^{m-1}_{x\vee y,\Gamma}}{\Delta t^{2}} \qquad \forall \;\; m \in \{1, 2, \dots, n-1\} \tag{5.2}$$

And, the acceleration for the final time step is obtained using second order backward difference of the displacement , i.e.:

$$a^{n}_{x\vee y,\Gamma} := \frac{d^{n}_{x\vee y,\Gamma} - 2d^{n-1}_{x\vee y,\Gamma} + d^{n-2}_{x\vee y,\Gamma}}{\Delta t^{2}} \tag{5.3}$$

The numerical acceleration for both the directions are plotted for the node located at (0.284,0.19)m in figures 5.1e and 5.1f. It can be seen that the numerical acceleration possess noise. The amount of noise present in numerical acceleration for both IQN-ILS and fixed point acceleration are tabulated in table 5.2. The amount of noise in interface acceleration is higher for the simulation using IQN-ILS framework. For horizontal acceleration, the noise for both the frameworks are marginally different; whereas for vertical acceleration, the noise is significantly different.

5.1.2. Correlation analysis between numerical acceleration and force data

In table 5.1 it was shown that the force exerted by the fluid on the flap is relatively noise free for the simulation using fixed under-relaxation framework in comparison to IQN-ILS framework. And, in previous section it was shown for a node on the flap interface that numerical acceleration using fixed under-relaxation has less noise in comparison to IQN-ILS framework. This was also observed for the remaining 289 nodes. From the above two statements, it can be inferred that either less noise in force exerted by the fluid domain is the reason for less noise in acceleration of the node on the structure interface or less noise in acceleration on a structure interface node is the reason for less noise in force exerted by the fluid domain. If the latter is true, then it will also be in agreement with the hypothesis made in section 4.2. To ascertain if the former or latter is true in the above inference it is beneficial to perform a correlation analysis between the force and numerical acceleration data. Before performing correlation analysis, couple of modifications were made to the force and acceleration data for simplicity. Instead of performing correlation analysis for horizontal and vertical components separately, the magnitude of force and acceleration shall be considered. And, the force and acceleration magnitudes obtained using fixed under-relaxation framework shall be considered as reference data, i.e. force and

Nodes	x[m]	y[m]	$max(P(a_\Gamma, F_\Gamma))$	delay value (q)
D	0.339	0.19	0.519	4
E	0.425	0.19	0.531	4
F	0.515	0.21	-0.326	4
A	0.6	0.20	-0.152	6
avg. all nodes	N/A	N/A	-0.447	4

Table 5.3: Cylinder with Trailing Flap — Maximum cross-correlation coefficient between relative acceleration and relative force on Mesh 2ref+2 configuration for time interval 3.308 - 3.808 [s]

acceleration magnitudes obtained using IQN-ILS framework are subtracted by the corresponding magnitudes from fixed point under-relaxation. The relative force and acceleration magnitudes obtained after the above two operations, shall be employed for correlation analysis.

Suppose $a_\Gamma^m := \sqrt{(a_{x,\Gamma}^m)^2 + (a_{y,\Gamma}^m)^2}_{IQN} - \sqrt{(a_{x,\Gamma}^m)^2 + (a_{y,\Gamma}^m)^2}_{ref}$ and $F_\Gamma^m := \sqrt{(F_{x,\Gamma}^m)^2 + (F_{y,\Gamma}^m)^2}_{IQN} - \sqrt{(F_{x,\Gamma}^m)^2 + (F_{y,\Gamma}^m)^2}_{ref}$, then the cross-correlation coefficient for a delay of q time steps is defined as:

$$P(a_\Gamma, F_\Gamma) := \frac{\sum_{m=0}^{n}(a_\Gamma^m - \overline{a}_\Gamma)(F_\Gamma^{m+q} - \overline{F}_\Gamma)}{\sqrt{\sum_{m=0}^{n}(a_\Gamma^m - \overline{a}_\Gamma)^2}\sqrt{\sum_{m=0}^{n}(F_\Gamma^{m+q} - \overline{F}_\Gamma)^2}} \tag{5.4}$$

The above coefficient can range from -1 to 1, with negative values implying negative correlation and positive values implying positive correlation. If P is -1 or +1 then there is a high correlation between the two time series. If $P = 0$ then there is no correlation between the two time series. Two identical signals if out of phase will have low correlation. In such cases, a non zero delay value can be employed to compensate the phase shift.

5.1.3. Numerical Results

Relative acceleration was numerically computed for four randomly selected nodes on the flap interface. The locations of these four nodes are tabulated in table 5.3. Also, the average value of relative acceleration for these interface nodes was computed. In figure 5.2, the relative accelerations as well as relative force are plotted in order to visually correlate the perturbations. The vertical lines clearly show that large noise spikes in relative acceleration occur at the same time in most of the nodes. This is also further proven by the presence of spikes at similar intervals in the average relative acceleration plot. And, whenever these large spikes occur in relative acceleration, noise spikes were also observed in the relative force data in the proximity of the vertical lines. This indicates there is some sort of relationship between the numerical acceleration and force data. Also in table 5.3, the maximum correlation coefficient and the delay in time steps are plotted. Positive correlation was observed for nodes D and E. Whereas negative correlation was observed for nodes F and A. And the strength of correlation seems to decay for nodes beyond the midpoint. Especially for node A, which is on the trailing edge of flap, there appears to be no correlation. A negative correlation coefficient is observed for the data averaged over all nodes. Since its value is relatively strong, we can conclude that the noise in interface acceleration on most of the nodes exhibit a relationship with the noise in force data. Also the maximum correlation coefficient for nodes D, E, F and the averaged data were observed when the relative force data is shifted backward in time by four time steps. This is in agreement with the noise spikes in force data appearing few time steps after the red lines in figure 5.2. Thus, it can be concluded that negligible noise in interface displacement data amplify in interface acceleration data, which when coupled to the flow solver can cause perturbations in the force data. Hence, the above finding also validates the hypothesis made in section 4.2.

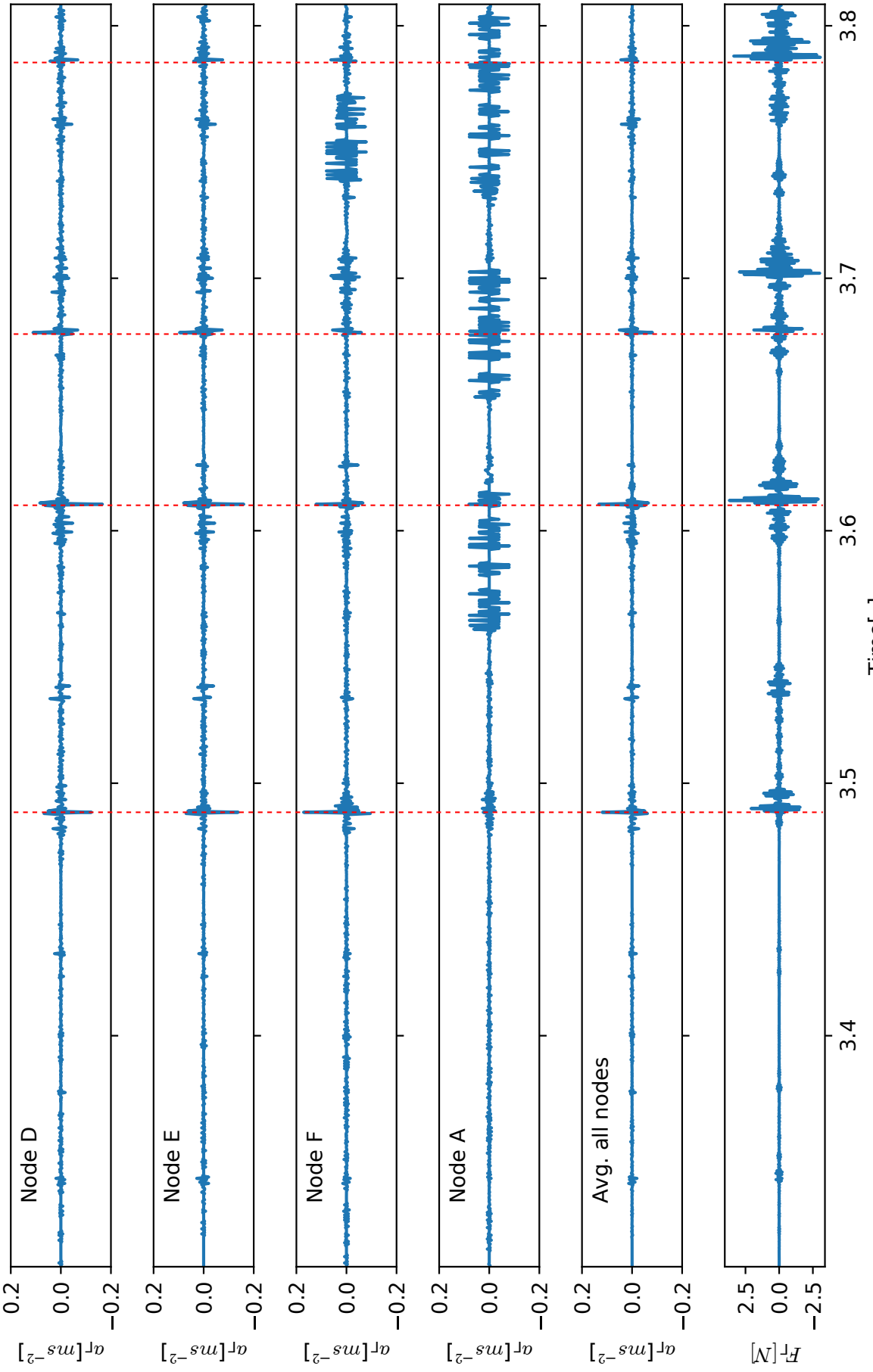


Figure 5.2: Cylinder with Trailing Flap — Relative acceleration for four different nodes, average relative acceleration and relative force on Mesh 2ref2+2 configuration for time interval 3.308 - 3.808 [s]

Prerequisites	Description
h	Number of previous time steps required by the smoothing operator for noise filtering
N	Total number of nodes on the flap interface
n	Total number of time steps
w	Relevant parameters for the smoothing operator

Table 5.4: General Noise Filter Framework — Description of filter prerequisites

5.2. General framework for noise filter in interface displacement

Based on the conclusion made in the previous section, it can be reasoned that by reducing the noise in the acceleration numerically computed on the nodes of the flap interface, the force data will exhibit less noise during a partitioned FSI simulation. And, noise in the numerical acceleration can be dampened by filtering out the noise in interface displacement. To this end, a general framework has been created for filtering out noise from interface displacement data. This framework can be integrated with any temporal coupling framework within `preCICE` through its python Callback interface (see [68] for more information). As of this writing, this integration is pending. Therefore, it is not possible to comment on its effectiveness in regards to dampening the noise in force data.

Algorithm 1 General Noise Filter Framework to smoothen displacement data at all interface nodes

Require: : h, N, n, t, w
  for $i \leftarrow 1$ **to** N **do**
    $d_x \leftarrow U_x[:, i]$
    $d_y \leftarrow U_y[:, i]$
    for $j \leftarrow h + 1$ **to** n **do**
      $d_x temp \leftarrow S(d_x[j - h : j], t[j - h : j], w)$
      $d_y temp \leftarrow S(d_y[j - h : j], t[j - h : j], w)$
      $d_x \leftarrow d_x temp[h + 1]$
      $d_y \leftarrow d_y temp[h + 1]$
    end for
    $U_x[:, i] \leftarrow d_x$
    $U_y[:, i] \leftarrow d_y$
  end for

The pseudocode for the general noise filter framework is portrayed in algorithm 1. The prerequisites for this framework are tabulated in table 5.4. This general framework has a smoothing operator S which filters out noise by replacing the existing displacement for current time step with extrapolated displacement from h previous time steps. Three types of smoothing operators were investigated to perform this extrapolation. They are Smoothing spline filter, Savitzky-Golay filter and Sinusoidal curve-fitting filter. Detailed information and their effectiveness in filtering out noise from interface displacement shall be discussed in the remaining sections of this chapter. As discussed in section 5.1.1, computed numerical acceleration shall be employed for judging the effectiveness of above mentioned filters in dampening noise. Integration of this framework with `preCICE` requires an addition inner loop in the algorithm so that the extrapolation process can be performed for each sub-iteration in IQN-ILS framework. This filtering can be done on interface displacement data for all nodes before they are coupled to the flow solver. If it works as intended, then another side benefit of this filtering framework is potential reduction in number of sub-iterations required for convergence of IQN-ILS framework. In previous chapter, it was concluded that IQN-ILS framework generates less noise in force data when the displacement tolerance is stricter. However, convergence of IQN-ILS framework is not a trivial task for stricter displacement tolerance due to the issue of some components of interface residual lying outside the column space of inverse of interface Jacobian. Therefore, the general noise filter framework proposed in this chapter aims to achieve less noise in force data while employing a relatively lenient displacement tolerance.

5.3. Smoothing spline filter

A spline function is a set of polynomial up to degree n, defined for all the time intervals encompassing a time series, that are piece-wise continuous up to an arbitrary order of continuity $C^o; o < n$[69]. Through various algorithms, this spline function can be employed to interpolate a discrete time series. Discrete time series can also be interpolated using polynomial interpolation. However, when polynomial interpolation is performed with higher degrees, the interpolated values exhibit oscillations near the boundaries of a time series. This is described as Runge's phenomenon[70]. To avoid the above mentioned issue, polynomial spline interpolation enables the utilization of lower degree piece-wise continuous polynomials to interpolate a discrete time series. The most common polynomial spline interpolation is the natural cubic spline which is a spline of the smallest polynomial degree that has continuity C^2. Continuity of at least C^2 is required for the spline to have least bending. Consider a discrete time series $\mathbf{g}$ with $r+1$ samples, whose spline function $s(t)$ is defined for $t \in [t_0, t_r]$. Let the cubic polynomial for r time intervals in the above mentioned spline function be defined as:

$$s_m := a_m + b_m(t - t_m) + c_i(t - t_m)^2 + d_i(t - t_m)^3 \tag{5.5}$$

The coefficients in the above family of polynomials are obtained by enforcing the following conditions required for natural cubic spline.

$$\begin{aligned} s_m(t_i) = g_m &= s_{m-1}(t_m),\ m = 1, \dots, r-1 \\ \dot{s}_m(t_m) &= \dot{s}_{m-1}(t_m),\ m = 1, \dots, r-1 \\ \ddot{s}_m(t_m) &= \ddot{s}_{m-1}(t_m),\ m = 1, \dots, r-1 \\ \ddot{s}_0(t_0) &= \ddot{s}_{r-1}(t_r) = 0 \end{aligned} \tag{5.6}$$

The natural cubic splines can be employed for interpolating a discrete time series, i.e. the spline function passes through the samples of the discrete time series. However the smoothing operator in the general noise filtering framework requires smoothing of the time series. This can be executed by minimization of:

$$\arg\min_{s(t_m)} \sum_{m=0}^{r} (m_i - s(m_i))^2 + \kappa \int s(t)^2 dt \tag{5.7}$$

where κ is the smoothing parameter. For $\kappa > 0$, the family of polynomials obtained using the above minimization is referred to as cubic spline smoothing. When $\kappa = 0$, it is equivalent to cubic spline interpolation. The algorithms involved in the automatic estimation of smoothing parameter and executing cubic smoothing spline are beyond the scope of this research. See [71], for more information on polynomial smoothing spline algorithms utilized in `SciPy`.

5.3.1. Setup in `SciPy`

Smoothing spline filtering is performed using the `UnivariateSpline` function in `SciPy` library of `Python3`. As shown earlier in this section, in addition to the interface displacement vector, time vector is also required for using smoothing spline smoother. The additional parameters for the smoothing operator such as smoothing parameter, weights etc. are set to their default values. The default value for κ is equal to h. The relative acceleration magnitude at the structure interface described in 5.1.2 shall be used for evaluating the effectiveness of smoothing spline filter.

5.3.2. Numerical results

The smoothing spline filter is executed for varying values of h. The smoothness function shall be employed again for comparing the smoothness of the averaged relative interface acceleration obtained with and without smoothing spline filter. The results of this analysis are tabulated in table 5.5. It can be seen that the smoothing spline filter is capable of dampening the noise in interface acceleration magnitude across all nodes on the structure interface. Also, when more previous time steps are employed to extrapolate interface displacement, the noise in averaged relative interface acceleration decreases. The least noise was observed when interface displacement from 21 previous time steps were employed in the smoothing operator. However, for $h > 21$, the noise in the averaged relative interface acceleration

Filter	h	κ	$\Theta(\mathbf{a}_\Gamma)$
No Filter	N/A	N/A	7.421
Smoothing Spline	5	5	6.258
Smoothing Spline	10	10	4.875
Smoothing Spline	15	15	4.671
Smoothing Spline	20	20	4.031
Smoothing Spline	21	21	4.022
Smoothing Spline	25	25	5042.877

Table 5.5: Cylinder with Trailing Flap — Smoothness function of averaged relative interface acceleration with and without smoothing spline filter

is amplified, indicating the shortcomings of extrapolating interface displacement from larger number of time steps. Also, the plot of averaged relative interface acceleration with and without smoothing spline filter are portrayed in figure 5.3. It can be seen that the magnitude of the spikes are smaller for the scenarios with smoothing spline filter (h=10,20). And, the magnitude of spikes are much smaller for $h = 20$ in comparison to $h = 10$ when smoothing spline filter is employed. However, on further examination of the plot for $h = 20$, it can be seen that the spikes of lesser magnitude are superimposed with a sinusoidal wave of increasing amplitude, whose period is approximately equal to 20 time steps, which is the h value. As seen in the plot for h=25, when this superposition is left unchecked, the amplitude of the sinusoidal wave grows exponentially with time. Suppose instead of 1000 samples in the current scenario, if there are 5000 samples, then it is probable that the smoothing spline filter with $h = 20$ would exhibit a similar phenomenon, thereby leading to non physical values for interface acceleration. For more information the gradual amplification of noise for $h > 21$, see section 5.6. Such phenomenon is not noticeable for the smoothing spline filter with $h = 10$. Moreover, the data storage requirement for the smoothing spline filter with $h = 10$ would be halved in comparison to the scenario with $h = 20$. Therefore, based on the analysis presented in this section, it is recommended to rely upon interface displacement data from 10 previous time steps for the smoothing spline filter.

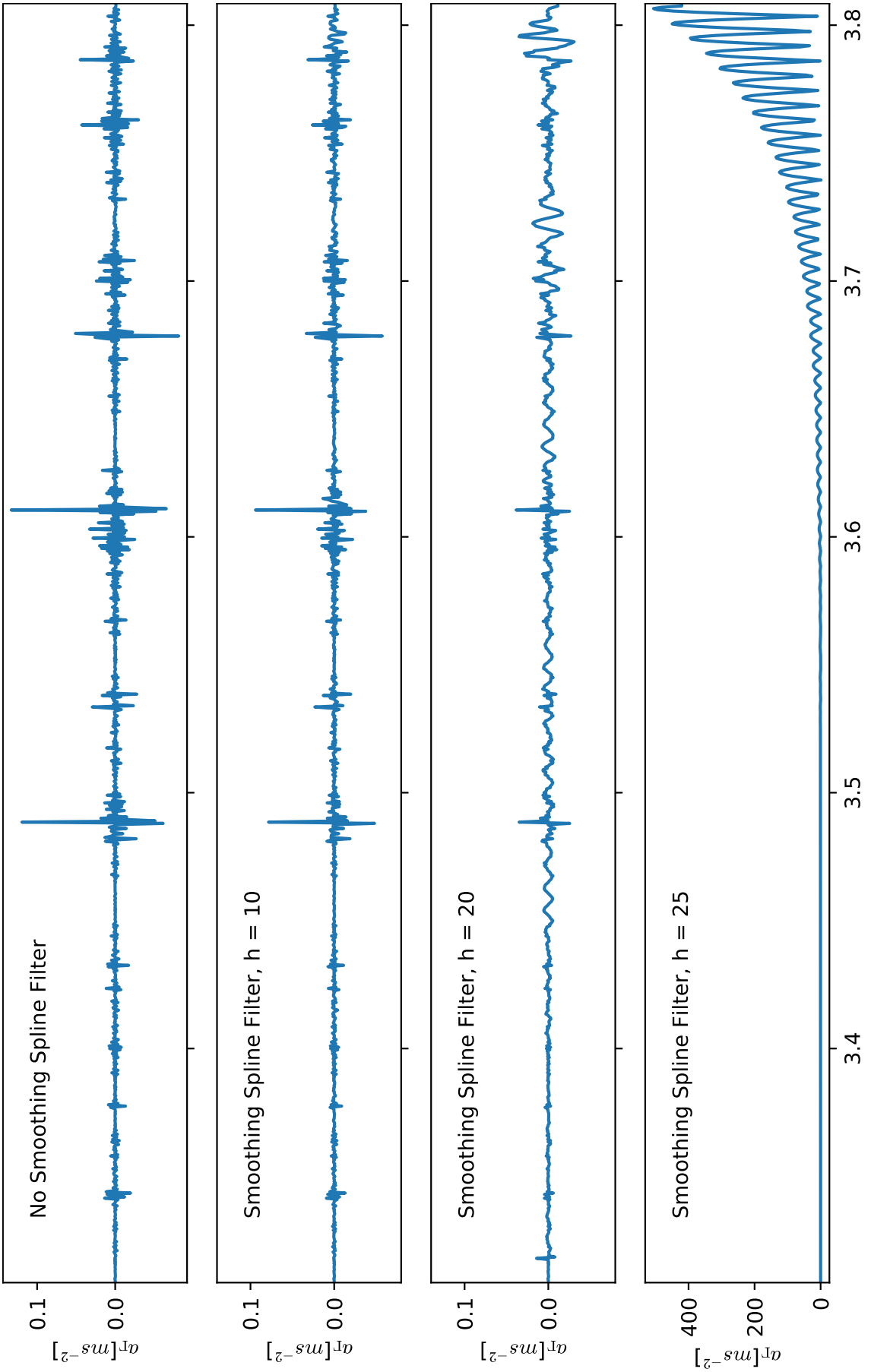


Figure 5.3: Cylinder with Trailing Flap — Averaged relative interface acceleration on Mesh 2ref2+2 configuration with and without smoothing spline filtering for time interval 3.308 - 3.808 [s]

5.4. Savitzky-Golay filter

Savitzky Golay filter is a popular numerical smoothing and differentiation algorithm developed by Savitzky and Golay in [1]. In this noise filtering technique, noise is filtered out by performing a polynomial interpolation on small time windows of a time series through minimisation of least squares. Savitzky-Golay filter requires odd number of terms in this time window. And, it should be a contiguous subset of the time series to be smoothened. If l is the number of terms in the time window, then $l < h$. Unlike the computation of extrapolation coefficients for least squares minimisation in the IQN-ILS framework, an alternative algorithm based on convolution has been employed in lieu of the expensive QR decomposition in Savitzky-Golay filter. Suppose a time series $\mathbf{g}$ has h samples, then convolution using time window with l samples is described as:

$$g_j^* = \frac{1}{M} \sum_{m=-\frac{l-1}{2}}^{\frac{l-1}{2}} e_m g_{j+i} \quad \forall j \in [\frac{l-1}{2}, h - \frac{l-1}{2}] \tag{5.8}$$

where g_j^* is the smoothened value, e_i are the convolution coefficients and M is the normalization factor. Convolution is analogous to weighted average of the samples in the time window, with the exception that some of the weights can also be negative. The methodology to compute the coefficients/normalization factor for smoothing using least squares polynomial interpolation is shown in [1]. The convolution table containing the convolution coefficients and normalization factor for all possible time window sizes of cubic polynomial smoothing is tabulated in appendix C. Unlike smoothing spline filter, Savitzky Golay filter does not require the independent time vector to perform interpolation. However, it requires all the samples in a time series to have identical time step.

5.4.1. Setup in `SciPy`

Savitzky-Golay filtering is performed using the `savgol_filter` function in `SciPy` library of `Python3`. The additional parameters for this function are the length of time window l, order of the polynomial, and `mode` parameter. The `mode` parameter decides what values are assigned for samples outside the interval $[\frac{l-1}{2}, h - \frac{l-1}{2}]$. The horizontal and vertical displacements for the current time step are the last terms in the $d_x temp$ and $d_y temp$ arrays of the general filtering framework in pseudocode 1. The last terms are definitely outside the Savitzky-Golay time window of $[\frac{l-1}{2}, h - \frac{l-1}{2}]$. Therefore, the Savitzky-Golay filter cannot compute the smoothened value of interface displacement for current time step. Default value for `mode` is `interp`. In the `interp` mode, samples outside the interval $[\frac{l-1}{2}, h - \frac{l-1}{2}]$ are assigned values based on cubic polynomial interpolation from the smoothened samples within the interval. Thus, the smoothened value for the current time step is obtained as a cubic interpolation from the nearest smoothened samples that are obtained using convolution.

5.4.2. Numerical results

The Savitzky-Golay filtering is executed for various values of h and l. For a particular value of l, the effectiveness of noise dampening is marginally affected by h value. Therefore, in the results for all possible l values, the h values were selected such that it resulted in the least noise in the averaged relative interface acceleration after an optimization study. Some of the results of this analysis are tabulated in table 5.6.

The filter with time window length of 21 samples has the least noise in relative interface acceleration. And, as observed for the smoothing spline filter, the noise in averaged $\mathbf{a}_\Gamma$ decreases with more number of samples in the time window up to 23 samples. For filtering with time window of 23 samples or higher results in additional noise in averaged $\mathbf{a}_\Gamma$. The results of Savitzky-Golay filter for $l = 11, 21, 25$ are portrayed in comparison to data without filtering in figure 5.4. For the scenarios with $l = 21, 25$ the noise has been dampened, but the noise seems to be superimposed with a sinusoidal wave that increases in amplitude. Especially, for the filtering with $l = 25$, the noise due to the sinusoidal artefact is much larger in comparison to the actual numerical noise. These observations are similar in comparison to the ones made for smoothing spline filtering. For more information on the gradual amplification of noise for $l > 21$, see section 5.6. Therefore, the recommended time window length for performing Savitzky-Golay filtering is 11 samples. As for the data from h previous time steps is concerned, it can

Filter	l	h	$\Theta(\mathbf{a}_\Gamma)$
No Filter	N/A	N/A	7.421
Savitzky-Golay	5	16	7.094
Savitzky-Golay	11	17	4.874
Savitzky-Golay	15	26	4.710
Savitzky-Golay	21	22	4.053
Savitzky-Golay	25	30	1551.326

Table 5.6: Cylinder with Trailing Flap — Smoothness function of averaged relative interface acceleration with and without Savitzky-Golay filter

be assigned any value that is greater then the number of samples in time window of Savitzky-Golay filtering. Again, higher value of h requires more computational memory. Thus, it is prudent to choose h that is in the neighbourhood of s value.

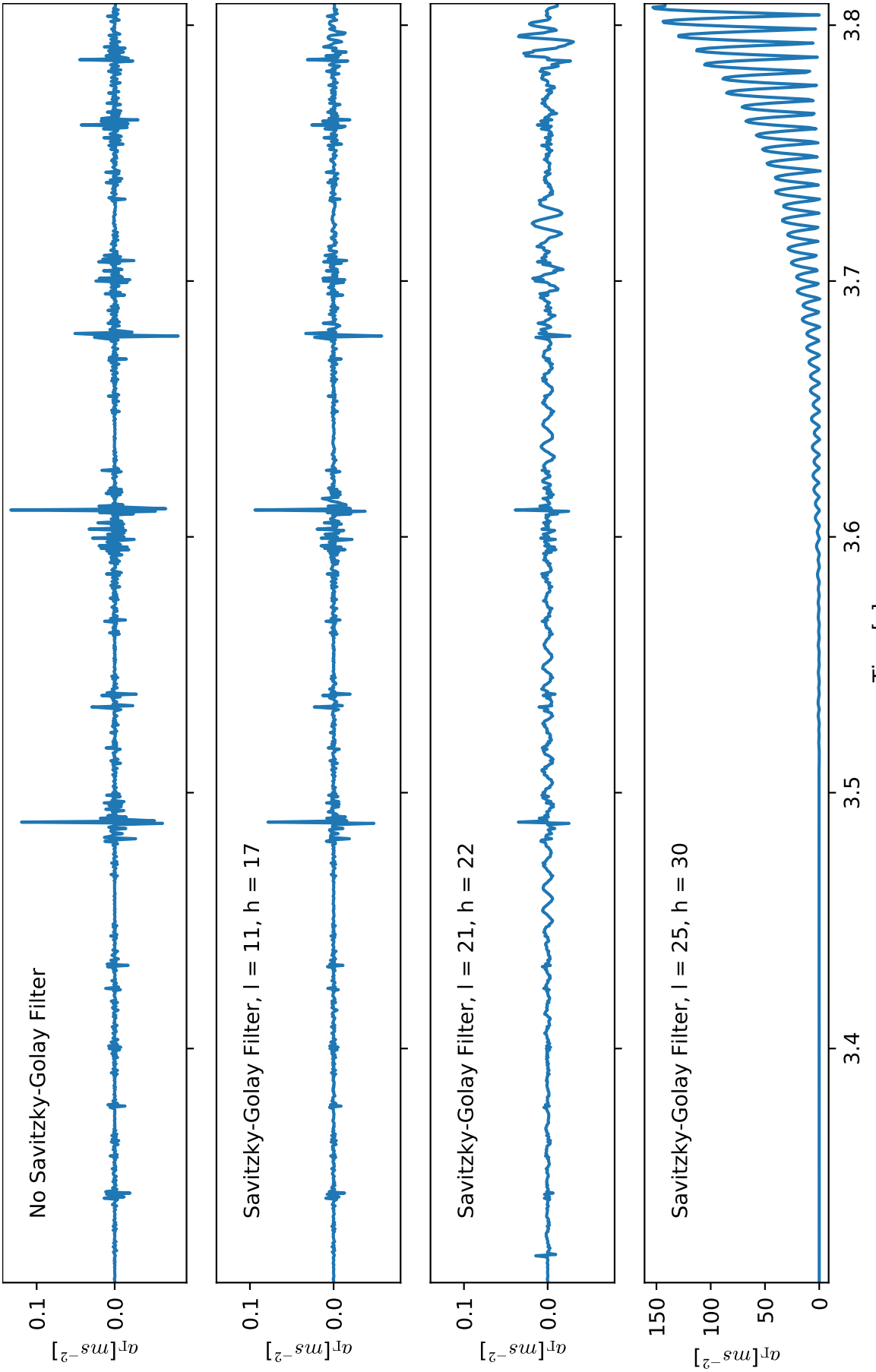


Figure 5.4: Cylinder with Trailing Flap — Averaged relative interface acceleration on Mesh 2ref2+2 configuration with and without Savitzky-Golay filtering for time interval 3.308 - 3.808 [s]

5.5. Sinusoidal curve fitting filter

The displacement plots in figure 5.1 resemble a sinusoidal wave of varying amplitude. This served as a motivation to fit sinusoidal curves for all time intervals within the time series in view of smoothening interface displacement in the general noise filtering framework. Suppose a time series $\mathbf{g}$ has h samples. Let the sine function to approximate the samples within the time interval be defined as ζ:

$$g(t, \mho) := \zeta \sin(\eta t + \phi) + \tau \tag{5.9}$$

where ζ is the amplitude, η is the angular frequency, ϕ is the phase angle and τ is the offset. $\mho$ is nothing but the set of above mentioned parameters. Let the residual for the curve fitting be defined as:

$$r_m := g(t_m) - g(t_m) \tag{5.10}$$

Then the sinusoidal curve fitting is performed by computing the parameters in $\mho$ such that the residual is minimized using least squares. It is executed by solving the below mentioned system of over determined equations.

$$2 \sum_{m=0}^{h} r_m \frac{\partial r_m}{\partial \mho_j} = 0 \tag{5.11}$$

Since sine function is non linear, the system matrix is non linear. Therefore, it is not possible to obtain a closed form of solution for the parameters. The parameters are obtained in an iterative fashion. The curve fitting routine of `SciPy` uses Levenberg–Marquardt algorithm to resolve non-linear least squares minimization. The details of LM algorithm are beyond the scope of this research. See [72, 73] for more information about this algorithm.

5.5.1. Setup in `SciPy`

Curve fitting optimization is executed in python using `curve_fit` function in `SciPy` library. A model sine function similar to the one in (5.9) with its independent variable and parameters is created. The only difference is that the angular frequency is treated as an independent variable since the optimization was much more troublesome when it is employed in optimization. The frequency is computed from $\Im$ previous samples using the procedure discussed in section 3.3.5. Since at least one period of samples is required to compute the frequency, the curve fitting optimization can only start after $\Im$ time steps that correspond to the first period. As, the optimization continues, the frequency is computed for one whole period with respect to the interface displacement in the current time step. Both interface displacement and time samples are required for performing curve fitting. Initial values for amplitude, phase angle and offset have to be provided. Initial estimate of amplitude is computed as $\sqrt{2}\sigma(\mathbf{g})$, where $\sigma(\mathbf{g})$ is the standard deviation of $\mathbf{g}$. Initial estimate of offset and phase angle is assumed as $\overline{\mathbf{g}}$ and 0 respectively.

5.5.2. Numerical Results

The sinusoidal curve fitting is employed for various values of h. The results of this analysis are tabulated in table 5.7. For all scenarios, the sinusoidal curve fitting filter resulted in addition of noise to the original relative interface acceleration. This higher smoothness function value can be attributed to the large but infrequent spikes in relative interface acceleration as shown in figure 5.5. The original reasoning for the random spikes in sinusoidal filtering was attributed to the shift in the computed frequency at the instance of spike occurrence. Therefore, the sinusoidal curve fitting optimization was repeated, but this time the frequency of the signal employed for non-linear least squares minimization is constrained to be fixed. This fixed frequency was computed from the last period of the signal. The plots for this analysis are portrayed in figure 5.6. It can be seen that the spikes still exist.

On further investigation, a warning text was shown by the `curve_fit` function when the noise filtering is being performed using $h = 10$ for the node located at $[0.58, 0.19]m$. It stated, "Covariance of the parameters could not be estimated". The horizontal and vertical acceleration plots for this node have been plotted in figure 5.7. In the neighbourhood of $t = 3.7[s]$, a huge numerical artefact is observed for horizontal acceleration. This matches with the location of the spike for $h = 10$ in figure 5.6. Further, when the calculated parameters were investigated, it was found for $t = 3.693[s]$ that the predicted

Filter	h	$\Theta(\mathbf{a}_\Gamma)$
No Filter	N/A	7.421
Sine Fit	5	17.116
Sine Fit	10	8.8302
Sine Fit	15	14.729
Sine Fit	20	24.834
Sine Fit	25	29.474
Sine Fit	30	50.697

Table 5.7: Cylinder with Trailing Flap — Smoothness function of averaged relative interface acceleration with and without sinusoidal curve fitting filter

amplitude jumps from -9.828×10^{-4} to -3.942×10^{-4}, and the predicted phase angles jumps from -1.292 to 9.631×-7 for the sine curve fitting of horizontal interface displacement, resulting in this large artefact.

For the scenarios with higher h value, the noise is superimposed on a sinusoidal wave. For more information the gradual amplification of noise for $h > 21$, see section 5.6. Also, due to non linear least squares minimization, this filtering is observed to be computationally more expensive than the other two noise smoothing operators.

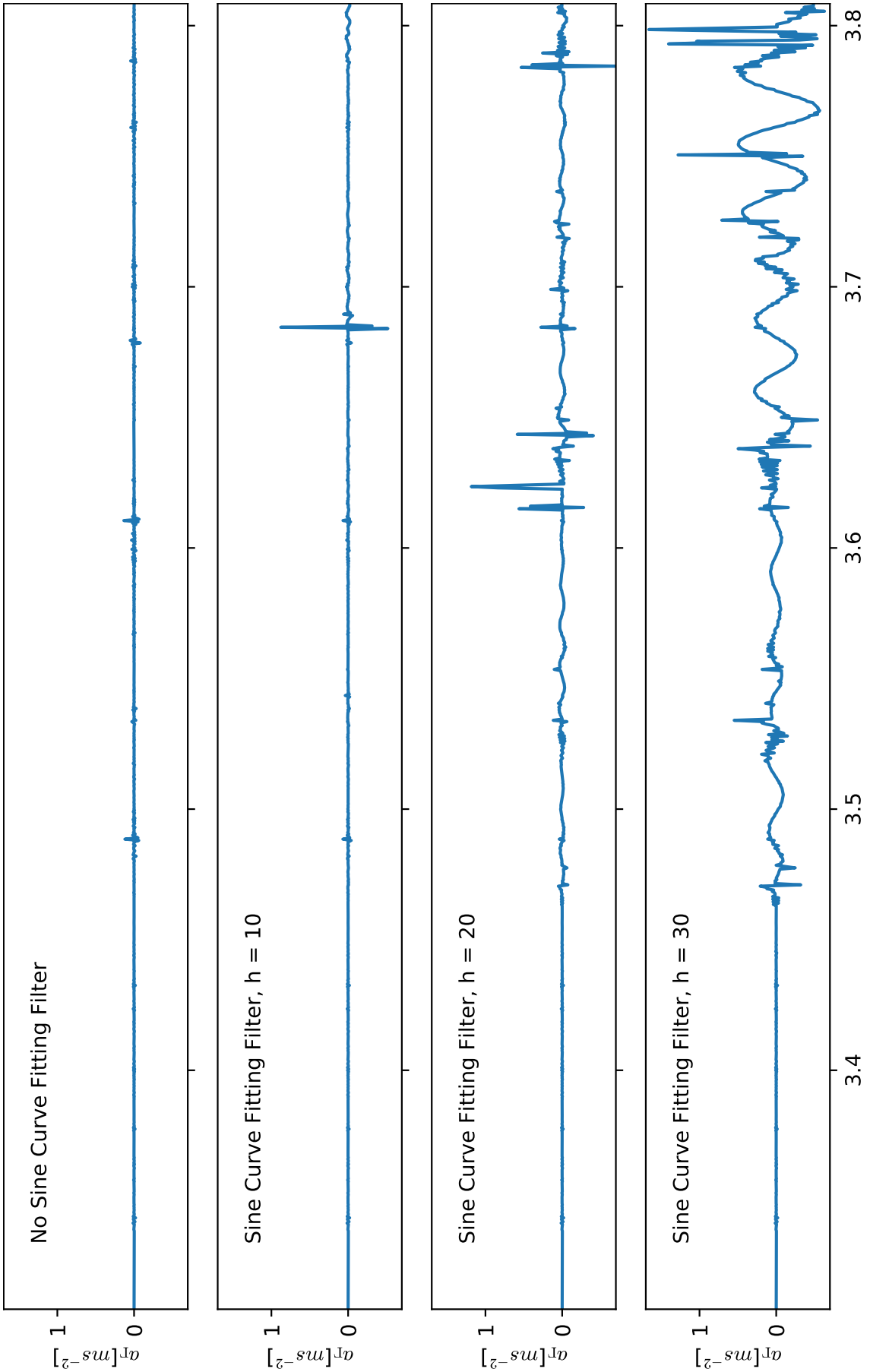


Figure 5.5: Cylinder with Trailing Flap — Averaged relative interface acceleration on Mesh 2ref2+2 configuration with and without sinusoidal curve fitting filter for time interval 3.308 - 3.808 [s]

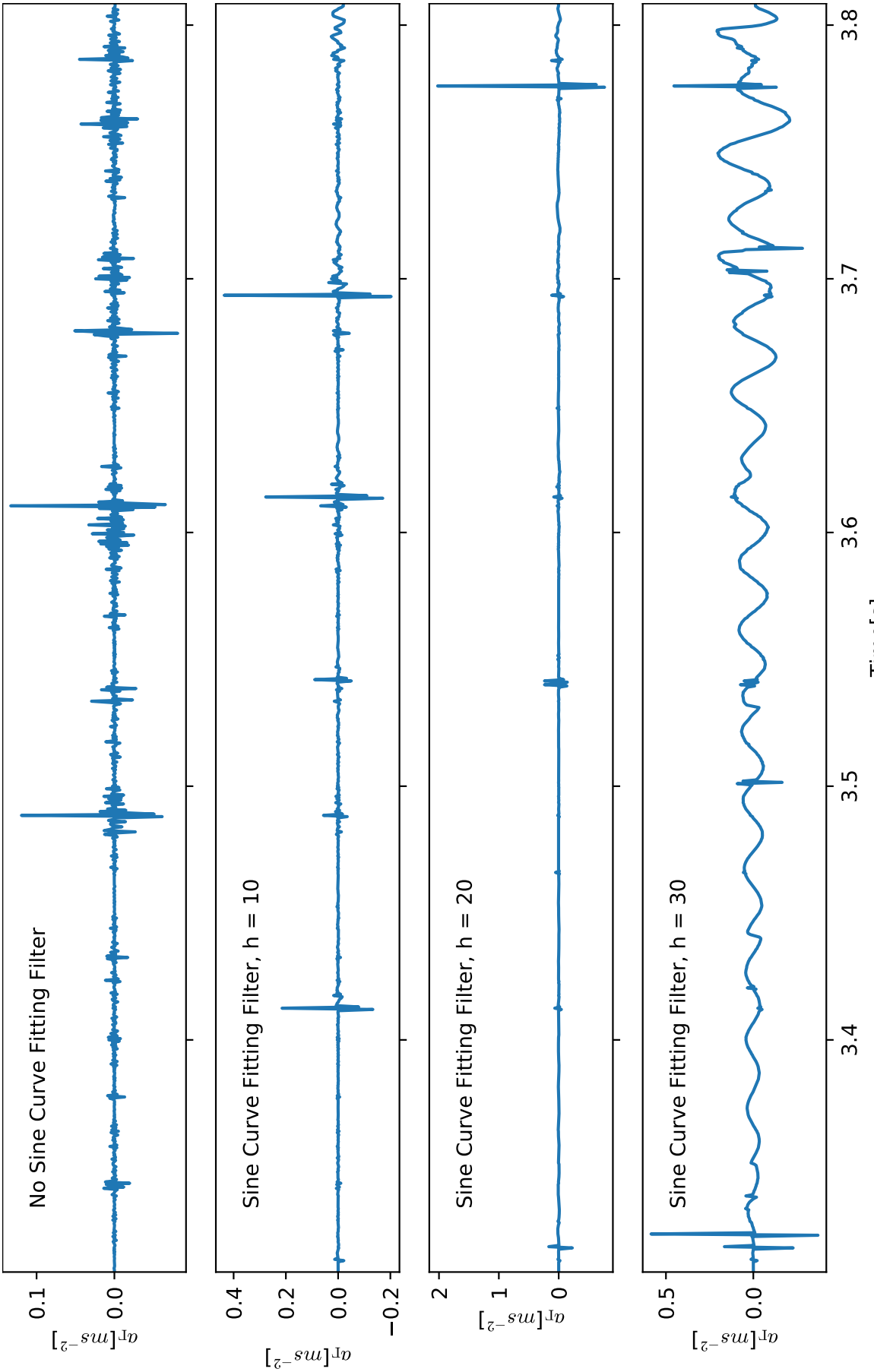
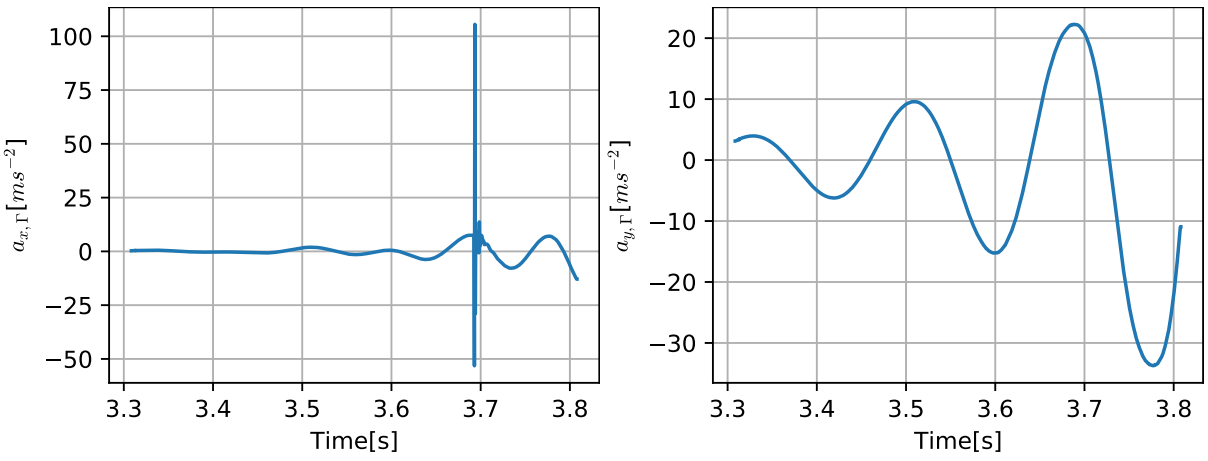


Figure 5.6: Cylinder with Trailing Flap — Averaged relative interface acceleration on Mesh 2ref2+2 configuration with and without sinusoidal curve fitting filter using fixed frequency for time interval 3.308 - 3.808 [s]



(a) Horizontal Acceleration

(b) Vertical Acceleration

Figure 5.7: Cylinder with Trailing Flap — Horizontal and vertical interface acceleration for node located at (0.58,0.19) on Mesh 2ref2+2 configuration smoothened with sinusoidal curve fitting filter using fixed frequency for time interval 3.308 - 3.808 [s]

5.6. Sine wave superposition in noise filter

For all the three smoothing filter operators, when larger number of previous samples are employed for smoothing, it was generally observed that although the actual noise in the signal is dampened more significantly, it was also responsible for causing sinusoidal superposition in relative interface acceleration. To investigate this issue further, the averaged smoothened horizontal and vertical interface displacements are compared to their original counterparts in figures 5.8 and 5.9. It can be seen that the smoothened interface displacement oscillates around the original interface displacement, resulting in the sinusoidal superposition phenomenon. This is because of successive smoothing of time steps using smoothened values from previous time steps. Smoothing causes the interpolated value to diverge from the true interface displacement. It results in the systemic error slowly creeping into the measurements resulting in the sinusoidal superposition by the general noise filtering framework. When the filters are integrated with `preCICE`, it is expected that the fluid solver will generate different interface force for the smoothened displacement, which causes the structure solver to give different displacement for next sub-iteration. As long as the displacement tolerance is satisfied, the systemic error will not creep into the solution.

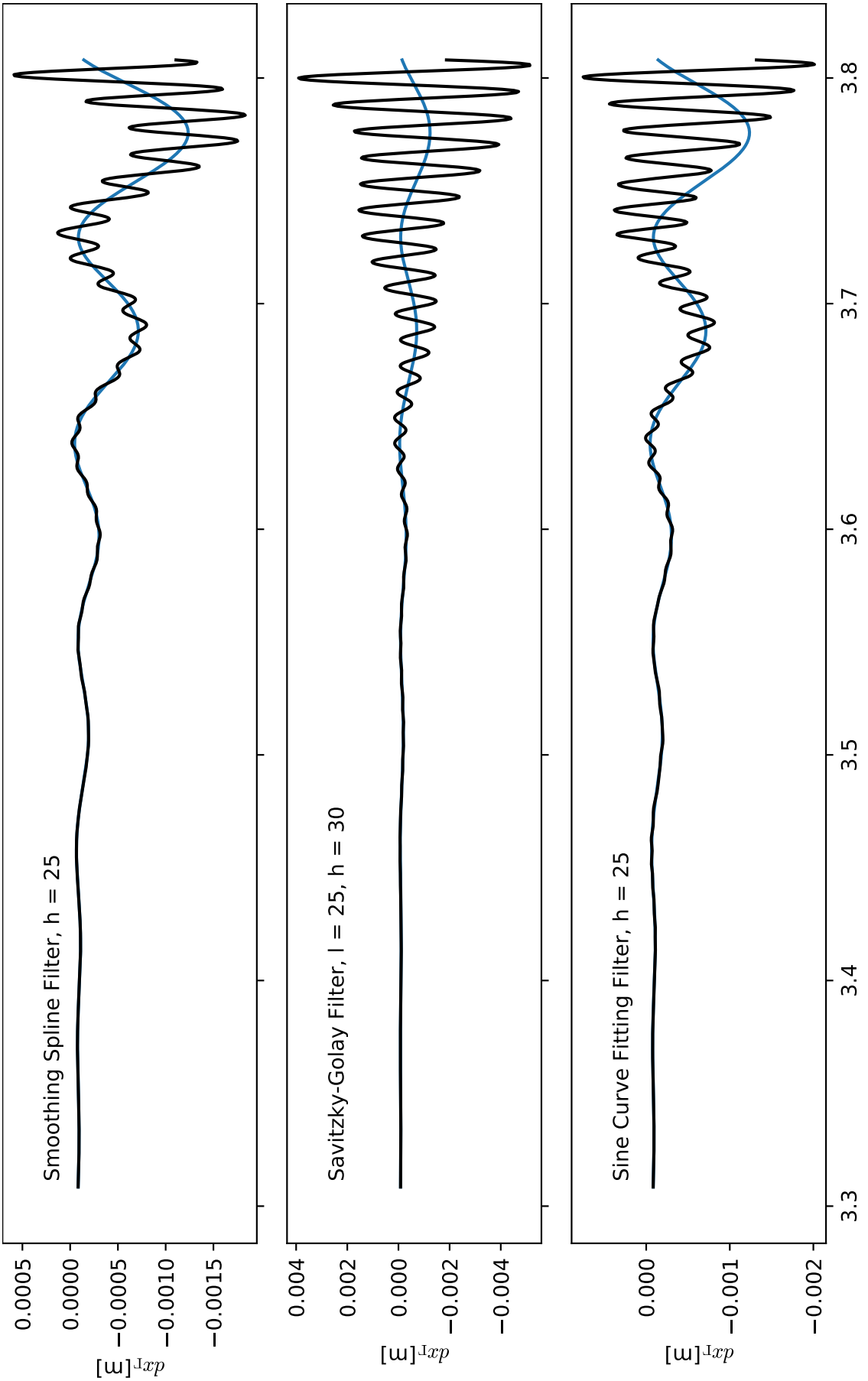


Figure 5.8: Cylinder with Trailing Flap — Averaged horizontal interface displacement on Mesh 2ref2+2 configuration for noise smoothing with larger number of previous samples for time interval 3.308 - 3.808 [s]. —— : Without filter; —— : With filter.

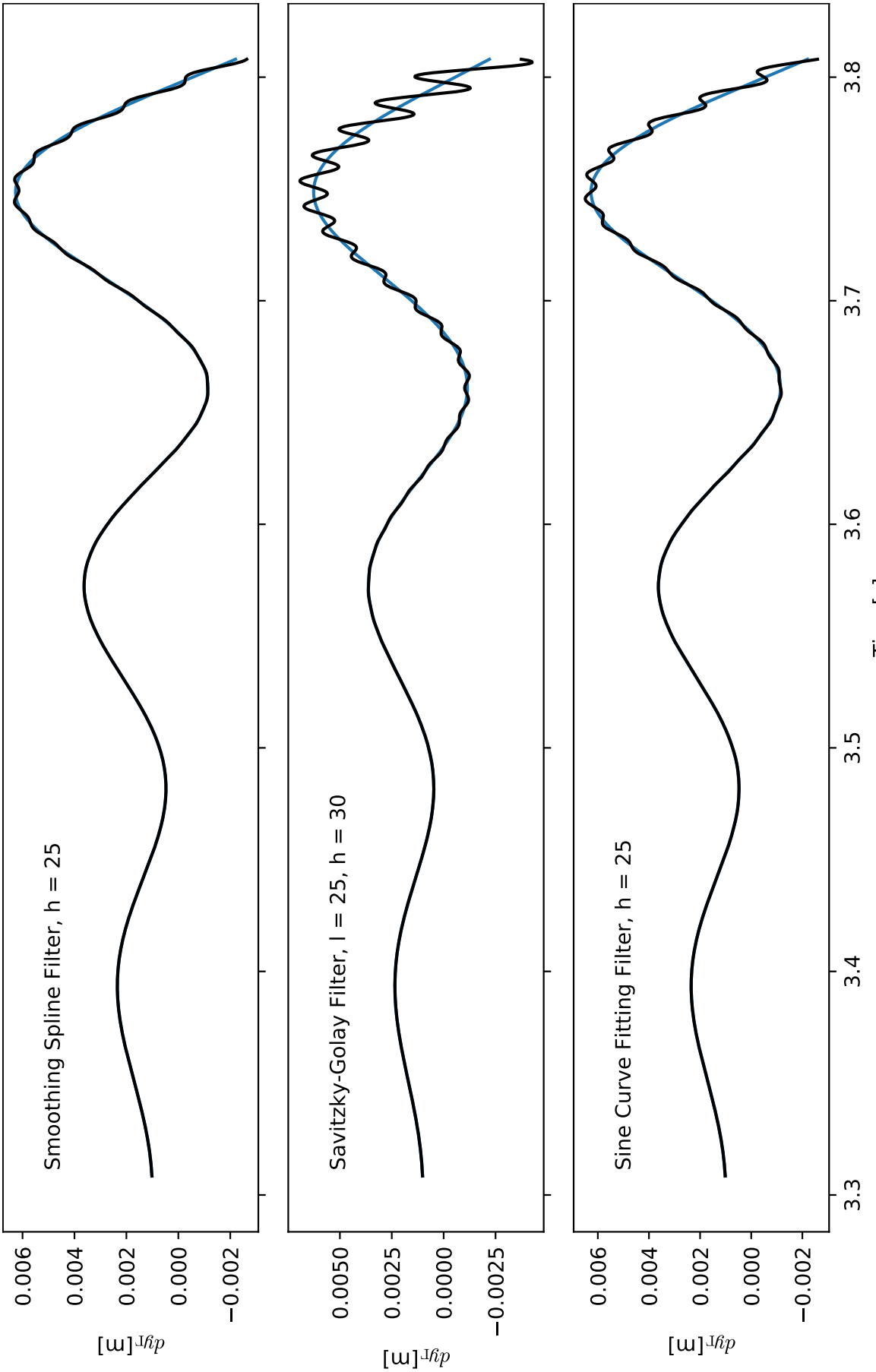


Figure 5.9: Cylinder with Trailing Flap — Averaged vertical interface displacement on Mesh 2ref2+2 configuration for noise smoothing with larger number of previous samples for time interval 3.308 - 3.808 [s]. —: Without filter; —: With filter.

5.7. Conclusions

In section 5.1, the hypothesis for the reasoning behind noise formation in force data was proven using correlation analysis between the averaged relative interface acceleration and force magnitudes. A general noise filtering framework to dampen the noise in interface displacement was proposed in 5.2 in hope of obtaining noise free force data from the fluid solver. Three noise smoothing operators for the general noise filtering framework — Smoothing spline filter, Savitzky-Golay filter and Sinusoidal curve fitting filter were discussed in sections 5.3, 5.4 and 5.5 respectively. Smoothing spline and Savitzky-golay filter showed promising results when the smoothing is performed using less number of previous interface displacement samples. Huge spikes were observed at arbitrary locations for sinusoidal curve fitting filter. This is because of the non linear least squares minimization failing for certain combination of variables and parameters. And, all three filters exhibited sinusoidal superposition phenomenon when the smoothing is performed with larger number of previous interface displacement samples. This is due to a systemic error generated by repeated smoothing at every time step in the absence of correcting mechanism by sub-iterations in the solvers. The performance of these filtering frameworks were computed on interface displacements that was already computed. Unfortunately, complete fulfilment of the second half of revised research objective requires these filtering frameworks to be integrated with `preCICE`, which could not be finished due to time constraints.

6

Conclusions and Recommendations

The original objective of this research was to evaluate the robustness of IQN-ILS temporal coupling framework for partitioned FSI. It fell through when the forces computed on the flap in the Cylinder with trailing flap benchmark test case was plagued with noise for certain combinations of fluid and structure parameters. Also, filtering techniques were developed towards the end of this research in view of smoothing out the noise generated by IQN-ILS framework.

6.1. Conclusions

Flow and structure solvers — `OpenFOAM` and `CalculiX` were validated in view of utilizing them for partitioned FSI. `OpenFOAM` was validated with the CFD benchmarks from [55]. `CalculiX` was validated with the CSM benchmarks from [55]. A FSI test-bed has been formulated by integrating `OpenFOAM` and `CalculiX` with their respective adapters within `preCICE`. The FSI simulations in the above mentioned test-bed were performed with IQN-ILS temporal coupling framework. This test-bed was validated for the steady state FSI1 benchmark prescribed in [55]. However, complications arose when validating the transient benchmarks FSI2 and FSI3. With the allocated computational resources, the FSI benchmarks were only possible for Mesh 1+1, Mesh 2+2 and Mesh 3+3 configurations. Due to the poor resolution of the meshes mentioned above, the computed quantities were on average 10% off the reference data. And, the forces computed in the FSI3 benchmark exhibited significant noise, which is not desired for robustness evaluation. Since, the investigation and addressal of the force noise from FSI3 benchmark was feasible in this research, it was adopted as the revised objective for this research.

A hypothesis was proposed in section 4.2, wherein the error in interface displacement, if any, will amplify by three orders of magnitude for $\Delta t = 0.0005s$ on differentiation. Since, force acting on the flap is a function of first derivative of interface displacement, minuscule error in interface displacement will amplify into significant error in force, which manifests as noise. The findings in section 4.3 support the above mentioned hypothesis indirectly by showing that smaller interface residual leads to smaller iterative error, resulting in lower noise in force data. Several other settings of the temporal coupling framework were investigated to check their influence on noise in sections 4.4 - 4.8. From these analyses, it was concluded that IQN-ILS(10) framework on Mesh 2ref2 + 2 configuration with QR-1 filtering tolerance of 10^{-8} exhibited the least noise in force data. But, the optimized IQN-ILS framework still generated more noise in force data in comparison to fixed point iteration with constant under-relaxation. In fact, the forces computed using this framework were noise free. Among all the noise prevention measures that was attempted, fixed point iteration with constant under-relaxation is the only measure that is capable of preventing noise in force data. However, this is at the cost of excessive amount of expensive sub-iterations in comparison to IQN-ILS framework. Even though this preventive measure satisfies the first part of revised objective, it is not much of an improvement over the status quo in industrial applications.

A general noise filtering framework was proposed as a noise mitigation measure for the force noise in IQN-ILS framework. The noise mitigation is performed by eliminating or dampening the minuscule errors in interface displacement before it is coupled with the fluid solver. Three noise filters were proposed to function as the smoothing/filtering operator in this framework. They were smoothing spline

filter, Savitzky-Golay filter and sinusoidal curve fitting filter. These filters were tested on an interface displacement series that was already computed. Smoothing spline and Savitzky-Golay filter showed promising results in regards to dampening the noise in interface displacement. However, the effectiveness of these filters in removing noise from force data was not tested, since it requires integrating the noise filtering framework with `preCICE`, which is pending as of this writing. Unfortunately, this means that the second half of the revised objective has only been partially fulfilled.

6.2. Recommendations

Based on the work during this research, the author proposes the following recommendations with the hope of improving the quality of life experience with `preCICE`, `OpenFOAM`, `CalculiX` and the noise filtering frameworks proposed in this research for future researchers.

- User guides and tutorials for dynamic mesh formulation of the `OpenFOAM` solvers are sparse as of this writing. Invocation of dynamic mesh formulation within `OpenFOAM` can be much more streamlined, thereby making it easier for third party programs to utilize it.
- Due to the above issue additional sub routines have to be developed within `OpenFOAM` adapter of `preCICE` to support dynamic mesh formulation for FSI. At this moment, only Laplacian smoothing mesh motion is supported. High aspect ratio cells near the trailing edge of flap get distorted in FSI2 and FSI3 benchmarks. If support for RBF interpolation mesh motion is introduced, the above mentioned issue can be relieved. Until then, it is recommended that the flow mesh does not employ very high aspect ratio cells.
- In quasi 2D coupling, there is an issue with the coupling of second order elements of structure with the fluid cells at the interface. The force from the fluid face centre is not equally distributed among the vertex and edge centred nodes at certain locations on the upper flap surface. Resolving this will allow the employment of second order elements in structure mesh for FSI simulations.
- Restarting simulations with `CalculiX` is limited. Binary files required for `CalculiX` is created when the simulation finishes. Addition of feature to save the latest binary file within the `STEP` function of `CalculiX` can fix it.
- Aitken's framework exhibited significantly higher noise in the force data. Due to time constraints it could not be investigated. Further investigation of noise generation in Aitken's framework is desirable.
- Implementation of multi-level/fidelity optimization to `preCICE` is beneficial for performing FSI simulations on finer meshes like Mesh 5+5 configuration and above.
- Trigonometric definition of sine function has been employed for formulating the sinusoidal curve fitting filter. This resulted in crash of the `curve_fit` optimization in `SciPy` when the initial frequency guess is farther away from correct value. Adopting the series definition of sine function in a future version can circumvent this issue during curve fitting optimization.
- Integration of the noise filtering framework with `preCICE` is useful for evaluating the effectiveness of the proposed filters in a live simulation. This can be executed through the python callback interface in `preCICE`.

A

Laplacian Mesh Motion Insufficiency

As mentioned in section 3.3.5, the CFD solver validation was initially performed with a different batch of flow meshes. Here too, seven grids with different number of hexahedron cells were generated to perform mesh independence study for all the CFD tests. A coarse faithful reproduction of the grids employed in this study is illustrated in figure A.1. Also, the number of cells in each mesh is tabulated in table A.1.

The spatial distribution of mesh cells in the older batch of fluid meshes were designed such that it accurately captures the viscous and vortex shedding effects from the geometry. In comparison to the new batch of flow meshes, it has finer spatial discretization near the geometry, and coarser spatial discretization away from the geometry. Another consequence of this optimization is the high aspect ratio cells in the wake region. On average, the new batch of flow meshes have 25% more cells than the old batch. This results in faster computation time for the older batch of flow meshes.

CFD1, CFD2, CFD3 benchmarks were performed for these meshes. With the exception of tolerances for linear equation solver and final pressure, velocity; identical solver settings that were discussed in section 3.3.4 were employed for the benchmarks here as well. The linear equation solver and final pressure-velocity tolerances employed in this section were 10^{-7} and 10^{-6} respectively. Thus the tolerances employed for the older batch of meshes are not as strict as the ones employed for the new batch of meshes. The results of CFD1 and CFD2 tests are tabulated in table A.2. The most accurate lift and drag forces are within the acceptable limits of the reference solution. With the exception of drag from CFD2 test, the results for old batch are less accurate than the ones from new batch for steady state benchmarks. This can be attributed to the lenient tolerances employed for old batch of flow meshes. See table 3.5 for the CFD1 and CFD2 benchmark results for the new batch of flow meshes.

The CFD3 benchmark results for the old batch of fluid meshes are enclosed in table A.3. Again, the results are within the acceptable limits of the reference solution. However, it was not the case for the mean lift force obtained from the most refined new flow mesh. This can be due to the finer spatial discretization near the geometry for the old mesh, resulting in accurate modelling of vortex shedding. See table 3.6 for the CFD3 benchmark results for the new batch of flow meshes.

FSI benchmarks were attempted for old batch of flow meshes in conjunction with new batch of structure meshes discussed in section 3.4.1. Unfortunately, it was not possible to exploit the accurate modelling of vortex shedding from the old batch of flow meshes for FSI benchmarks. The FSI2 and FSI3 benchmarks failed for the Mesh 3+3 configuration. The failure of these benchmarks can be attributed

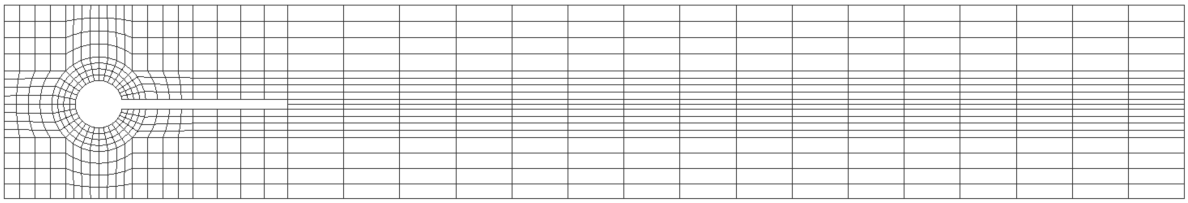


Figure A.1: Cylinder with Trailing Flap — Sample of earlier iteration of flow mesh with 672 hexahedron cells

Mesh	1	2	3	4	5	6	7
Cells	10752	24192	43008	67200	96768	131712	172032

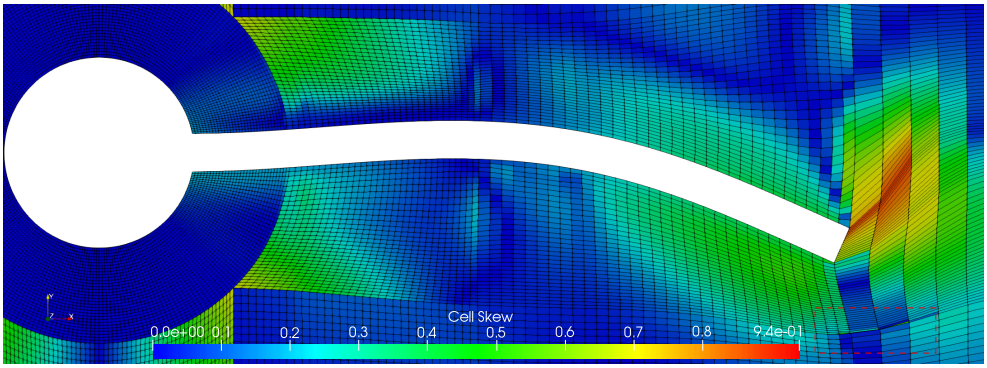
Table A.1: Cylinder with Trailing Flap — Number of flow domain cells in older batch of flow meshes

Mesh	**CFD1**		**CFD2**	
	Drag[N]	**Lift**[N]	**Drag**[N]	**Lift**[N]
1	14.1861	1.1061	137.4459	10.7630
2	14.2294	1.1066	136.9534	10.5592
3	14.2474	1.1082	136.7952	10.4590
4	14.2568	1.1099	136.7292	10.4153
5	14.2627	1.1113	136.6971	10.4005
6	14.2667	1.1123	136.6804	10.3989
7	14.2698	1.1131	136.6712	10.4026
Reference	14.29	1.119	136.7	10.53
Deviation[%]	0.1413	0.5255	0.0211	1.2102

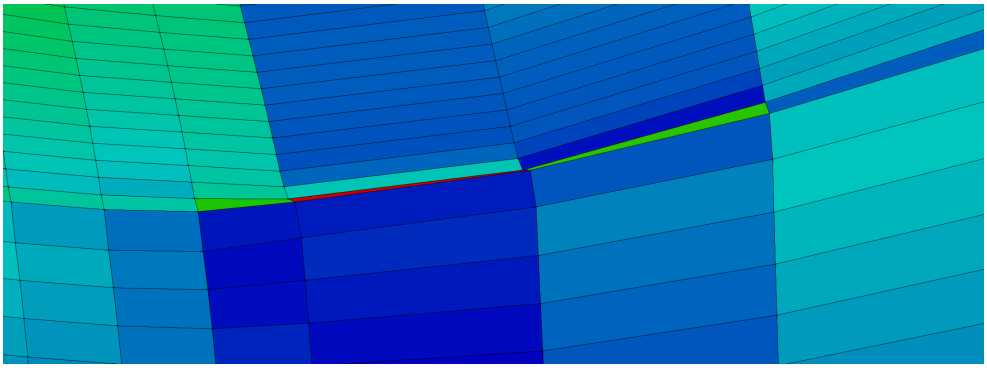
Table A.2: Cylinder with Trailing Flap — Numerical results of CFD1 and CFD2 tests for old batch of meshes

Mesh	**Drag**			**Lift**		
	Mean[N]	**Amp.**[N]	**Freq.**[s^{-1}]	**Mean**[N]	**Amp.**[N]	**Freq.**[s^{-1}]
1	451.0747	1.2841	7.9234	12.5865	212.2857	7.9145
2	443.9379	5.0896	4.3565	-33.7309	388.9900	4.3481
3	441.9272	5.4231	4.3893	-13.7305	422.6083	4.3883
4	441.2345	5.6532	4.4056	-12.8724	433.5968	4.4046
5	440.8654	5.7489	4.4116	-12.1553	438.8452	4.4109
6	440.6069	5.7858	4.4143	-12.0130	441.3366	4.4136
Ref.	439.45	5.6183	4.3956	-11.893	437.81	4.3956
Dev.[%]	-0.2632	-2.9804	-0.4252	-1.0086	-0.8055	-0.4084

Table A.3: Cylinder with Trailing Flap — Numerical results of CFD3 test for old batch of meshes



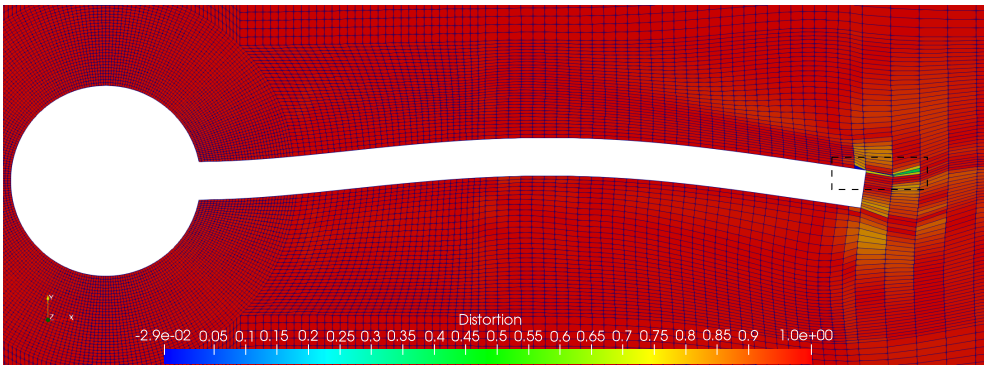
(a) Cell Skew plot without magnification



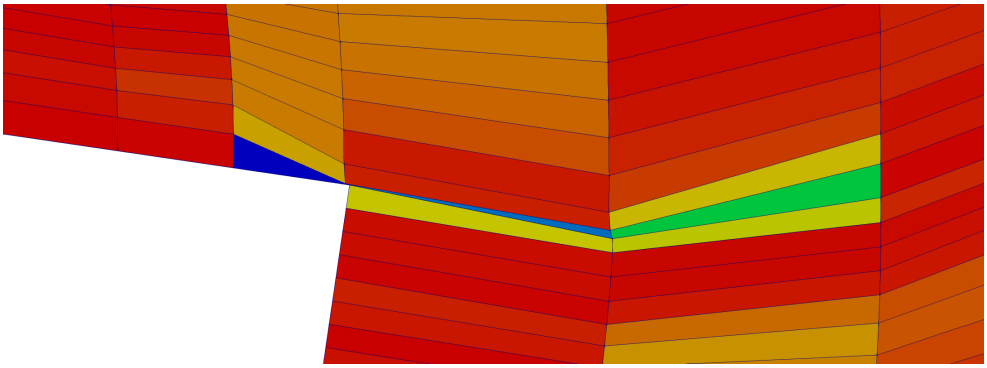
(b) Cell Skew plot of magnified portion

Figure A.2: Cylinder with Trailing Flap — Cell Skew Plot for FSI2 benchmark at 8.34s

to localized poor mesh quality, causing the flow solver to crash. The cell skewness and cell distortion plot for FSI2 and FSI3 benchmark respectively, at the instance of failure are enclosed in figures A.2 and A.3 respectively. The FSI2 simulation failure can be attributed to a single cell in the highlighted portion having very high skewness, whereas the FSI3 simulation failure can be attributed to couple of cells in the highlighted region having negative cell distortion. Both the failures are due to high aspect ratio cells in the downstream region of flap, which seem to behave poorly with the Laplacian mesh motion algorithm. In view of these issues, the new batch of meshes, that was discussed in section 3.3.1, were developed such that very high aspect ratio cells are avoided by ensuring uniform cell shape across the entire flow domain, thereby allowing the author to perform FSI2 and FSI3 benchmarks.



(a) Cell Distortion plot without magnification



(b) Cell Distortion plot of magnified portion

Figure A.3: Cylinder with Trailing Flap — Cell Distortion Plot for FSI3 benchmark at 4.075s

B

2D FSI with Quadratic Elements

A passing mention of CSM tests on old batch of structural meshes composed of quadratic hexbricks was made in section 3.4.4. Indeed, CSM benchmarks were successfully performed for this batch of meshes. Mesh 1 configuration of such family of meshes is illustrated in figure B.1. Based on the recommendation in [12], quadratic hexbricks (C3D20) with $2 \times 2 \times 2$ reduced integration points were employed for these meshes to avoid the issue of shear locking in slender beams. Six meshes with different number of quadratic hexbrick elements were generated to performed mesh independence study. Number of elements and nodes in each mesh is tabulated in table B.1.

The original batch of structural meshes were initially chosen for its capability to accurately represent non linear deformations in comparison to linear hexbrick structural meshes discussed in section 3.4.1. CSM1, CSM2 and CSM3 benchmark results for this batch of meshes are enclosed in tables B.2 and B.3 respectively. All the benchmark results are in good agreement with its reference counterparts. The most refined linear hexbrick mesh from the new batch of structural meshes has more number of elements than the most refined quadratic hexbrick mesh from the old batch of structural meshes. As expected, the CSM1 and CSM2 benchmark results for Mesh7 of linear hexbrick family of meshes is more accurate than the results for Mesh6 of quadratic hexbrick family of meshes. However, it is not the case for CSM3 bechmark results. In spite of lower number of elements, Mesh 5 results for quadratic hexbrick configuration are more accurate than the Mesh 7 results for linear hexbrick configuration. See tables 3.10 and 3.11 for the CSM benchmark results of linear hexbrick family of meshes.

Unfortunately, it was not possible to employ the old batch of structural meshes for FSI benchmarks. All three FSI benchmarks for the old batch of structural meshes in conjunction with old batch of fluid meshes crashes within few initial sub-iterations. This can be attributed to the presence of nodes on the midpoints of elements, resulting in additional row of nodes on the interface of the flap. The benchmarks prescribed in [55] are two dimensional in nature. However, `OpenFOAM` and `CalculiX` are three dimensional solvers. In order to perform 2D simulation with these solvers, mesh of 1m thickness had to be employed for both the solvers. Also, the third dimension along z axis had to be constrained in both the solvers in order to mimic a 2D simulation. The problem arises in the spatial coupling of force from the fluid interface to structure interface. Assuming that there is no void between the fluid and structure interface, and the edges of fluid cell coincide with the edges of solid elements;

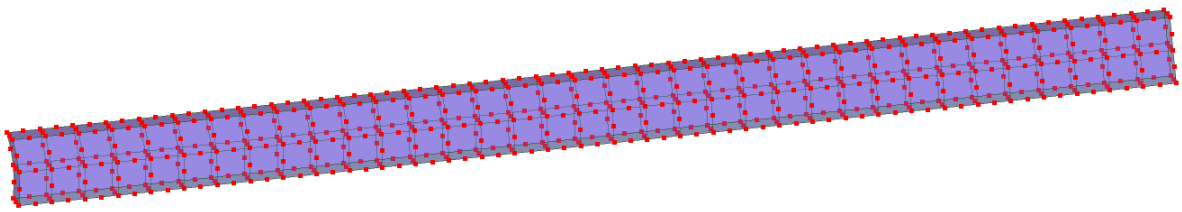


Figure B.1: Cylinder with Trailing Flap — Old iteration of structural mesh with 70 quadratic hexahedrons

Mesh	1	2	3	4	5	6
Elements	70	280	1120	4480	17920	71680
Nodes	678	2333	8583	32843	128403	507683

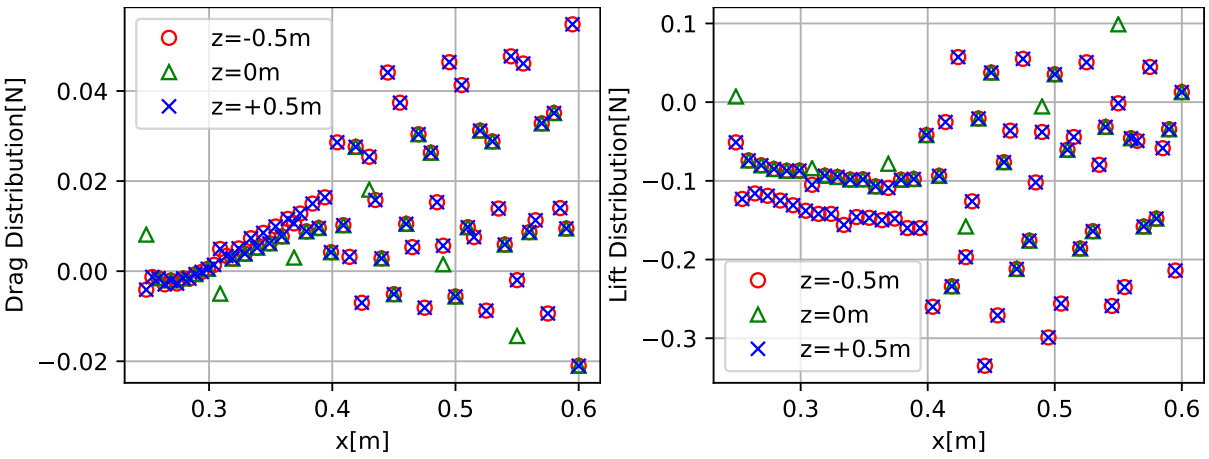
Table B.1: Cylinder with Trailing Flap — Number of structural domain elements and nodes in old batch of structural meshes

Mesh	**CSM1**		**CSM2**	
	Ux[mm]	**Uy**[mm]	**Ux**[mm]	**Uy**[mm]
1	-7.1527	-65.9166	-0.4666	-16.9240
2	-7.1789	-66.0554	-0.4684	-16.9611
3	-7.1842	-66.0839	-0.4687	-16.9686
4	-7.1865	-66.0960	-0.4689	-16.9719
5	-7.1875	-66.1013	-0.4690	-16.9733
6	-7.1879	-66.1034	-0.4690	-16.9738
Reference	-7.187	-66.1	-0.469	-16.97
Deviation[%]	-0.0127	-0.0052	0.0002	-0.0226

Table B.2: Cylinder with Trailing Flap — Numerical results of CSM1 and CSM2 tests for old batch of structural meshes

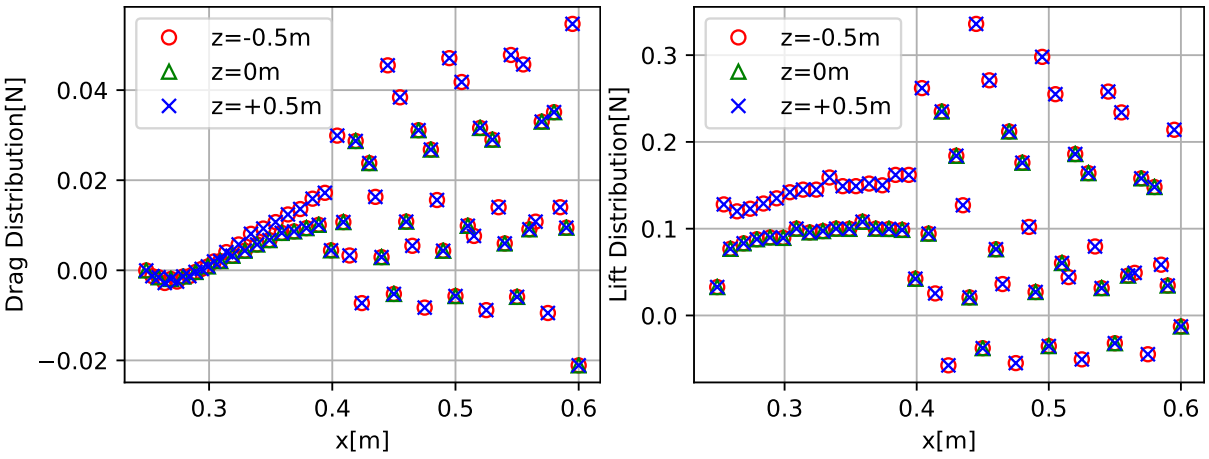
Mesh	**Ux**			**Uy**		
	Mean[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]	**Mean**[mm]	**Amp.**[mm]	**Freq.**[s^{-1}]
1	-14.2020	14.2017	1.09683	-63.4274	64.9164	1.09529
2	-14.2271	14.2266	1.09577	-63.5029	65.0454	1.09417
3	-14.2309	14.2305	1.09555	-63.5116	65.0707	1.09394
4	-14.2325	14.2321	1.09545	-63.5203	65.0853	1.09384
5	-14.2339	14.2336	1.09541	-63.5247	65.0908	1.09380
Ref.	-14.305	14.305	1.0995	-63.607	65.16	1.0995
Dev.[%]	0.4967	0.4994	0.3720	0.1295	0.1061	0.5184

Table B.3: Cylinder with Trailing Flap — Numerical results of CSM3 test for old batch of structural meshes



(a) Drag distribution on upper interface

(b) Lift distribution on upper interface



(c) Drag distribution on lower interface

(d) Lift distribution on lower interface

Figure B.2: Cylinder with Trailing Flap — Force distribution across midpoint and vertex nodes of upper and lower flap interface for FSI1 benchmark on old Mesh 1+1 configuration at 2s

force has to be mapped from one face centre on the fluid interface to three nodes on the structure interface, for a fixed x and y location; i.e the midpoint node coincides with face centre, whereas the vertex nodes are 0.5m away from the face centre in z direction. Conservative force coupling does not guarantee a faithful reproduction of force distribution from fluid interface to the solid interface. The force distribution plots on the interface nodes of the flap for the FSI1 benchmark on old Mesh 1 configuration for the fluid in conjunction with the quadratic hexbrick Mesh1 configuration for the flap at 2s are enclosed in figure B.2. The forces are equally distributed amongst the midpoint nodes and vertex nodes on the lower interface. Whereas for six nodes on the upper interface, forces for midpoint nodes are different from forces for vertex nodes. RBF interpolation ensures that the force distribution is preserved along x and y axis during mapping. However, 2D force mapping requires the force on midpoint node to be equal to the force on vertex node for a fixed x and y location. This can be attributed to the global RBF interpolation having issues in executing 2D mapping between pseudo 3D meshes. Since FSI1 benchmark has a steady state solution, and the forces involved in the coupling are of lesser magnitude in comparison to other FSI benchmarks, the FSI1 simulation for this configuration was able to generate a steady state solution. But the computed horizontal displacement at point A was about an order of magnitude lower than the reference solution. Also, this configuration is expected to fail for FSI2 and FSI3 benchmarks owing to the magnitude of the coupling quantities involved. Thus, it would be a futile exercise in employing quadratic hexbrick mesh for FSI validation.

To circumvent the above mentioned issue, a partial force mapping was attempted by coupling the force from face centres of the old batch of fluid meshes to only the vertex nodes on the solid

Mesh	**Ux**[mm]	**Uy**[mm]	**drag**[N]	**lift**[N]
1 + 1	0.0223	0.8685	14.1870	0.7299
2 + 1	0.0227	0.8758	14.2302	0.7316
Ref.	0.0227	0.8209	14.295	0.7638
Dev.[%]	0.1701	-6.6848	0.4533	4.2197

Table B.4: Cylinder with Trailing Flap — FSI1 benchmark results for old batch of fluid and structure meshes

interface. The results for FSI1 benchmark for Mesh 1+1 and Mesh 2+1 configurations are enclosed in table B.4. Very good results were observed for FSI1 benchmark for horizontal displacement and force. Yet, coupling fluid meshes with quadratic hexbrick Mesh 2 and finer grids resulted in the failure of simulation. This can be attributed to the very large number of structure nodes on the interface for C3D20 brick elements in comparison to the face centres on the fluid interface, causing issues similar to the ones discussed in section 4.6. Therefore, in view of all the issues faced with quadratic hexbrick meshes, it was decided to employ linear hexbrick meshes for performing FSI benchmarks.

C

Convolution table for Savitzky Golay Filter using Cubic Polynomials

Points/I	25	23	21	19	17	15	13	11	9	7	5
j-12	-253										
j-11	-138	-42									
j-10	-33	-21	-171								
j-9	62	-2	-76	-136							
j-8	147	15	9	-51	-21						
j-7	222	30	84	24	-6	-78					
j-6	287	43	149	89	7	-13	-11				
j-5	343	54	204	144	18	42	0	-36			
j-4	387	63	249	189	27	87	9	9	-21		
j-3	422	70	284	224	34	122	16	44	14	-2	
j-2	447	75	309	249	39	147	21	69	39	3	-3
j-1	462	78	324	264	42	162	24	84	54	6	12
j	467	79	329	269	43	167	25	89	59	7	17
j+1	462	78	324	264	42	162	24	84	54	6	12
j+2	447	75	309	249	39	147	21	69	39	3	-3
j+3	422	70	284	224	34	122	16	44	14	-2	
j+4	387	63	249	189	27	87	9	9	-21		
j+5	343	54	204	144	18	42	0	-36			
j+6	287	43	149	89	7	-13	-11				
j+7	222	30	84	24	-6	-78					
j+8	147	15	9	-51	-21						
j+9	62	-2	-76	-136							
j+10	-33	-21	-171								
j+11	-138	-42									
j+12	-253										
NORM	5175	805	3059	2261	323	1105	143	429	231	21	35

Table C.1: Savitzky Golay Filter — Convolution coefficients and normalization factors for cubic polynomial smoothing[1]

Bibliography

[1] A. Savitzky and M. J. Golay, *Smoothing and differentiation of data by simplified least squares procedures.* Analytical chemistry **36**, 1627 (1964).

[2] *How a kingfisher helped reshape japan's bullet train,* .

[3] D. Lentink, S. R. Jongerius, and N. L. Bradshaw, *The scalable design of flapping micro-air vehicles inspired by insect flight,* in *Flying insects and robots* (Springer, 2009) pp. 185–205.

[4] *Multiphysics,* .

[5] J. Degroote, K.-J. Bathe, and J. Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction,* Computers and Structures **87**, 793 (2009).

[6] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels, *Performance of partitioned procedures in fluid-structure interaction,* Computers and Structures **88**, 446 (2010).

[7] S. Chen and G. D. Doolen, *Lattice boltzmann method for fluid flows,* Annual Review of Fluid Mechanics **30**, 329 (1998), https://doi.org/10.1146/annurev.fluid.30.1.329 .

[8] C. Eliodoro, V. Karlin Ilya, N. Gorban Alexander, and B. Konstantinos, *Efficient simulations of detailed combustion fields via the lattice boltzmann method,* International Journal of Numerical Methods for Heat & Fluid Flow **21**, 494 (2011).

[9] *Continuum Mechanics,* Wikipedia.

[10] A. Spencer, *Continuum Mechanics*, Dover Books on Physics (Dover Publications, 2012).

[11] S. Turteltaub, *Continuum Mechanics Lecture Notes,* TU Delft (2019).

[12] G. Dhondt, *The Finite Element Method for Three-Dimensional Thermomechanical Applications* (2004).

[13] F. White, *Viscous Fluid Flow*, College Ie (McGraw-Hill, 2006).

[14] F. Blom, *A monolithical fluid-structure interaction algorithm applied to the piston problem,* Computer Methods in Applied Mechanics and Engineering **167**, 369 (1998).

[15] Q. Zhang and T. Hisada, *Analysis of fluid-structure interaction problems with structural buckling and large domain changes by ale finite element method,* Computer Methods in Applied Mechanics and Engineering **190**, 6341 (2001).

[16] 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**, 1 (2004).

[17] J. Hron and S. Turek, *A monolithic fem/multigrid solver for an ale formulation of fluid-structure interaction with applications in biomechanics,* Lecture Notes in Computational Science and Engineering **53**, 146 (2006).

[18] M. Gee, U. Küttler, and W. Wall, *Truly monolithic algebraic multigrid for fluid-structure interaction,* International Journal for Numerical Methods in Engineering **85**, 987 (2011).

[19] A. Van Zuijlen and H. Bijl, *Coarse level newton-krylov acceleration of sub-iterations in partitioned fluid-structure interaction,* (2011) pp. 814–825.

[20] S. Deparis, M. Discacciati, and A. Quarteroni, *A domain decomposition framework for fluid-structure interaction problems,* in *Computational Fluid Dynamics 2004*, edited by C. Groth and D. W. Zingg (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 41–58.

[21] S. Badia, A. Quaini, and A. Quarteroni, *Splitting methods based on algebraic factorization for fluid-structure interaction,* SIAM Journal on Scientific Computing **30**, 1778 (2008), https://doi.org/10.1137/070680497 .

[22] A. de Boer, A. van Zuijlen, and H. Bijl, *Review of coupling methods for non-matching meshes,* Computer Methods in Applied Mechanics and Engineering **196**, 1515 (2007).

[23] D. J. Rixen, H. Bijl, A. H. van Zuijlen, and A. de Boer, *Fluid structure interaction lectore notes,* TU Delft (2008).

[24] J. Donea, S. Giuliani, and J. Halleux, *An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions,* Computer Methods in Applied Mechanics and Engineering **33**, 689 (1982).

[25] C. Farhat, P. Geuzaine, and C. Grandmont, *The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids,* Journal of Computational Physics **174**, 669 (2001).

[26] C. Farhat and 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**, 499 (2000).

[27] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran, *Arbitrary lagrangian–eulerian methods,* in *Encyclopedia of Computational Mechanics* (American Cancer Society, 2004) Chap. 14, https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470091355.ecm009 .

[28] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges, K. Lee, A. Lott, Q. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. P. Randles, D. Reynolds, B. Rivière, U. Rüde, T. Scheibe, J. Shadid, B. Sheehan, M. Shephard, A. Siegel, B. Smith, X. Tang, C. Wilson, and B. Wohlmuth, *Multiphysics simulations: Challenges and opportunities,* The International Journal of High Performance Computing Applications **27**, 4 (2013), https://doi.org/10.1177/1094342012468181 .

[29] J. Degroote and J. Vierendeels, *Multi-level quasi-newton coupling algorithms for the partitioned simulation of fluid-structure interaction,* Computer Methods in Applied Mechanics and Engineering **225-228**, 14 (2012).

[30] C. Farhat, M. Lesoinne, and N. Maman, *Mixed explicit/implicit time integration of coupled aeroelastic problems: Three□field formulation, geometric conservation and distributed solution,* International Journal for Numerical Methods in Fluids **21**, 807 (1995).

[31] H. Matthies and J. Steindorf, *Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction,* Computers and Structures **80**, 1991 (2002).

[32] U. Küttler and W. Wall, *Fixed-point fluid-structure interaction solvers with dynamic relaxation,* Computational Mechanics **43**, 61 (2008).

[33] B. Irons and R. Tuck, *A version of the aitken accelerator for computer iteration,* International Journal for Numerical Methods in Engineering **1**, 275 (1969).

[34] C.Vuik and D. Lahaye, *Scientific computing lecture notes,* TU Delft (2017).

[35] D. Knoll and D. Keyes, *Jacobian-free newton-krylov methods: A survey of approaches and applications,* Journal of Computational Physics **193**, 357 (2004).

[36] C. Michler, E. van Brummelen, and R. de Borst, *An interface newton-krylov solver for fluid-structure interaction,* International Journal for Numerical Methods in Fluids **47**, 1189 (2005).

[37] J.-F. Gerbeau and M. Vidrascu, *A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows,* Mathematical Modelling and Numerical Analysis **37**, 631 (2003).

[38] R. Haelterman, A. Bogaers, K. Scheufele, B. Uekermann, and M. Mehl, *Improving the performance of the partitioned qn-ils procedure for fluid-structure interaction problems: Filtering,* Computers and Structures **171**, 9 (2016).

[39] J. Vierendeels, L. Lanoye, J. Degroote, and P. Verdonck, *Implicit coupling of partitioned fluid-structure interaction problems with reduced order models,* Computers and Structures **85**, 970 (2007).

[40] A. van Zuijlen, S. Bosscher, and H. Bijl, *Two level algorithms for partitioned fluid-structure interaction computations,* Computer Methods in Applied Mechanics and Engineering **196**, 1458 (2007).

[41] S. Piperno, C. Farhat, and B. Larrouturou, *Partitioned procedures for the transient solution of coupled aroelastic problems part i: Model problem, theory and two-dimensional application,* Computer Methods in Applied Mechanics and Engineering **124**, 79 (1995).

[42] E. Järvinen, P. Råback, M. Lyly, and J.-P. Salenius, *A method for partitioned fluid-structure interaction computation of flow in arteries,* Medical Engineering and Physics **30**, 917 (2008).

[43] P. Causin, J. Gerbeau, and F. Nobile, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems,* Computer Methods in Applied Mechanics and Engineering **194**, 4506 (2005).

[44] E. Van Brummelen, *Partitioned iterative solution methods for fluid-structure interaction,* International Journal for Numerical Methods in Fluids **65**, 3 (2011).

[45] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels, *Stability of a coupling technique for partitioned solvers in fsi applications,* Computers and Structures **86**, 2224 (2008).

[46] E. Van Brummelen and R. De Borst, *On the nonnormality of subiteration for a fluid-structure-interaction problem,* SIAM Journal on Scientific Computing **27**, 599 (2006).

[47] OpenCFD Ltd, *OpenFOAM,* https://www.openfoam.com/.

[48] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, *A tensorial approach to computational continuum mechanics using object-oriented techniques,* Computers in Physics **12**, 620 (1998), https://aip.scitation.org/doi/pdf/10.1063/1.168744 .

[49] G. Dhondt and K. Wittig, *Calculix,* http://www.calculix.de/.

[50] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, and B. Uekermann, *precice – a fully parallel library for multi-physics surface coupling,* Computers and Fluids **141**, 250 (2016).

[51] G. Chourdakis, *A general OpenFOAM adapter for the coupling library preCICE*, Master's thesis, Technische Universität München (2017).

[52] D. Risseeuw, *Fluid Structure Interaction Modelling of Flapping WIngs*, Master's thesis, Delft University of Technology (2019).

[53] L. C. Yau, *Conjugate Heat Transfer with the Multiphysics Coupling Library preCICE*, Master's thesis, Technische Universität München (2016).

[54] A. Rusch, *Extending SU^2 to fluid-structure interaction via preCICE*, Bachelor's thesis, Technische Universität München (2016).

[55] S. Turek and J. Hron, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow,* Lecture Notes in Computational Science and Engineering **53**, 371 (2006).

[56] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®*, Vol. 113 (Springer International Publishing, 2015).

[57] S. Patankar and D. Spalding, *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows,* International Journal of Heat and Mass Transfer **15**, 1787 (1972).

[58] J. Van Doormaal and G. Raithby, *Enhancements of the simple method for predicting incompressible fluid flows,* Numerical Heat Transfer **7**, 147 (1984).

[59] C. Maliska and G. Raithby, *Calculating three-dimensional fluid flows using nonorthogonal grids.* (Pineridge Press Ltd, 1983).

[60] R. Issa, *Solution of the implicitly discretised fluid flow equations by operator-splitting,* Journal of Computational Physics **62**, 40 (1986).

[61] C. Kelley and D. Keyes, *Convergence analysis of pseudo-transient continuation,* SIAM Journal on Numerical Analysis **35**, 508 (1998).

[62] R. Clough, *The Finite Element Method in Plane Stress Analysis* (American Society of Civil Engineers, 1960).

[63] C. Ashcraft and R. G. Grimes, *Spooles: An object-oriented sparse matrix library,* in *PPSC* (1999).

[64] B. W. Uekermann, *Partitioned Fluid-Structure Interactionon Massively Parallel Systems*, Ph.D. thesis, Technische Universität München (2016).

[65] *Acceleration Configuration – preCICE Wiki*.

[66] B. Gatzhammer, *Efficient and Flexible PartitionedSimulation of Fluid-Structure Interactions*, Ph.D. thesis, Technische Universität München (2014).

[67] *Mesh Exchange Configuration — preCICE Wiki*.

[68] *Actions Configuration — preCICE Wiki*.

[69] J. H. Ahlberg, E. N. Nilson, and J. L. . Walsh, *The theory of splines and their applications* (Academic Press, New York, NY, 1967).

[70] J. F. Epperson, *On the runge example,* The American Mathematical Monthly **94**, 329 (1987).

[71] P. Dierckx, *Curve and Surface Fitting with Splines*, Monographs on numerical analysis (Clarendon Press, 1995).

[72] K. Levenberg, *A method for the solution of certain non-linear problems in least squares,* Quarterly of Applied Mathematics **2**, 164 (1944).

[73] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters,* Journal of the Society for Industrial and Applied Mathematics **11**, 431 (1963).