Efficient numerical methods for partitioned fluid-structure interaction simulations

Blom, David

DOI
10.4233/uuid:4f467feb-7155-4465-9508-650f3f507847

Publication date
2017

Document Version
Final published version

Citation (APA)
Blom, D. (2017). *Efficient numerical methods for partitioned fluid-structure interaction simulations*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:4f467feb-7155-4465-9508-650f3f507847

Efficient numerical methods for partitioned fluid-structure interaction simulations

David Blom

Efficient numerical methods for partitioned fluid-structure interaction simulations

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 27 maart 2017 om 12:30 uur

door

David Stephen Blom

ingenieur luchtvaart en ruimtevaart
geboren te Bulawayo, Zimbabwe.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. ir. H. Bijl
copromotor: Dr. ir. A. H. van Zuijlen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. H. Bijl,	Delft University of Technology
Dr. ir. A. H. van Zuijlen,	Delft University of Technology
Onafhankelijke leden:	
Prof. dr. ir. C. Vuik,	Delft University of Technology
Prof. dr. M. Mehl,	Universität Stuttgart
Prof. dr. P. Birken,	Lund University
Prof. dr. ir. J. Vierendeels,	Universiteit Gent
Prof. dr. ir. B. Koren,	Eindhoven University of Technology
Prof. dr. ir. L. L. M. Veldhuis,	Delft University of Technology, reservelid

Keywords: fluid-structure interaction, manifold mapping, strongly coupled, spectral deferred correction, higher order time integration

I AM WHO I AM

Summary

Fluid-structure interaction simulations are crucial for many engineering problems. For example, the blood flow around new heart valves or the deployment of airbags during a car crash are often modeled with fluid-structure interaction simulations. Also, to design safe parachutes, simulations are carried out to model the unsteady deformations of the parachute during a jump. Thus, there is an apparent need for multi-physics software codes which can model fluid-structure interaction problems.

However, current state-of-the-art solvers cannot be used for design or optimization studies of for example aircraft structures due to long simulation times. This is mainly caused by a large number of coupling iterations needed to reach convergence within each time step for a strongly coupled fluid-structure interaction simulation. Also, a large number of time steps are required to reach an acceptable accuracy in time for unsteady simulations. Hence, there is an urgency for efficiency improvements of fluid-structure interaction solvers.

In this thesis, two approaches are investigated to decrease the computational times for a fluid-structure interaction simulation: multi-level acceleration of the coupled problem, and the use of higher order time integration schemes.

Part I. Multi-level acceleration with manifold mapping for partitioned fluid-structure interaction

Highly sophisticated software codes are available for each single physical phenomenon. It is desirable to reuse these codes for multi-physics simulations such as a fluid-structure interaction problem where separate fluid and solid solvers need to be coupled. This partitioned approach is in contrast to the monolithic approach where all governing equations are implemented in a single software package and solved as a large system of equations.

The focus of the first part of this thesis is to accelerate the convergence of strongly coupled fluid-structure interaction problems, where the fluid solver and the structure solver are considered as black boxes.

We apply manifold mapping to a partitioned fluid-structure interaction problem to decrease the number of sub-iterations of the high-fidelity fluid-structure interaction model and are the first to do so. Low-fidelity models or coarse models based on a coarse grid discretization are used, as well as coarse models based on a different governing equation for the fluid-structure interaction problem.

Numerical experiments showed the high potential of the coupling scheme, which can reduce the number of fine model evaluations of a partitioned fluid-structure interaction simulation by approximately 50 % by transferring most of the work to the coarse model. The convergence of the high-fidelity model is accelerated even further when information from previous time steps is reused. Hence, manifold mapping can be applied effectively to a partitioned fluid-structure interaction simulation, and its use results in a significant decrease in the number of high fidelity coupling iterations.

Part II. Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using spectral deferred corrections

The second part of this thesis focuses on the urgent need to increase the efficiency of unsteady simulations. Numerical simulations of unsteady phenomena, such as the flow over a wind turbine, or the blood flow through an artery, are computationally expensive due to a large number of time steps required to reach an acceptable accuracy.

We use the finite volume method with collocated arrangement of the flow variables to spatially discretize the incompressible Navier-Stokes equations. To dampen the checker-board pressure field which appears in the standard collocated grid, commonly the interpolation procedure proposed by Rhie and Chow is utilized for steady-state problems. However, when applied directly to the transient formulation, this method does not ensure time consistency due to pressure oscillations when decreasing the time step.

Higher order time integration scheme have a potential gain in efficiency

and accuracy in time. Hence, the primary goal of the second part of this thesis is to present a solution procedure which preserves theoretical orders of accuracy in time for spectral deferred corrections, integral deferred corrections, the Picard Integral Exponential Solver, and implicit Runge-Kutta methods when applied to incompressible flow problems and fluid-structure interaction problems discretized with the finite volume method on collocated grids.

The good accuracy and efficiency of the time integration methods have been demonstrated for several numerical examples. The main advantage of the deferred correction schemes over diagonally-implicit Runge-Kutta schemes is the fact that the necessary coefficients to reach a desired order of accuracy are easily obtained by using the quadrature integration rules. For moderate accuracy, the (E)SDIRK methods proved to be competitive to SDC and IDC.

Samenvatting

Vloeistof-structuur interactie simulaties zijn essentieel om vele technische problemen op te lossen. Bijvoorbeeld, de bloedstroming rond nieuwe hartkleppen of het gebruik van airbags bij een auto-ongeluk wordt vaak gemodelleerd met vloeistof-vaste stof interactie simulaties. Ook voor het ontwerpen van veilige parachutes is het noodzakelijk dat simulaties worden uitgevoerd om de tijdsafhankelijke vervorming van de parachute tijdens een sprong te kunnen voorspellen. Er is dus een grote behoefte aan software pakketten die vloeistof-vaste stof interactie problemen efficiënt kunnen simuleren.

Echter, de huidige generatie software kan niet worden gebruikt voor ontwerp en optimalisatie studies van bijvoorbeeld vliegtuigstructuren vanwege lange simulatietijden. Dit wordt onder andere veroorzaakt doordat een groot aantal koppelings iteraties nodig zijn om te convergeren tijdens elke tijd stap voor een sterk gekoppeld vloeistof-vaste stof interactie simulatie. Er is ook een groot aantal tijd stappen nodig om een aanvaardbare precisie te behalen in tijd voor tijdsafhankelijke simulaties. Vandaar dat er een dringende behoefte is om de efficiëntie van vloeistof-vaste stof interactie simulaties te verbeteren.

In dit proefschrift zijn twee benaderingen onderzocht om de efficiëntie te verbeteren van vloeistof-vaste stof interactie simulaties: multi-level acceleratie van het gekoppelde probleem en het gebruik van hogere orde tijd integratie methoden.

Deel I. Multi-level acceleratie met manifold mapping toegepast op gepartitioneerde vloeistof-vaste stof interactie simulaties

Zeer geavanceerde software codes zijn beschikbaar voor iedere afzonderlijke fysische verschijnsel. Het is wenselijk om deze codes te hergebruiken voor

multi-physics simulaties zoals een vloeistof-vaste stof interactie probleem waarbij aparte vloeistof en structuur oplossers met elkaar moeten worden gekoppeld. Deze gepartitioneerde benadering staat in contrast met de monolithische benadering waarbij alle vergelijkingen worden geïmplementeerd in een enkel softwarepakket en opgelost als een groot stelsel van vergelijkingen. De focus van het eerste deel van dit proefschrift is het versnellen van de convergentie van sterk gekoppelde vloeistof-vaste stof interactie problemen waarbij de software codes van de afzonderlijke pakketten niet beschikbaar zijn.

We gebruiken manifold mapping voor het eerst voor een vloeistof-vaste stof interactie probleem om het aantal sub-iteraties van het high-fidelity model te verminderen. Low-fidelity modellen of grove modellen gebaseerd op een grof rooster worden gebruikt, evenals grove modellen gebaseerd op een andere vergelijking voor het vloeistof-vaste stof interactie probleem.

Numerieke experimenten tonen het grote potentieel van de koppelings methode aan, welke het aantal fine model evaluaties van een vloeistof-vaste stof interactie simulatie met ongeveer 50 % kan verminderen doordat het meeste werk wordt uitgevoerd door het grove model. De convergentie van het high-fidelity model wordt nog verder versneld wanneer informatie van vorige tijdstappen wordt hergebruikt. Manifold mapping kan dus worden toegepast op een gepartitioneerde vloeistof-vaste stof interactie probleem, en het gebruik ervan resulteert in een aanzienlijke daling van het aantal high-fidelity koppelings iteraties.

Deel II. Willekeurig hoge orde tijd integratie voor gepartitioneerde vloeistof-vaste stof interactie simulaties aan de hand van spectral deferred corrections

Het tweede deel van dit proefschrift richt zich op de noodzaak voor betere efficiëntie van tijdsafhankelijke simulaties. Numerieke simulaties van tijdsafhankelijke verschijnselen, zoals de stroming rondom een windturbine, of de stroming van bloed door een slagader, duren erg lang wat te wijten is aan het feit dat een groot aantal tijdstappen nodig is om een aanvaardbare precisie te bereiken.

We gebruiken de eindige volume methode om de incompressibele Navier-Stokes vergelijkingen te discretizeren in de ruimte waarbij de vloeistof variabelen op een collocated rooster zijn gedefinieerd. Om de oscillaties in de druk door het gebruik van een collocated rooster te verminderen, wordt gewoonlijk de momentum interpolatie methode van Rhie en Chow gebruikt. Deze is echter ontworpen voor tijdsonafhankelijke problemen. Wanneer deze method echter direct op de tijdsafhankelijke vergelijkingen wordt toegepast, is er geen garantie dat de methode consistent is in tijd door oscillaties in de druk wanneer de tijd stap wordt verminderd.

Hogere orde tijd integratie methoden zijn in potentie zeer efficiënt en nauwkeurig in de tijd. Vandaar dat het primaire doel van het tweede deel van dit proefschrift is om een methode te presenteren, die de theoretische orde van nauwkeurigheid in tijd behoudt voor spectral deferred corrections, integral deferred corrections, de Picard Integral Exponential Solver en impliciete Runge-Kutta methoden toegepast op incompressible vloeistof problemen en vloeistof-vaste stof interactie problemen die zijn gediscretizeerd met de eindige volume methode.

De goede nauwkeurigheid en efficiëntie van de tijdsintegratie methoden is aangetoond met verschillende voorbeelden. Het belangrijkste voordeel van de deferred correction method ten opzichte van diagonaal-impliciete Runge-Kutta methoden is het feit dat de coëfficiënten die nodig zijn om tot een gewenste orde van nauwkeurigheid te komen eenvoudig te berekenen zijn via de integratie regels.

Contents

Summary vii

Samenvatting xi

1. Introduction 1
- 1.1. Multi-level acceleration with manifold mapping for partitioned fluid-structure interaction . . . 2
- 1.2. Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using spectral deferred corrections . . . 5
- 1.3. Outline . . . 7

I. Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction 9

2. Coupling schemes for partitioned fluid-structure interaction 11
- 2.1. Fluid-structure interaction . . . 12
- 2.2. Manifold mapping . . . 14
- 2.3. Coarse model optimization . . . 22
- 2.4. Convergence measures . . . 24
- 2.5. Coupling schemes for strongly coupled fluid-structure interaction . . . 25

3. Numerical results for different coupling schemes 31
- 3.1. Unsteady flow in a one-dimensional tube . . . 32
- 3.2. Fixed cylinder with an attached flexible flap . . . 39
- 3.3. Wave propagation in a three-dimensional elastic tube . . . 48
- 3.4. Steady state for fixed cylinder with an attached flexible flap case . . . 55
- 3.5. Unsteady flow in a one-dimensional tube with three levels . 57

3.6. Wave propagation in a three-dimensional elastic tube with three levels 59

4. Conclusions on multi-level acceleration techniques for strongly coupled fluid-structure interaction 63
4.1. Conclusions 63
4.2. Recommendations 65

II. Arbitrarily high-order time integration for partitioned fluid-structure interaction simulations using spectral deferred corrections 67

5. Higher order time integration schemes 69
5.1. Time integration 70
5.2. Spectral deferred corrections 70
5.3. Integral deferred corrections 75
5.4. Picard Integral Exponential Solver (PIES) 75
5.5. Implicit Runge-Kutta 80

6. Spatial and temporal discretization of the incompressible Navier-Stokes equations 83
6.1. Introduction 84
6.2. Static grids 84
6.3. Face-velocity interpolation 85
6.4. Solution algorithm 86
6.5. Moving grids 88
6.6. Higher order time integration 90
6.7. Discrete Geometric Conservation Law 91
6.8. Boundary conditions at the fluid-structure interface 92
6.9. Integral deferred corrections applied to fluid-structure interaction simulations 92

7. Numerical results for incompressible flows 95
7.1. Travelling waves 96
7.2. Lid-driven cavity flow 100

7.3. Lid-driven cavity flow with time dependent boundary conditions . . . 105
7.4. Flow around a cylinder . . . 108

8. Numerical results for partitioned fluid-structure interaction simulations using integral deferred corrections 115
8.1. Unsteady flow in a one-dimensional tube . . . 116
8.2. Lid-driven cavity flow with a moving mesh . . . 123
8.3. Three-dimensional flow over an elastic structure . . . 127

9. Conclusions and recommendations for arbitrarily high order time integration 133
9.1. Incompressible flow . . . 133
9.2. Partitioned fluid-structure interaction simulations . . . 135

A. Coefficients for the Picard Integral Exponential Solver 137

B. Butcher tableaus for the SDIRK methods 139

C. Butcher tableaus for the ESDIRK methods 143

Bibliography 151

Acknowledgements 165

List of publications 167

Curriculum Vitæ 171

Chapter 1.

Introduction

Physics is actually too hard for physicists.

(David Hilbert)

Wikipedia[1] gives a definition of multi-physics:

> Multiphysics treats simulations that involve multiple physical models or multiple simultaneous physical phenomena. For example, combining chemical kinetics and fluid mechanics or combining finite elements with molecular dynamics. Multiphysics typically involves solving coupled systems of partial differential equations.

Fluid-structure interaction (FSI) is an example where multiple physical models are coupled [10]. Examples where fluid-structure interaction is apparent are aero-elasticity [47], arterial flow [35] and airbag deployment [75]. Also, the deployment of parachute systems [94] can be modeled with fluid-structure interaction solvers.

However, current state-of-the-art solvers cannot be used for design or optimization studies of aircraft structures due to large simulation times. For example, an optimization study may require thousands of evaluation points with each requiring an unsteady fluid-structure interaction simulation. One simulation may already take four weeks to perform. Hence, the simulations need to be performed within a reasonable time frame, thus giving the need for an efficiency improvement of fluid-structure interaction solvers.

[1] `https://en.wikipedia.org/wiki/Multiphysics`

In this thesis, two approaches are investigated to decrease the computational times for a fluid-structure interaction simulation: multi-level acceleration of the coupled problem, and the use of higher order time integration schemes.

1.1. Multi-level acceleration with manifold mapping for partitioned fluid-structure interaction

Motivation Highly sophisticated software codes are available for each single physical phenomenon. It is desirable to reuse these codes for multi-physics simulations such as a fluid-structure interaction problem where separate fluid and solid solvers need to be coupled. This partitioned approach [48] is in contrast to the monolithic approach [7, 51, 95] where all governing equations are implemented in a single software package, and solved as a large system of equations. Besides the partitioned and monolithic approach, different mixed forms have been proposed such as a splitting method [3], which separates the fluid-structure system in a fluid velocity part and a pressure-structure part. The implementation of the discretization needs to be available with such an approach, which is generally not the case for commercial packages. This is also a drawback for methods which utilize the exact Jacobian [88].

A large number of coupling schemes which only consider input/output information of the fluid and solid solvers are already available in the literature, such as Aitken's method [70], vector extrapolation [71], Interface-GMRES(R) [77, 78], and the Interface Quasi-Newton Inverse Least Squares (IQN-ILS) technique [33, 35]. The IQN-ILS technique [33] is an efficient [32, 37] and robust black-box coupling algorithm for which convergence theorems are available in [56]. The IQN-ILS algorithm is mathematically equivalent to Anderson acceleration [2, 50, 97] which can be categorized as a multisecant method as discussed by Fang and Saad [46]. When applied to linear problems, it can be shown that Anderson acceleration is essentially equivalent to the GMRES method [97], which has also been shown for the IQN-ILS method [55]. An overview of several partitioned coupling techniques applicable to fluid-structure interaction can be

found in [3, 22].

The focus of this part is to accelerate the convergence of strongly coupled fluid-structure interaction problems, where the fluid solver and the structure solver are considered as black boxes. An implication of using only input and output information of the fluid and the structural solver is that a large number of sub-iterations is necessary to obtain a strongly coupled solution. This means that both the fluid and structure solver are called multiple times per time step. In case the coupling between the fluid flow and structural deformation is very strong (strongly coupled FSI), which is generally the case for large fluid/solid density ratios or due to the incompressibility of the fluid (see [21]), the convergence of the sub-iterations is hard to achieve. One idea is to perform a large amount of the sub-iterations with a low-fidelity model, instead of the current high-fidelity flow and structure models [90]. The question remains how to efficiently couple the multi-fidelity models.

Van Zuijlen and Bijl [101] developed a multi-level acceleration technique, which is based on the presumed existing geometric multigrid solver for the flow domain. Coarse and fine level sub-iterations are used alternately, resulting in substantial gains in computational costs. This technique hinges on the availability of a geometric multigrid solver for the fluid domain, which poses problems in case black-box solvers are considered.

In [23], [51], [60], and [87] the multi-grid solution technique is used in order to decrease the computational effort required to solve the fluid-structure interaction problem. Although multigrid methods can in principle be adapted in such a manner that the acceleration is performed only on the interface, current multigrid methods generally operate on the complete aggregated fluid-structure interaction problem. As opposed to the manifold-mapping technique presented in this work, which acts only on the fluid-structure interface, standard multigrid methods are therefore not applicable to black-box solvers, as these methods are inherently intrusive with respect to the source code of the fluid and structure solvers.

Degroote and Vierendeels [36] present the multi-level IQN-ILS algorithm which shows the potential of a multi-level acceleration framework for fluid-structure interaction where the fluid and structure solvers are considered as black boxes. A coarse grid is used as the low-fidelity model in order to build an approximate Jacobian which is reused by the finer grids to

accelerate the convergence of the IQN-ILS algorithm. However, the coarse grid is only used once per time step to built an approximate Jacobian, and find a better initial guess for the fine grid solution process. It can be expected that the performance of a coupling algorithm which uses a fine grid and coarse grid alternately results in less fine grid iterations as is the case for multi-grid.

Originating from multi-fidelity optimization, the aggressive space mapping algorithm has been used by Scholcz et al. [90] to efficiently couple a high-fidelity model with a low-fidelity model for a fluid-structure interaction problem. Scholcz et al. [90] consider the fluid-structure interaction interface problem as an optimization problem. A computationally inexpensive low-fidelity model is combined with a high-fidelity model in order to accelerate the convergence of the high-fidelity model.

Bandler et al. [6] proposed the original space mapping algorithm for modeling and design of engineering devices and systems. Thereafter, the aggressive space mapping approach [4] was introduced based on a quasi-Newton iteration, which utilizes each fine model iterate as soon as it is available. The basis of the space mapping algorithm is the parameter extraction step which establishes the mapping and updates the surrogate. However, a breakdown of the algorithm may occur due to non-uniqueness of the parameter extraction step [5]. Output space mapping [68] aims at reducing the misalignment between the coarse and fine models by adding the difference between the two to the response of the coarse model. An overview of different space mapping algorithms in given in [5]. When applied to a fluid-structure interaction problem as in this contribution, the different space mapping techniques consider the fluid and solid solvers as black-boxes, i.e. the used discretizations for the fluid and structural domains cannot be changed.

Echeverria et al. [43] introduce the manifold mapping algorithm as an alternative to the aforementioned space mapping algorithms. A notable difference between manifold mapping and aggressive space mapping is the fact that the algorithm has provable convergence to the solution of the high-fidelity model, under the assumption that the manifold mapping iterate converges and that the optimization problems are well defined as shown in [43]. However, for a fluid-structure interaction problem, it is difficult to show whether these conditions are met and are therefore not included

here. It is therefore considered as a hypothetical difference between the space mapping and manifold mapping technique.

Goal We apply manifold mapping for the first time to a partitioned fluid-structure interaction problem in order to decrease the number of sub-iterations of the high-fidelity fluid-structure interaction model. The fluid and solid solvers are considered as black-boxes. Low-fidelity models or coarse models based on a coarse grid discretization are used, as well as coarse models based on a different governing equation for the fluid-structure interaction problem. The research question is formulated as follows: can manifold mapping be effectively applied to a partitioned fluid-structure interaction simulation, and does its use result in a significant decrease in the number of high fidelity coupling iterations?

1.2. Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using spectral deferred corrections

Motivation Unsteady incompressible flows play an important role in many engineering and scientific problems. Examples include flow over a wind turbine [61], blood flow through an artery [11], and the interaction with a propeller of a ship [1]. Numerical simulations of such unsteady phenomena are computationally expensive due to a large number of time steps required to reach an acceptable accuracy. Therefore, there is an urgent need to increase the efficiency of these simulations.

The second order backward differencing scheme (BDF2) is commonly applied to discretize the unsteady terms of the Navier-Stokes equations [12]. The BDF2 scheme is unconditionally stable, which explains its popularity. One way to obtain a temporally accurate solution while maintaining computational cost at a reasonable level is to apply a high-order time integration scheme. However, third order and higher order BDF schemes are not unconditionally stable, which prohibits their wide use.

An alternative to BDF2 is the diagonally implicit Runge-Kutta (DIRK) method, which can be constructed at arbitrarily high order while retaining

good stability properties [67, 83]. But their derivation becomes increasingly difficult due to an increase in order conditions when increasing the order of the scheme [67]. Examples of the application of DIRK methods to incompressible flow are available in literature [65, 66, 81], and have shown to outperform BDF2 in terms of accuracy and computational costs.

A different approach to time integration is taken by Dutt et al. [40], who have introduced the spectral deferred correction scheme (SDC). High-order approximations are computed for a time step by solving a series of correction equations on a number of sub-steps. The corrections are applied iteratively to an initial solution defined on the sub-steps of a time step. Each correction iteration, or so-called correction sweep, increases the formal order of accuracy of the method [59, 79]. The sub-steps are chosen to correspond to quadrature nodes, and therefore the time integrals can be computed to a high order of accuracy.

Closely related to SDC methods is the integral deferred correction (IDC) technique. IDC methods have been developed to be able to use uniform sub-steps within one time step [29]. These methods still achieve a high order of accuracy. One notable advantage of implicit IDC methods is the fact that these methods are L-stable or unconditionally stable, which is not necessarily the case for spectral deferred correction schemes. Christlieb et al. [28] use the integral deferred correction method within an operator splitting framework to obtain higher order approximations of partial differential equations.

Besides these methods, the Picard Integral Exponential Solver (PIES) can be used to obtain a user specified accuracy of the solution. Also, the scheme possesses good stability properties, i.e. it is unconditionally stable in case of implicit time stepping [69]. High accuracy is obtained by assuming that the solution at the next time step can be represented by a combination of complex exponential functions.

In this thesis, the widely used finite volume method with collocated arrangement of the flow variables is used to spatially discretize the incompressible Navier-Stokes equations. In order to dampen the checker-board pressure field which appears in the standard collocated grid, commonly the interpolation procedure proposed by Rhie and Chow [85] is utilized for steady-state problems. However applied directly to the transient formulation this method does not ensure time consistency due to pressure

oscillations when decreasing the time step [91]. Yu et al. [99, 100] have further shown that other momentum interpolation schemes can also be time step and under relaxation factor dependent.

Goal Higher order time integration scheme have a potential gain in efficiency and accuracy in time. Hence, the main goal of the second part is to present a solution procedure which preserves theoretical orders of accuracy in time for SDC, IDC, and PIES when applied to incompressible flow problems and fluid-structure interaction problems discretized with the finite volume method on collocated grids. We show the expected order of accuracy is reached for both the velocity, as well as for the pressure.

1.3. Outline

This thesis is divided into two parts, and has the following structure:

Part I. Multi-level acceleration with manifold mapping for partitioned fluid-structure interaction

- Chapter 2 introduces several coupling methods including manifold mapping for black-box partitioned fluid-structure interaction simulations.
- Thereafter, the potential of manifold mapping is shown with several test case in Chap. 3.
- Chapter 4 summarizes the first part and gives recommendations for future research directions.

Part II. Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using spectral deferred corrections

- Chapter 5 introduces spectral deferred corrections, integral deferred corrections, the Picard Integral Exponential Solver, and diagonally implicit Runge-Kutta methods.

- Chapter 6 gives the solution procedures for incompressible flows on static grids and moving grids whilst preserving the theoretical order of convergence of the time integration scheme considered.
- The numerical results are split up into two separate chapters. First, the efficiency and accuracy of SDC, IDC and PIES are compared to implicit Runge-Kutta methods and second order backward differencing for incompressible flows in Chap. 7.
- Thereafter, several fluid-structure interaction examples are shown in Chap. 8.
- Chapter 9 gives conclusions and recommendations on the use of spectral deferred correction methods for incompressible flows and fluid-structure interaction.

Part I.

Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction

Chapter 2.

Coupling schemes for partitioned fluid-structure interaction

> Music can change the world because it can change people.
>
> *(Bono)*

As mentioned in the introduction, we use the partitioned approach instead of the monolithic approach for fluid-structure interaction simulations. Hence, having the advantage that we consider the fluid and solid solvers as black boxes, and are able to reuse existing software codes, whether they are licensed as open source or closed source. Different coupling algorithms exist to solve for the strongly coupled or monolithic solution. Here, we focus on the manifold mapping approach and shortly discuss other techniques.

The fluid-structure interaction problem is introduced in Sect. 2.1 utilizing a black-box notation. Section 2.2 introduces the manifold mapping algorithm, and after that, an overview is given of several other coupling schemes. The next chapter compares the performance with numerical results of the different coupling schemes.

Parts of this chapter have been published in

- D. S. Blom et al. "Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 296 (2015), pp. 211–231. DOI: 10.1016/j.cma.2015.08.004;
- D. S. Blom et al. "Multi-Level Acceleration of Parallel Coupled Partitioned Fluid-Structure Interaction with Manifold Mapping". In: *Lecture Notes in Computational Science and Engineering*. Vol. 105. Springer International Publishing, 2015, pp. 135–150. DOI: 10.1007/978-3-319-22997-3_8;

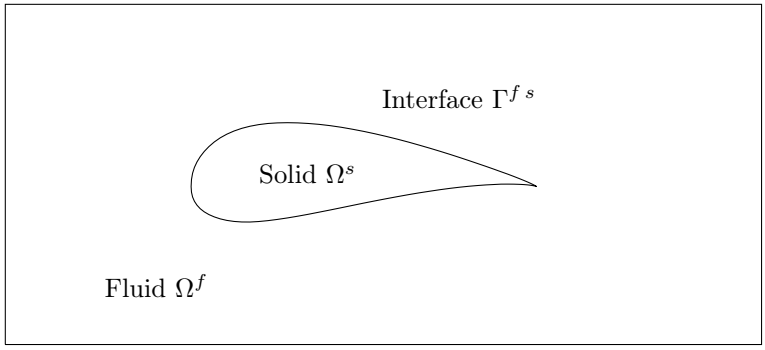


Figure 2.1.: Schematic of a fluid-structure interaction problem with a fluid domain Ω^f and solid domain Ω^s. The interface $\Gamma^{f\,s}$ is defined as the boundary between the two domains.

2.1. Fluid-structure interaction

The fluid-structure interaction problem is partitioned into a fluid domain Ω^f and a solid domain Ω^s, as shown in Fig. 2.1 The separate domains are coupled on the fluid-structure interface $\Gamma^{f\,s}$ through the interface conditions consisting of the kinematic and dynamic boundary conditions.

The fluid solver and solid solver are considered as black boxes. In other words, only the input and output information is accessible. Whether a compressible or incompressible, viscous or inviscid flow is considered does not influence the used methodology described in this part. Also, different models for the solid domain can be applied, since only the input and output information from the fluid solver and the solid solver is considered to be accessible.

Therefore, at each time step the response of the fluid solver F_f is defined as

$$\boldsymbol{y} = F_f\left(\boldsymbol{x}\right), \tag{2.1}$$

where $\boldsymbol{x}$ denotes the displacement of the fluid-structure interface, and $\boldsymbol{y}$ denotes the force acting on the fluid-structure interface. The response of

- D. S. Blom et al. "Acceleration of strongly coupled fluid-structure interaction with manifold mapping". In: *Proceedings of WCCM XI, Proceedings of ECCM V, Proceedings of ECFD VI* WCCM XI (2014), pp. 1–12.

the structure solver F_s is consequently defined as

$$\boldsymbol{x} = F_s\left(\boldsymbol{y}\right). \tag{2.2}$$

Note that the mentioned variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are defined on the fluid-structure interface. The fluid and solid solvers solve for the state variables in their internal domain, for example, velocity, pressure, and displacement. Generally, the number of degrees of freedom of the interface variables are much smaller compared to the number of degrees of freedom of the state variables, since the interface is defined on a smaller dimension. For example, for a three-dimensional flow around a wind turbine, the interface is the surface of the wind turbine.

For a sequential execution of both solvers, the fixed point equation

$$\boldsymbol{x} = F_s \circ F_f\left(\boldsymbol{x}\right) \tag{2.3}$$

must be satisfied at every time step. This can also be written as the interface residual $\boldsymbol{R}$

$$\boldsymbol{R}\left(\boldsymbol{x}\right) = F_s \circ F_f\left(\boldsymbol{x}\right) - \boldsymbol{x}, \tag{2.4}$$

which is solved with a minimization or optimization procedure aimed to find the optimal solution $\boldsymbol{x}^\star$ such that

$$\boldsymbol{x}^\star = \arg\min_{\boldsymbol{x}} \left|\left|F_s \circ F_f\left(\boldsymbol{x}\right) - \boldsymbol{x}\right|\right|_2. \tag{2.5}$$

For a parallel execution of both solvers, a different fixed point equation can be formulated:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_f\left(\boldsymbol{x}\right) \\ F_s\left(\boldsymbol{y}\right) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}. \tag{2.6}$$

When written as a residual, the following notation applies:

$$\boldsymbol{R}\left(\boldsymbol{x}, \boldsymbol{y}\right) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_f\left(\boldsymbol{x}\right) \\ F_s\left(\boldsymbol{y}\right) \end{bmatrix} - \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}. \tag{2.7}$$

For the sake of reuse of notation for both the sequential and parallel case, we also refer to the last definition of the residual as $\boldsymbol{R}\left(\boldsymbol{x}\right)$. In this case, $\boldsymbol{x}$ consists of both displacement and force values.

Standard approaches used to solve the strongly coupled fluid-structure

interaction problem are the Gauss-Seidel method [47], fixed under-relaxation [26], Aitken under-relaxation [70], and the IQN-ILS method [33, 36]. These minimization procedures require the evaluation of the interface residual multiple times per time step, which is considered to be expensive for high-fidelity models.

Algorithms originally developed to accelerate a high-fidelity optimization process by using a lower fidelity model, i.e. multi-fidelity optimization, such as aggressive space mapping [4] and output space mapping [68], can also be used to minimize the interface residual. The high-fidelity optimization process is accelerated by using a surrogate model which closely resembles the response of the high-fidelity model, but requires much less computational time. Naturally, the surrogate model is less accurate than the high-fidelity model, but it is corrected by the high-fidelity model during the optimization process. Scholcz et al. [90] proposed a framework for the application of the space mapping technique to partitioned fluid-structure interaction problems, to speed up the convergence of high-fidelity simulations. The application of the aggressive space mapping algorithm was demonstrated on various academic fluid-structure interaction problems.

Here, the use of manifold mapping [42] is investigated to solve the partitioned fluid-structure interaction problem. Manifold mapping has proven to be an efficient algorithm resulting in less high-fidelity iterations compared to aggressive space mapping [42] and output space mapping [38].

2.2. Manifold mapping

Manifold mapping is a surrogate-based optimization technique, which means that the quality of the initial solution or approximation of the low-fidelity model is iteratively improved. The goal of a surrogate-based optimization technique is to decrease the computational time of the optimization process of the high fidelity model by combining it with a low fidelity model. Here, a fluid-structure interaction simulation is considered for which the coupling represents the optimization problem. Refer to Eqn. (2.5) for the staggered case, and to Eqn. (2.7) for the residual of the parallel case.

In the following subsection, the basic terminology is introduced, and the manifold mapping algorithm is shown. The reader is referred to Echeverria et al. [43] for the theoretical basis of the technique.

2.2.1. Manifold mapping terminology

Two types of models are distinguished: a fine model and a coarse model. It is assumed that the fine model is accurate, but requires a high computational cost to evaluate. On the contrary, the coarse model is considered to be computationally less costly, but also less accurate in comparison to the fine model.

Fine model

The fine model response is denoted by $\boldsymbol{f} : X \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$, where $\boldsymbol{x} \in X$ represents the control variable of the optimization problem. The fine model cost function is defined as $\mathcal{F}(\boldsymbol{x}) = ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{q}||$, which represents the discrepancy between the design specification $\boldsymbol{q} \in \mathbb{R}^n$ and a particular response of the model $\boldsymbol{f}(\boldsymbol{x})$. Therefore, a minimization problem needs to be solved:

$$\boldsymbol{x}_f{}^{\star} = \arg\min_{\boldsymbol{x} \in X} ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{q}||. \tag{2.8}$$

When the manifold mapping technique is applied to the fluid-structure interaction problem, the fine model response is defined as the interface residual:

$$\boldsymbol{f}(\boldsymbol{x}) = F_s \circ F_f(\boldsymbol{x}) - \boldsymbol{x}. \tag{2.9}$$

The design specification $\boldsymbol{q}$ is the zero vector for the fine model. For consistency, $\boldsymbol{q}$ is not dropped from the formulation, since the design specification of the coarse model is not the zero vector, but updated at each manifold mapping iteration.

Coarse model

The coarse model response is denoted by $\boldsymbol{c}^z : Z \subset \mathbb{R}^m \rightarrow \mathbb{R}^m$, where $\boldsymbol{z} \in Z$ represents the control variable of the coarse model:

$$\boldsymbol{c}^z(\boldsymbol{z}) = C_s \circ C_f(\boldsymbol{z}) - \boldsymbol{z}. \tag{2.10}$$

C_f and C_s represent the coarse model fluid and solid operators. Similar to the fine model cost function, the coarse model cost function is defined as $\mathcal{C}(\boldsymbol{x}) = \left|\left|\boldsymbol{p}^{-1}(\boldsymbol{c}^z(\boldsymbol{p}(\boldsymbol{x}))) - \boldsymbol{q}_k\right|\right|$, with the mapping $\boldsymbol{p} : X \rightarrow Z$, and the inverse of the mapping $\boldsymbol{p}^{-1} : Z \rightarrow X$. $\boldsymbol{p}$ can be interpreted as a mapping

from the fine model design space to the coarse model design space. The design specification $\boldsymbol{q}_k \in \mathbb{R}^n$ of the coarse model is iteratively updated during the minimization process of the fine model. Subscript k represents the iteration counter of the manifold mapping algorithm. Therefore, the optimization of the surrogate model, which uses the coarse model, is defined as

$$\boldsymbol{x}_c^\star = \arg\min_{\boldsymbol{x} \in X} \left|\left|\boldsymbol{p}^{-1}\left(\boldsymbol{c}^z\left(\boldsymbol{p}\left(\boldsymbol{x}\right)\right)\right) - \boldsymbol{q}_k\right|\right|. \tag{2.11}$$

The optimization procedure of the coarse model is performed on the so-called coupling grid, as is also the case for the multi-level IQN-ILS algorithm [36]. The coupling grid has the same dimension as the dimension of the design space X and can represent the fine model responses exactly. Hence, the coarse model response $\boldsymbol{c}^z$ is interpolated to the coupling grid at every coarse model evaluation with the mapping $\boldsymbol{p}^{-1}$, and the control variable $\boldsymbol{x}$ is also interpolated from the coupling grid to the coarse model interface at every coarse model evaluation with the mapping $\boldsymbol{p}$. In case a coarse grid is used for the coarse model, then the mappings $\boldsymbol{p}$ and $\boldsymbol{p}^{-1}$ represent restriction and interpolation operators on the fluid-structure interface. Radial basis function interpolation is used for the mapping $\boldsymbol{p}$ between the fine model design space X and the coarse model design space Z, and for the mapping $\boldsymbol{p}^{-1}$ as well. Figure 2.2 shows the concept of the coupling grid. Only the fluid model is shown in the figure.

In the remainder of this work, the coarse response is denoted by $\boldsymbol{c}$ defined in X

$$\boldsymbol{c}\left(\boldsymbol{x}\right) = \boldsymbol{p}^{-1}\left(\boldsymbol{c}^z\left(\boldsymbol{p}\left(\boldsymbol{x}\right)\right)\right). \tag{2.12}$$

This means, the mapping $\boldsymbol{p}$ and the inverse mapping $\boldsymbol{p}^{-1}$ are implicitly assumed for simplicity, and are not included in the formulations.

Standard approaches for solving the partitioned fluid-structure interaction problem as the Gauss-Seidel method [47], fixed under-relaxation [26], Aitken under-relaxation [70], and the IQN-ILS method or Anderson acceleration [33, 36] can readily be applied to the coarse model optimization problem in case the design specification $\boldsymbol{q}_k$ is included in the interface residual. Section 2.3 describes the used algorithm to solve the coarse model optimization problem.

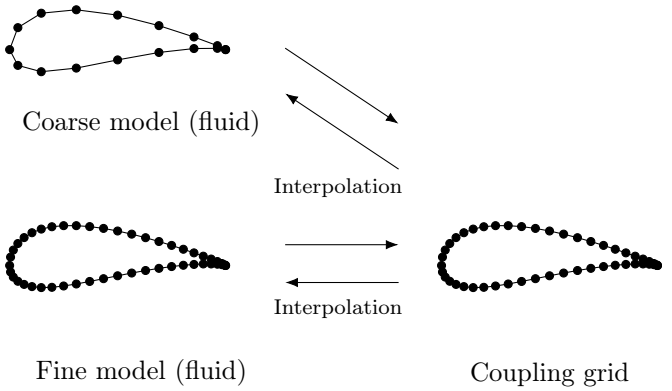


Figure 2.2.: Schematic of the coupling grid used for the interpolation between the fine model and coarse model of the fluid and solid solvers. Only the fluid part is shown here. Note that the interface only consists of the surface of the airfoil in this case.

2.2.2. Manifold mapping algorithm

The manifold mapping algorithm, as proposed in [41], introduces the mapping $\boldsymbol{S} : \boldsymbol{c}(X) \rightarrow \boldsymbol{f}(X)$ with the goal to correct for the misalignment between the fine and coarse model. With the mapping $\boldsymbol{S}$, the response $\boldsymbol{c}\left(\boldsymbol{x}_f^\star\right)$ is mapped to $\boldsymbol{f}\left(\boldsymbol{x}_f^\star\right)$, and the tangent plane for $\boldsymbol{c}(X)$ at $\boldsymbol{c}\left(\boldsymbol{x}_f^\star\right)$ is mapped to the tangent plane for $\boldsymbol{f}(X)$ at $\boldsymbol{f}\left(\boldsymbol{x}_f^\star\right)$. Figure 2.3 shows a sketch of the rotated and translated coarse model.

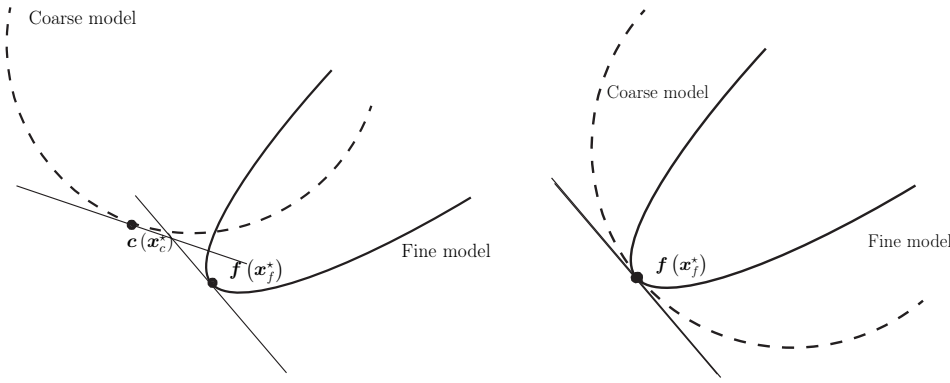
$\boldsymbol{S}$ is defined as the affine mapping

$$\boldsymbol{S}(\boldsymbol{c}(\boldsymbol{x})) = \boldsymbol{f}(\boldsymbol{x}_f{}^\star) + \overline{S}(\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{c}(\boldsymbol{x}_f{}^\star)) \tag{2.13}$$

for an unknown solution $\boldsymbol{x}_f{}^\star$ and with $\overline{S}$ specified as

$$\overline{S} = J_f(\boldsymbol{x}_f) J_c{}^\dagger(\boldsymbol{x}_f). \tag{2.14}$$

The Jacobian of the fine model J_f and the Jacobian of the coarse model J_c are, however, assumed to be unavailable for a black-box fluid-structure interaction problem and will be approximated. The symbol † indicates the pseudo-inverse. Now, each manifold mapping iterate is defined by a coarse



(a) Original models

(b) Rotated and translated coarse model

Figure 2.3.: Manifold mapping alignment of the coarse model using rotation and translation

model optimization:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}\in X} \left|\left| \boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{c}(\boldsymbol{x}_k) + \overline{S}^{\dagger} \left(\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{q} \right) \right|\right|, \tag{2.15}$$

where the pseudo-inverse of the manifold mapping function $\overline{S}^{\dagger}$ is approximated by a sequence T_k, resulting in

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}_k|| \tag{2.16}$$

with

$$\boldsymbol{q}_k = \boldsymbol{c}(\boldsymbol{x}_k) - T_k \left(\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{q} \right). \tag{2.17}$$

The approximation of the pseudo-inverse of the manifold mapping function T_k requires the singular value decomposition of the matrices ΔC_k and ΔF_k at each iteration k. The columns of ΔC and ΔF span the coarse and fine model tangent space at the current iteration as in

$$\Delta F_k = \left[\boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}(\boldsymbol{x}_k), \cdots, \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}\left(\boldsymbol{x}_{\max(k+1-n,0)}\right) \right] \tag{2.18}$$

and

$$\Delta C_k = \left[\boldsymbol{c}\left(\boldsymbol{x}_{k+1}\right) - \boldsymbol{c}\left(\boldsymbol{x}_k\right), \cdots, \boldsymbol{c}\left(\boldsymbol{x}_{k+1}\right) - \boldsymbol{c}\left(\boldsymbol{x}_{\max(k+1-n,0)}\right) \right]. \quad (2.19)$$

The singular value decomposition's of ΔC and ΔF are defined as:

$$\Delta C_k = U_c \, \Sigma_c \, V_c^T \quad \text{and} \quad \Delta F_k = U_f \, \Sigma_f \, V_f^T. \quad (2.20)$$

The design specification $\boldsymbol{q}_k$ is updated at each iteration k with T_k given by [41]

$$T_k = \Delta C_k \, {\Delta F_k}^{\dagger} + \left(I - U_c \, {U_c}^T\right)\left(I - U_f \, {U_f}^T\right). \quad (2.21)$$

The manifold mapping procedure is started with an extrapolation step of the solutions of the previous time steps in order to initialize the algorithm with a good initial guess. The resulting algorithm is shown in Fig. 2.4.

In Echeverria et al. [43] it is noted that it can be shown that the manifold mapping algorithm can be based on any matrix $\overline{S}$ satisfying $J_c^{\dagger}\left(\boldsymbol{x}_f^{\star}\right)\overline{S}^{\dagger} = J_f^{\dagger}\left(\boldsymbol{x}_f^{\star}\right)$. There is complete freedom how $\overline{S}$ handles components in the complement of the range of U_c. The general case is $\overline{S}^{\dagger} = J_c\left(\boldsymbol{x}_f^{\star}\right) J_f^{\dagger}\left(\boldsymbol{x}_f^{\star}\right) + \left(I - U_c \, U_c^T\right) A$ with A any $m \times m$ matrix. The particular case $A = \left(I - U_f \, U_f^T\right)$ is considered here such that components in the complement of the range of the left singular vectors U_f are included, as used before in Echeverria and Hemker [41].

Generally, the number of degrees of freedom n on the fluid-structure interface is much larger than the number of columns of ΔC_k and ΔF_k. Therefore, the computational cost of the manifold mapping technique is limited. It mainly consists of the singular value decomposition of two $n \times k$ matrices. To reduce the number of fine model evaluations per time step, the matrices ΔC_k and ΔF_k can be combined with the information from r previous time steps, following the IQN-ILS(r) approach [33]:

$$\Delta F_k = \left[\Delta F \;\; \Delta F^u \;\; \ldots \;\; \Delta F^{u-r+2} \;\; \Delta F^{u-r+1}\right] \quad (2.22)$$

$$\Delta C_k = \left[\Delta C \;\; \Delta C^u \;\; \ldots \;\; \Delta C^{u-r+2} \;\; \Delta C^{u-r+1}\right], \quad (2.23)$$

with $u+1$ marking the recent time step. The notation MM(r) indicates that information from r time steps is reused. When information from

1: $\boldsymbol{x}_0 = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}||$
2: $T_0 = I$
3: **for** $k = 0 \rightarrow k_{\max}$ **do**
4: $\quad \boldsymbol{q}_k = \boldsymbol{c}(\boldsymbol{x}_k) - T_k(\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{q})$
5: $\quad \boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}_k||$
6: $\quad \Delta F = \left[\boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}(\boldsymbol{x}_k), \cdots, \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}\left(\boldsymbol{x}_{\max(k+1-n,0)}\right)\right]$
7: $\quad \Delta C = \left[\boldsymbol{c}(\boldsymbol{x}_{k+1}) - \boldsymbol{c}(\boldsymbol{x}_k), \cdots, \boldsymbol{c}(\boldsymbol{x}_{k+1}) - \boldsymbol{c}\left(\boldsymbol{x}_{\max(k+1-n,0)}\right)\right]$
8: $\quad$ Solve $U_f \Sigma_f {V_f}^T = \Delta F$ with a singular value decomposition
9: $\quad$ Solve $U_c \Sigma_c {V_c}^T = \Delta C$ with a singular value decomposition
10: $\quad \Delta F^\dagger = V_f {\Sigma_f}^\dagger {U_f}^T$
11: $\quad T_{k+1} = \Delta C \Delta F^\dagger + \left(I - U_c {U_c}^T\right)\left(I - U_f {U_f}^T\right)$
12: $\quad$ **if** converged **then**
13: $\quad\quad$ **break**
14: $\quad$ **end if**
15: **end for**

Figure 2.4.: The manifold mapping (MM) algorithm

previous time steps is reused, the mapping matrix T_k does not need to be initialized with the identity matrix at line 2, but can be determined with Eqn. (2.21) (after the first time step).

The coarse optimization problem can be solved with a coupling scheme of the users choice. Here, Anderson acceleration is employed to solve the coarse optimization problem. Note that the coupling scheme needs to meet the design specification $\boldsymbol{q}_k$, which can be included in the formulation of the residual for the FSI problem.

The mapping matrix T_k is of size $n \times n$ which can be prohibitively large for large scale applications. To reduce the memory requirements of the algorithm, the iteratively updated design specification $\boldsymbol{q}_k$ can be directly determined with only matrix-vector multiplications such that the memory requirements do not exceed $n \times k$:

$$\begin{aligned} \boldsymbol{q}_k &= \boldsymbol{c}(\boldsymbol{x}_k) - T_k(\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{q}) \\ &= \boldsymbol{c}(\boldsymbol{x}_k) - \boldsymbol{\alpha} - \Delta C\left(\Delta F^\dagger \boldsymbol{\alpha}\right) + U_c\left[U_c^T(\boldsymbol{\alpha} - \boldsymbol{\beta})\right] + \boldsymbol{\beta}, \end{aligned} \tag{2.24}$$

with $\boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{q}$, and $\boldsymbol{\beta} = U_f\left(U_f^T \boldsymbol{\alpha}\right)$.

However, in case the simulation environment allows to store the mapping matrix T_k in memory, a secant update similar to the approach taken by Bogaers et al. [19] can be used:

$$T_k^{u+1} = T_k^u + (\Delta C - T_k^u \Delta F) \Delta F^\dagger, \tag{2.25}$$

where the pseudo inverse of ΔF can be computed as $\Delta F^\dagger = \left(\Delta F^T \Delta F\right)^{-1} \Delta F^T$ or via a singular value decomposition in the same manner as used by the original manifold mapping algorithm. The advantage of this approach is that the user does not need to specify the number of time steps the coupling scheme needs to reuse.

If the fluid and structure solvers are executed in serial, a separate synchronization step is necessary. Once the solution has been found by the fine model, the degrees of freedom of the coarse model need to be corrected [36]. Two different approaches can be applied. One approach is to restrict the data in the entire fluid and structure domain from the fine model to the coarse model. If such a system is not available since the used solvers are black box solvers, the interface traction calculated by the fine model can be applied to the fluid-structure interface of the coarse model, whereafter the structural equations are solved by the coarse model.

When using a mathematical notation, omitting the mappings, we simply have the evaluation of:

$$\boldsymbol{x}_c = C_s \left(\boldsymbol{y}_{k+1}\right). \tag{2.26}$$

Note that this approach can result in a difference between the solution of the flow and structure domain after a certain number of time steps.

During the manifold mapping iterations, the same interface displacement is applied on the fine model and also on the coarse model. Therefore, only the coarse structural model needs to be synchronized with the fine model if the second synchronization approach is used. If the fine model and the coarse model use exactly the same structural model, the second approach will result in a perfectly synchronized coarse model.

However, in case the fluid and structure solvers are executed in parallel, the same interface displacement and interface traction are already applied on the fine model as well as the coarse model. Therefore, a separate synchronization step is not necessary if the data in the entire fluid and structure domain is not restricted from the fine model to the coarse

model.

2.3. Coarse model optimization

Anderson acceleration [2, 50, 97] is used to solve the optimization problem of the coarse model (2.16). Anderson acceleration is mathematically equivalent to the IQN-ILS technique [33]. The algorithm is adapted to solve for the design specification $\boldsymbol{q}_k$. The technique is a quasi-Newton method with the aim to minimize the fluid-structure interface residual. To meet the design specification $\boldsymbol{q}_k$, Newton-Raphson iterations are used with index j:

$$\frac{\partial \boldsymbol{c}\left(\boldsymbol{x}_j\right)}{\partial \boldsymbol{x}} \Delta \boldsymbol{x}_j = \boldsymbol{q}_k - \boldsymbol{c}\left(\boldsymbol{x}_j\right) \tag{2.27}$$

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \Delta \boldsymbol{x}_j. \tag{2.28}$$

The desired value of the residual is the design specification $\boldsymbol{q}_k$, and the difference between the desired value and the current value $\Delta \boldsymbol{c} = \boldsymbol{q}_k - \boldsymbol{c}\left(\boldsymbol{x}_j\right)$ is approximated as a linear combination of the known $\Delta \boldsymbol{c}_i = \boldsymbol{c}\left(\boldsymbol{x}_i\right) - \boldsymbol{c}\left(\boldsymbol{x}_j\right)$:

$$\Delta \boldsymbol{c} \approx \sum_{i=0}^{j-1} \alpha_j^i \Delta \boldsymbol{c}_i = V_j \, \boldsymbol{\alpha}_j, \tag{2.29}$$

where V_j is given as

$$V_j = \left[\Delta \boldsymbol{c}_{j-1} \ldots \Delta \boldsymbol{c}_0\right]. \tag{2.30}$$

This is an over-determined problem, and is therefore solved in a least squares sense. The QR-decomposition of V_j is calculated by using Householder transformations:

$$V_j = Q_j R_j. \tag{2.31}$$

The coefficient vector $\boldsymbol{\alpha}_j$ is found by solving the triangular system

$$R_j \boldsymbol{\alpha}_j = {Q_j}^T \left(\boldsymbol{q}_k - \boldsymbol{c}\left(\boldsymbol{x}_j\right)\right) \tag{2.32}$$

for $\boldsymbol{\alpha}_j$ by using back substitution.

The residual of the least squares solution $\delta \Delta \boldsymbol{c}$ is defined as:

$$\delta \Delta \boldsymbol{c} = \Delta \boldsymbol{c} - V_j \, \boldsymbol{\alpha_j} = \boldsymbol{q}_k - \boldsymbol{c}\left(\boldsymbol{x}_j\right) - V_j \, \boldsymbol{\alpha_j}. \tag{2.33}$$

$\tilde{\boldsymbol{x}}_0 = F_s \circ F_f(\boldsymbol{x}_0)$ and $\boldsymbol{c}_0 = \tilde{\boldsymbol{x}}_0 - \boldsymbol{x}_0$
$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \omega(\boldsymbol{c}_0 - \boldsymbol{q}_k)$
for $j = 1 \rightarrow j_{\max}$ **do**
    $\tilde{\boldsymbol{x}}_j = F_s \circ F_f(\boldsymbol{x}_j)$ and $\boldsymbol{c}_j = \tilde{\boldsymbol{x}}_j - \boldsymbol{x}_j$
    $V_j = [\Delta\boldsymbol{c}_{j-1} \dots \Delta\boldsymbol{c}_0]$ with $\Delta\boldsymbol{c}_i = \boldsymbol{c}_i - \boldsymbol{c}_j$
    $W_j = [\Delta\boldsymbol{x}_{j-1} \dots \Delta\boldsymbol{x}_0]$ with $\Delta\boldsymbol{x}_i = \boldsymbol{x}_i - \boldsymbol{x}_j$
    $V_j = Q_j R_j$
    Solve $R_j \boldsymbol{\alpha}_j = {Q_j}^T (\boldsymbol{q}_k - \boldsymbol{c}_j)$ for $\boldsymbol{\alpha}_j$ using back substitution
    $\Delta\boldsymbol{x}_j = \beta(\boldsymbol{c}_j - \boldsymbol{q}_k) + (\beta V_j + W_j)\boldsymbol{\alpha}_j$
    $\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \Delta\boldsymbol{x}_j$
    if converged **then**
        break
    end if
end for

Figure 2.5.: Anderson acceleration (equivalent to the IQN-ILS method)

The $\Delta\boldsymbol{x}$ that corresponds to $\Delta\boldsymbol{c}$ is calculated as a linear combination of the previous $\Delta\boldsymbol{x}_j$, and is denoted with $\Delta\tilde{\boldsymbol{x}}_j$:

$$\Delta\tilde{\boldsymbol{x}}_j = \sum_{i=0}^{k-1} \alpha_j^i \Delta\boldsymbol{x}_i = W_j \boldsymbol{\alpha}_j, \tag{2.34}$$

where $\Delta\boldsymbol{x}$ is specified as $\Delta\boldsymbol{x}_i = \boldsymbol{x}_i - \boldsymbol{x}_j$. W_j is determined with

$$W_j = [\Delta\boldsymbol{x}_{j-1} \dots \Delta\boldsymbol{x}_0]. \tag{2.35}$$

This results in the relation

$$\begin{aligned} \Delta\boldsymbol{x}_j &= \Delta\tilde{\boldsymbol{x}}_j - \delta\Delta\boldsymbol{c} \\ &= W_j\boldsymbol{\alpha}_j - \boldsymbol{q}_k + \boldsymbol{c}(\boldsymbol{x}_j) + V_j\,\boldsymbol{\alpha_j}. \end{aligned} \tag{2.36}$$

Optionally, under relaxation can be applied on $\delta\Delta\boldsymbol{c}$, resulting in the following update:

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \beta(\boldsymbol{c}(\boldsymbol{x}_j) - \boldsymbol{q}_k) + (\beta V_j + W_j)\boldsymbol{\alpha}_j, \tag{2.37}$$

with the relaxation parameter β. The final algorithm is shown in Fig. 2.5.

2.4. Convergence measures

To terminate the fine model iterations, a relative convergence criterion based on input and output information is used with a tolerance τ_f:

$$\frac{||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{q}||_2}{||\boldsymbol{x} + \boldsymbol{q}||_2 + \varepsilon} \leq \tau_f, \tag{2.38}$$

with $\varepsilon = \sqrt{2^{-52}} = 2^{-26}$ when computing with double precision floats. For the results shown in this part, the relatively strict convergence criterion $\tau_f = 10^{-5}$ is used for the fine model. The convergence measure of the coarse model is based on input and output information of the coarse model and the iteratively updated design specification $\boldsymbol{q}_k$:

$$\frac{||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}_k||_2}{||\boldsymbol{x} + \boldsymbol{q}_k||_2 + \varepsilon} \leq \tau_c. \tag{2.39}$$

In order to ensure convergence of the manifold mapping iteration, the tolerance parameter τ_c of the coarse model is set to a stricter value $\tau_c = 10^{-6}$. Preliminary computations indicated that in case the convergence measure for the coarse model was set to the same tolerance as the fine model or even less strict, the iterations did not converge to the specified tolerance for the fine model.

A different convergence measure for the coarse model would be to monitor the convergence with respect to the first coarse model evaluation during the optimization step as follows:

$$\frac{||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}_k||_2}{||\boldsymbol{c}(\boldsymbol{x}_k) - \boldsymbol{q}_k||_2 + \epsilon} \leq \tau_c. \tag{2.40}$$

This measure can be named a relative residual convergence measure, as the first residual during an optimization step is used to scale the current residual. This approach results in less coarse model iterations, but does not influence the number of fine model iterations significantly. During the first manifold mapping iterations, the termination criterion is less strict compared to Eqn. (2.39), effectively reducing the number of coarse model iterations. The convergence measure becomes more strict, as the manifold mapping iteration process converges to the fine model optimum. Therefore,

a tolerance value of $\tau_c = 10^{-2}$ or $\tau_c = 10^{-3}$ is sufficient to reach convergence for the fine model, in contrast to the relative convergence measure shown in Eqn. (2.39).

2.5. Coupling schemes for strongly coupled fluid-structure interaction

The manifold mapping algorithm is compared to the IQN-ILS algorithm, multi-level IQN-ILS, aggressive space mapping, and output space mapping with several test cases. These techniques are shortly discussed in the following subsections.

2.5.1. IQN-ILS

The IQN-ILS algorithm, as discussed in Sect. 2.3, operates only on one level. Hence, this approach is not a multi-level technique. The difference with the algorithm as discussed in Sect. 2.3 is the fact that the design specification $\boldsymbol{q}_k$ is the zero vector. The interface residual is minimized directly with the fine model, instead of utilizing a multi-level approach.

2.5.2. ML-IQN-ILS

The multi-level quasi-Newton coupling algorithm [36] is based upon the IQN-ILS algorithm. The main difference is the use of coarse grids to accelerate the convergence of the finest level. The ML-IQN-ILS technique constructs the approximation of the inverse of the Jacobian on coarser grid levels, and then iteratively improves the Jacobian on the finest grid level.

First, the interface residual is minimized on the coarsest grid level. Then, starting from the current solution, the residual is minimized again on a finer grid reusing the Jacobian information generated on the coarser grid level. The algorithm is shown in Fig. 2.6.

The displacement and residual are defined on the so-called coupling grid. At each iteration on every grid level, the fluid and solid solvers interpolate the data from the surface boundary of their grid to the coupling grid.

When compared to manifold mapping or other space mapping algorithms, the multi-level IQN-ILS technique does not return to the coarse

$l = k = 0$
$\tilde{\boldsymbol{x}}_0 = F_s \circ F_f\left(\boldsymbol{x}_0\right)$ and $\boldsymbol{R}_0 = \tilde{\boldsymbol{x}}_0 - \boldsymbol{x}_0$
$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \omega \boldsymbol{R}_0$
for $i = 1 \rightarrow g$ **do**
    while $||\boldsymbol{R}_k|| > \tau$ or $l = 0$ **do**
        $l = 0$
        $\tilde{\boldsymbol{x}}_k = F_s \circ F_f\left(\boldsymbol{x}_k\right)$ and $\boldsymbol{R}_k = \tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k$
        Construct V and W
        Calculate the QR-decomposition $V_k = Q_k R_k$
        Solve $\boldsymbol{R}_k \boldsymbol{\alpha}_k = -{Q_k}^T \boldsymbol{R}_k$ for $\boldsymbol{\alpha}_k$ using back substitution
        $\Delta \boldsymbol{x}_k = \boldsymbol{R}_k + \left(V_k + W_k\right) \boldsymbol{\alpha}_k$
        $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta \boldsymbol{x}_k$
        $k = k + 1$
    end while
    $l = 1$
end for

Figure 2.6.: The ML-IQN-ILS method. g is the number of grid levels.

level after a fine level evaluation. Therefore, the multi-level IQN-ILS performs fewer iterations on the coarse levels compared to manifold mapping, aggressive space mapping, and output space mapping.

2.5.3. Aggressive space mapping (ASM)

Similar to manifold mapping, the aggressive space mapping technique [4] performs a minimization of the coarse model after each fine model iteration. The algorithm is based upon a quasi-Newton method in combination with first-order derivative approximations which are updated by the Broyden formula. Another approach would be to apply the IQN-ILS formulation as in the ASM-ILS technique [90].

The fine model response $\boldsymbol{f}\left(\boldsymbol{x}\right)$ needs to be minimized at every time step of the simulation. The space mapping technique introduces the space mapping function such that

$$\boldsymbol{f}\left(\boldsymbol{x}\right) = \boldsymbol{P}\left(\boldsymbol{x}\right) - \boldsymbol{x}_c^{\star}, \tag{2.41}$$

where $\boldsymbol{x}_c^\star$ represents the solution of the coarse model minimization. The space mapping function $\boldsymbol{P}$ is defined as

$$\boldsymbol{x}_c = \boldsymbol{P}(\boldsymbol{x}) = \arg\min_{\boldsymbol{x} \in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}_f)|| . \tag{2.42}$$

Thus, the coarse model is optimized with the aim to satisfy the design specification $\boldsymbol{f}(\boldsymbol{x}_f)$ for a given $\boldsymbol{x}_f$.

The quasi-Newton method is applied to (2.41), which requires a minimization of the coarse model at each Newton iterate. Therefore, the solution $\boldsymbol{x}_k$ is updated at each iteration k with

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + J_k^{-1}(\boldsymbol{x}_c^\star - \boldsymbol{x}_{k_c}) . \tag{2.43}$$

The inverse of the Jacobian of the space mapping function J_k^{-1} is updated directly with the Sherman-Morrison formula

$$J_k^{-1} = J_{k-1}^{-1} + \frac{\Delta\boldsymbol{x}_k - J_{k-1}^{-1}\Delta\boldsymbol{z}_k}{\Delta\boldsymbol{x}_k^T J_{k-1}^{-1}\Delta\boldsymbol{z}_k}\left(\Delta\boldsymbol{x}_k^T J_{k-1}^{-1}\right), \tag{2.44}$$

with $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\Delta\boldsymbol{z} = \boldsymbol{z}_k - \boldsymbol{z}_{k-1}$, and $\boldsymbol{z}_k$ is determined by evaluating the space mapping function or the so-called parameter extraction process

$$\boldsymbol{x}_{k_c} = \arg\min_{\boldsymbol{x} \in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}_k)|| . \tag{2.45}$$

Anderson acceleration as described in Sect. 2.3 is used to find the solution of the coarse model optimization problem where the design specification is given as $\boldsymbol{q}_k = \boldsymbol{f}(\boldsymbol{x}_k)$.

Contrary to manifold mapping, the convergence of the space mapping iterate to the solution of the fine model problem is not guaranteed, as proven in Echeverria and Hemker [41]. Scholcz et al. [90] also observe this problem for a fluid-structure interaction problem. Their solution is to switch from the space mapping algorithm to the Anderson acceleration method in case the method is not converging.

2.5.4. Output space mapping (OSM)

Output space mapping aims at reducing the misalignment of the coarse model response with the fine model response by adding the difference $\boldsymbol{d}_k =$

$\boldsymbol{x}_0 = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x})||$
for $k = 0 \rightarrow k_{\max}$ **do**
  $\boldsymbol{d}_k = \boldsymbol{c}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_k)$
  $\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{d}_k||$
  if converged **then**
    break
  end if
end for

Figure 2.7.: Output space mapping (OSM-0)

$\boldsymbol{c}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_k)$ to the coarse model response $\boldsymbol{c}$. At each space mapping iteration k, the following coarse model optimization problem is solved:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{d}_k||, \tag{2.46}$$

which can be interpreted as the manifold mapping algorithm where the mapping matrix T_k is kept constant as the identity matrix, which is denoted by OSM-0. The algorithm is shown in Fig. 2.7.

A second formulation of the output space mapping technique includes the Jacobian of the misalignment of the coarse model response with the fine model response:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}\in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{d}_k - J_k \cdot (\boldsymbol{x} - \boldsymbol{x}_k)||, \tag{2.47}$$

denoted by OSM-1. In this work, the Jacobian J_k is updated at each iteration k with the Broyden update

$$J_k = J_{k-1} + \frac{\Delta\boldsymbol{d}_k - J_{k-1}\Delta\boldsymbol{x}_k}{||\Delta\boldsymbol{x}_k||}\Delta\boldsymbol{x}_k^T, \tag{2.48}$$

with $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\Delta\boldsymbol{d}_k = \boldsymbol{d}_k - \boldsymbol{d}_{k-1}$. The algorithm is shown in Fig. 2.8.

1: $\boldsymbol{x}_0 = \arg\min_{\boldsymbol{x} \in X} ||\boldsymbol{c}(\boldsymbol{x})||$
2: $J_0 = I$
3: **for** $k = 0 \rightarrow k_{\max}$ **do**
4: $\quad \boldsymbol{d}_k = \boldsymbol{c}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_k)$
5: $\quad \boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x} \in X} ||\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{d}_k - J_k \cdot (\boldsymbol{x} - \boldsymbol{x}_k)||$
6: $\quad \Delta\boldsymbol{x}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$
7: $\quad \Delta\boldsymbol{d}_k = \boldsymbol{d}_{k+1} - \boldsymbol{d}_k$
8: $\quad J_{k+1} = J_k + \frac{\Delta\boldsymbol{d}_k - J_k \Delta\boldsymbol{x}_k}{||\Delta\boldsymbol{x}_k||} \Delta\boldsymbol{x}_k^T$
9: $\quad$ **if** converged **then**
10: $\quad\quad$ **break**
11: $\quad$ **end if**
12: **end for**

Figure 2.8.: Output space mapping (OSM-1)

Chapter 3.

Numerical results for different coupling schemes

> Life's most persistent and urgent question is, 'What are you doing for others?'
>
> *(Martin Luther King, Jr.)*

In this chapter the performance of the manifold mapping algorithm for fluid-structure interaction is demonstrated on several test problems: unsteady flow in a one-dimensional tube [34, 36], the two-dimensional incompressible laminar flow around a fixed cylinder with an attached elastic cantilever benchmark [96], and three-dimensional incompressible flow through a flexible tube. A reduction of approximately 50 % in terms of high-fidelity iterations is achieved compared to the state-of-the-art Anderson acceleration algorithm. The convergence of the high-fidelity model is accelerated even further when information from previous time steps is reused.

Parts of this chapter have been published in

- D. S. Blom et al. "Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 296 (2015), pp. 211–231. DOI: 10.1016/j.cma.2015.08.004;
- D. S. Blom et al. "Multi-Level Acceleration of Parallel Coupled Partitioned Fluid-Structure Interaction with Manifold Mapping". In: *Lecture Notes in Computational Science and Engineering*. Vol. 105. Springer International Publishing, 2015, pp. 135–150. DOI: 10.1007/978-3-319-22997-3_8.

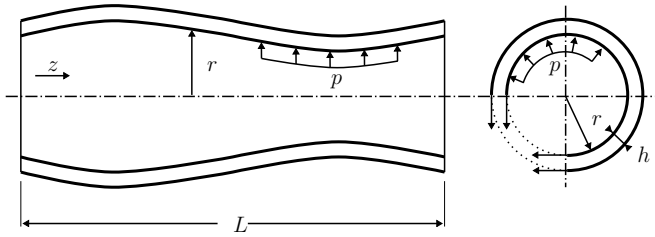


Figure 3.1.: Geometry of the one-dimensional model for the unsteady, incompressible flow through an elastic tube. The length L, the inner radius r, the pressure p acting on the inner wall of the tube, and the wall thickness h is shown. Deformation of the tube walls in radial direction is caused by the fluid pressure p acting on the inner tube walls. The figure is taken from Mehl et al. [76].

3.1. Unsteady flow in a one-dimensional tube

The first case consists of an incompressible, unsteady flow in a flexible tube [34, 36]. The tube has a circular cross-section and length L, as shown in Fig. 3.1. This example is straightforward to implement and is also a good example to verify the performance of a new coupling algorithm since the strength of the coupling can easily be changed.

3.1.1. Governing equation for the fluid model

The flow is governed by the equations for conservation of mass and conservation of momentum written in conservative form as

$$\frac{\partial a}{\partial t} + \frac{\partial av}{\partial z} = 0, \tag{3.1}$$

and

$$\frac{\partial av}{\partial t} + \frac{\partial av^2}{\partial z} + \frac{1}{\rho_f}\left(\frac{\partial ap}{\partial z} - p\frac{\partial a}{\partial z}\right) = 0, \tag{3.2}$$

with the coordinate z along the axis of the elastic tube, the cross-sectional area a and the velocity v in z-direction. r is the inner radius of the tube, the time is denoted by t, p represents the pressure, and ρ_f is the density of the fluid. Gravity and viscosity are not taken into account by the model.

3.1.2. Governing equation for the structure model

A Hookean constitutive relation describes the behavior of the elastic tube wall. The inertia of the tube wall is neglected with respect to the inertia of the fluid, resulting in a structure without mass. The stress acting in circumferential direction $\sigma_{\phi\phi}$ is approximated with

$$\sigma_{\phi\phi} = E\frac{r - r_0}{r_0} + \sigma_{\phi\phi_0} \tag{3.3}$$

for a given Young's modulus E and radius r_0 for which $\sigma_{\phi\phi} = \sigma_{\phi\phi_0}$. Only radial motion of the tube wall is allowed. The forces acting on the fluid-structure interface are in balance, thus

$$rp = \sigma_{\phi\phi} h. \tag{3.4}$$

The following relation can be found after substitution of the constitutive equation (3.3) into (3.4)

$$rp = \frac{Eh}{\rho r_0}(r - r_0) + r_0 p_0, \tag{3.5}$$

which can also be written as

$$a = a_0 \left(\frac{\frac{p_0}{2\rho_f} - c_{MK}^2}{\frac{p}{2\rho_f} - c_{MK}^2} \right)^2, \tag{3.6}$$

given the Moens-Korteweg wave speed c_{MK}

$$c_{MK} = \sqrt{\frac{Eh}{2\rho_f r_0}}. \tag{3.7}$$

A non-reflecting boundary condition

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{1}{c\rho_f}\frac{\mathrm{d}p}{\mathrm{d}t} \tag{3.8}$$

is imposed at the outlet of the tube.

3.1.3. Discretization

The tube is discretized with N cells of length Δz. The velocity and pressure of the flow are determined in the cell centers. A central discretization is employed for the continuity and momentum equation. However, the convective term of the momentum equation is discretized with a first-order upwind scheme. A backward Euler time integration scheme is used with a time step Δt.

Thus, the discretized conservation of mass in a control volume is given by

$$\frac{\Delta z}{\Delta t}\left(a_j - a_j^n\right) + v_{j+\frac{1}{2}} a_{j+\frac{1}{2}} - v_{j-\frac{1}{2}} a_{j-\frac{1}{2}} - \frac{\alpha}{\rho_f}\left(p_{j+1} - 2p_j + p_{j-1}\right) = 0, \quad (3.9)$$

and the discretized conservation of momentum in a control volume is

$$\frac{\Delta z}{\Delta t}\left(v_j a_j - v_j^n a_j^n\right) + v_j v_{j+\frac{1}{2}} a_{j+\frac{1}{2}} - v_{j-1} v_{j-\frac{1}{2}} a_{j-\frac{1}{2}} + \frac{1}{2\rho_f}\left(a_{j+\frac{1}{2}}\left(p_{j+1} - p_j\right) + a_{j-\frac{1}{2}}\left(p_j - p_{j-1}\right)\right) = 0. \quad (3.10)$$

The subscript j, $j+1$ and $j-1$ indicates the cell centers and the subscripts $j+\frac{1}{2}$ and $j-\frac{1}{2}$ denotes the values at the face of cells, calculated with $v_{j-\frac{1}{2}} = \frac{v_{j-1}+v_j}{2}$ and $v_{j+\frac{1}{2}} = \frac{v_j+v_j+1}{2}$. Coefficient $\alpha = \frac{a_0}{v_0+\frac{\Delta z}{\Delta t}}$ is a pressure stabilization term which is added to the continuity equation to remove any pressure wiggles as a result of the central discretization of the pressure in the momentum equation, with a given reference fluid velocity v_0.

The boundary condition at the outlet for the pressure is discretized as

$$p_{out} = 2\rho_f \left[c_{MK}^2 - \left(\sqrt{c_{MK}^2 - \frac{p_{out}^n}{2\rho_f}} - \frac{v_{out} - v_{out}^n}{4}\right)^2\right]. \quad (3.11)$$

The velocity at the inlet is calculated as

$$v_{in} = v_0 + \frac{v_0}{10}\sin^2\left(\pi n \tau\right), \quad (3.12)$$

where the dimensionless parameter τ is given by the dimensionless time

step $\tau = \frac{v_0 \Delta t}{L}$. Linear extrapolation is performed to determine the pressure at the inlet

$$p_{in} = 2p_1 - p_2, \tag{3.13}$$

and the velocity at the outlet

$$v_{out} = 2v_N - v_{N-1}. \tag{3.14}$$

The problem can be described with two dimensionless parameters, namely the dimensionless stiffness κ

$$\kappa = \frac{c_0}{v_0} = \frac{\sqrt{\frac{Eh}{2\rho_f r_0} - \frac{p_0}{2\rho_f}}}{v_0}, \tag{3.15}$$

and the dimensionless time step $\tau = \frac{v_0 \Delta t}{L}$. For the computations shown in this section, these two parameters are set to $\kappa = 10$ and $\tau = 0.01$, which results in a strong coupling between the fluid and the structure [36]. One period of the inlet boundary condition is simulated, thus 100 time steps. The initial conditions are the dimensionless velocity $\frac{v_0}{c_0} = 0.1$, the dimensionless pressure $\frac{p}{\rho_f c_0^2} = 0$, and the dimensionless cross-sectional area $\frac{a}{a_0} = 1$. A consistent interpolation is used on the interface with radial basis functions utilizing the globally supported thin plate spline as the basis function [18].

3.1.4. Numerical results

First, we study the performance of a range of coupling schemes as discussed earlier. After that, we shortly touch upon the choice of the termination criterion for the surrogate model.

Comparison of different coupling algorithms

The performance of the manifold mapping algorithm is compared with aggressive space mapping [5, 90], output space mapping [68], multi-level IQN-ILS (ML-IQN-ILS) [36], and with the IQN-ILS coupling technique [33]. The effect of reuse of information from previous time steps is also demonstrated. Table 3.1 shows the number of iterations for the fine model and the coarse model per time step averaged over the entire simulation. Three

Table 3.1.: One-dimensional flow through a flexible tube. The influence of the used coarse model is studied. Three different grid sizes for the coarse model are used: 100, 50 and 20. The manifold mapping algorithm (MM) is compared with the quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, aggressive space mapping (ASM), and output space mapping (OSM). The number of coupling iterations per time step has been averaged over the complete simulation.

Method	**FSI-100**		**FSI-50**		**FSI-20**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
QN(0)		10.2		10.2		10.2
IQN-ILS(0)		8.3		8.3		8.3
ML-IQN-ILS(0)	9.2	7.2	9.3	8.5	9.6	9.0
OSM-0(0)	56.8	12.8	84.2	19.3	222.1	50.3
OSM-1(0)	48.0	6.3	53.8	7.2	62.1	8.3
ASM(0)	24.7	5.4	27.2	6.0	35.9	7.8
MM(0)	24.4	5.2	26.7	5.5	36.0	7.3

coarse models with different grid sizes are used, namely with 20 cells (FSI-20), 50 cells (FSI-50) and 100 cells (FSI-100). The fine model uses 1000 cells for every computation. The initial solution of the displacement is determined by a state extrapolation from previous time steps for each numerical method under consideration. The convergence criterion for the fine model is set to $\tau_f = 10^{-6}$, and for the coarse model to $\tau_f = 10^{-7}$.

The use of manifold mapping leads to a reduction in the number of fine model evaluations of approximately 37 % compared to IQN-ILS for the FSI-100 case. When the grid size of the coarse model is decreased, the reduction in the number of fine model evaluations is approximately 33 % and 12 % for FSI-50 and FSI-20, respectively.

Similar results are shown in Table 3.2 when reusing information from previous time steps is considered. The use of manifold mapping with reuse of information from previous time steps leads to a reduction in the number of fine model evaluations of approximately 23 %, 16 % and 10 % compared to IQN-ILS with reuse for the FSI-100, FSI-50 and FSI-20 cases, respectively.

Table 3.2.: One-dimensional flow through a flexible tube. The influence of the used coarse model and the reuse of information from previous time steps is studied. The information from four previous time steps is reused. Three different grid sizes for the coarse model are used: 100, 50 and 20. The manifold mapping algorithm (MM) is compared with the quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, aggressive space mapping (ASM), and output space mapping (OSM). The number of coupling iterations per time step has been averaged over the complete simulation.

Method	**FSI-100**		**FSI-50**		**FSI-20**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
QN(4)		5.1		5.1		5.1
IQN-ILS(4)		3.1		3.1		3.1
ML-IQN-ILS(4)	3.8	3.4	3.8	3.8	3.8	4.4
OSM-0(4)	59.8	12.8	90.8	19.3	259.9	50.1
OSM-1(4)	67.6	6.0	67.9	6.2	51.0	6.7
ASM(4)	18.4	4.2	19.8	4.5	23.4	5.2
MM(4)	8.0	2.4	8.7	2.6	9.6	2.8

In contrast to manifold mapping and space mapping, the multi-level IQN-ILS algorithm optimizes the coarse model only once per time step. The constructed Jacobian of the coarse model is used to accelerate the convergence of the fine model. Manifold mapping and space mapping iteratively improve the quality of the coarse model approximation explaining the difference in fine model iterations, and also the increase in the number of coarse model iterations compared to ML-IQN-ILS.

Output space mapping with the Jacobian included in the surrogate model (OSM-1), and without the mapping Jacobian (OSM-0) are both considered. The use of the mapping Jacobian results in a significant decrease in the number of fine model evaluations compared to OSM-0. Reuse of information from previous time steps does not seem to result in an acceleration of the algorithm.

Concluding, manifold mapping shows the best performance regarding the number of fine model iterations in comparison with aggressive space mapping, multi-level IQN-ILS, and output space mapping. The reuse of information from previous time steps further accelerates the algorithm significantly.

Convergence measure for the surrogate model

For the surrogate model, different convergence measures can be used. Now, we compare the relative convergence measure as shown in Eqn. (2.39) with the relative residual convergence measure as shown in Eqn. (2.40). Table 2.40 shows the iteration counts when using 100 cells for the fine model, and 50 cells for the coarse model which uses a linearized fluid model. In each simulation, we use the manifold mapping algorithm to drive the interface residual below the tolerance $\tau_f = 1 \cdot 10^{-5}$.

The relative convergence measure for the first set of simulations is set to $\tau_c = 1 \cdot 10^{-7}$. The residual relative convergence measure for the second set of simulations is set to $\tau_c = 1 \cdot 10^{-3}$. As shown in the table, the residual relative convergence measure results in less coarse model iterations and does not influence the number of fine model iterations significantly when compared to the relative convergence measure. This is because the termination criterion allows for fewer iterations during the first manifold mapping iterations. The convergence measure becomes more strict, as the manifold mapping iteration process converges to the fine model optimum.

Table 3.3.: One-dimensional flow through a flexible tube. The influence of the termination criteria for the surrogate model is studied.

Convergence measure	**Reuse 0**		**Reuse 4**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
Relative convergence measure	32.8	4.1	10.4	2.1
Residual relative convergence measure	24.1	4.0	8.5	2.2

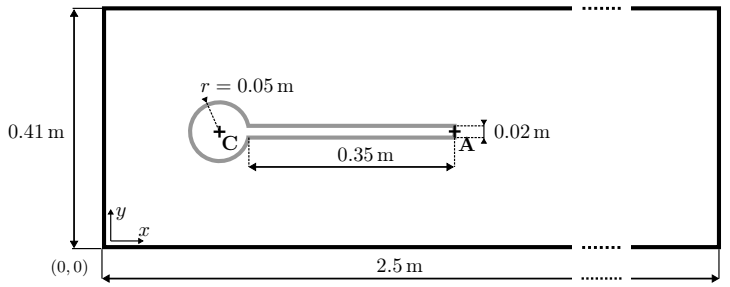


Figure 3.2.: Geometry of the cylinder with an attached flap benchmark originally proposed in [96]. The figure shows a channel flow with a fixed cylinder and a flexible flap attached to the cylinder. Point A located at the back of the flap is used as a reference point for the measurements of the displacement. The figure has been taken from Mehl et al. [76].

3.2. Fixed cylinder with an attached flexible flap

The second test case, originally proposed in [96], consists of a two-dimensional incompressible laminar flow around a fixed cylinder with an attached flexible cantilever. The geometry of the problem is shown in Fig. 3.2.

3.2.1. Governing equations of the fluid model

The governing equations for the fluid are given by the balance of momentum

$$\rho^f \frac{\partial \boldsymbol{v}^f}{\partial t} + \rho^f \left(\nabla \boldsymbol{v}^f\right) \boldsymbol{v}^f = \nabla \cdot \boldsymbol{\sigma}^f, \tag{3.16}$$

and the balance of mass, which reduces to a divergence free constraint on the velocity field due to incompressibility:

$$\nabla \cdot \boldsymbol{v}^f = 0 \qquad \text{in } \Omega^f, \tag{3.17}$$

where the velocity field is denoted by $\boldsymbol{v}^f$, the pressure field is denoted by p^f, and ρ^f represents the density. A Newtonian fluid is considered, which leads to the constitutive equation for the stress tensor $\boldsymbol{\sigma}^f$

$$\boldsymbol{\sigma}^f = -p^f \boldsymbol{I} + \rho^f \nu^f \left(\nabla \boldsymbol{v}^f + \nabla \boldsymbol{v}^{f^T} \right), \tag{3.18}$$

for a given kinematic viscosity ν^f.

3.2.2. Governing equations of the structure model

The configuration of the structure domain is described by the displacement $\boldsymbol{u}^s$. The structure is assumed to be elastic and compressible, and the governing equation is given by the balance of momentum

$$\rho^s \frac{\partial \boldsymbol{v}^s}{\partial t} + \rho^s \left(\nabla \boldsymbol{v}^s \right) \boldsymbol{v}^s = \nabla \cdot \boldsymbol{\sigma}^s + \rho^s \boldsymbol{g} \qquad \text{in } \Omega_t^s. \tag{3.19}$$

Equation (3.19) is modified to use the total Lagrangian description, i.e. with respect to a fixed initial reference state Ω^s, resulting in

$$\rho^s \frac{\partial^2 \boldsymbol{u}^s}{\partial t^2} = \nabla \cdot \left(J \boldsymbol{\sigma}^s \boldsymbol{F}^{-T} \right) + \rho^s \boldsymbol{g} \qquad \text{in } \Omega^s, \tag{3.20}$$

where the deformation gradient tensor $\boldsymbol{F}$ is defined as $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}^s$, and the Jacobian J is the determinant of the deformation gradient tensor $\boldsymbol{F}$. By applying the constitutive law for the St. Venant-Kirchhoff material, the Cauchy stress tensor $\boldsymbol{\sigma}^s$ is found by applying

$$\boldsymbol{\sigma}^s = \frac{1}{J} \boldsymbol{F} \left(\lambda^s \left(\operatorname{tr} \boldsymbol{E} \right) \boldsymbol{I} + 2\mu^s \boldsymbol{E} \right) \boldsymbol{F}^T, \tag{3.21}$$

with $\boldsymbol{E} = \frac{1}{2} \left(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I} \right)$, and the shear modulus μ^s [24].

3.2.3. Boundary conditions

A parabolic velocity profile is prescribed at the inlet, whereas a free outflow condition is prescribed at the outlet. No-slip conditions are prescribed for the fluid on the top and bottom wall, the cylinder, and the fluid-structure interface.

At the fluid-structure interaction Γ^{fs}, the balance of stresses is enforced through

$$\boldsymbol{\sigma}^f \boldsymbol{n} = \boldsymbol{\sigma}^s \boldsymbol{n} \qquad \text{on } \Gamma^{fs}, \tag{3.22}$$

with the unit vector $\boldsymbol{n}$ normal to the fluid-structure interface Γ^{fs}, and the stress tensors $\boldsymbol{\sigma}^f$ and $\boldsymbol{\sigma}^s$. Also, at the fluid-structure interface the velocities $\boldsymbol{v}^f$ and $\boldsymbol{v}^s$ must be equal:

$$\boldsymbol{v}^f = \boldsymbol{v}^s \qquad \text{on } \Gamma^{fs}. \tag{3.23}$$

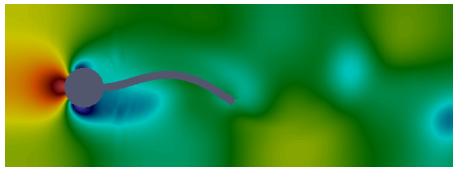
In case of inviscid flow, the slip boundary condition is applied.

3.2.4. Physical parameters

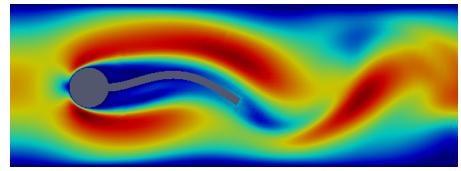
In [96] three different scenarios are presented with different solid to fluid density ratios. For the results shown in this section, the two unsteady fluid-structure interaction cases are selected, namely FSI2 and FSI3. The solid to fluid density ratio are set to $\frac{\rho^s}{\rho^f} = 10$ and $\frac{\rho^s}{\rho^f} = 1$, respectively. The Reynolds number based on the diameter of the cylinder is 100 for the FSI2 case, and 200 for the FSI3 case. The reader is referred to [96] for further details on this benchmark problem.

3.2.5. Discretization

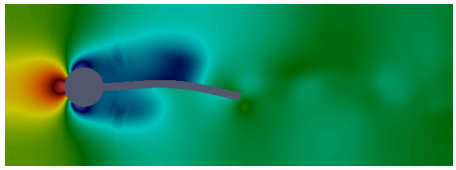
The fluid domain and structure domain are discretized with a second order finite volume method. A coupled solution algorithm [31] is employed, instead of the well known PISO pressure-velocity coupling technique. Here, the continuity and momentum equation are solved in a fully coupled implicit manner, instead of a segregated approach. A second order backward differencing scheme (BDF2) is used to integrate the governing equations in time. The fluid mesh is deformed with radial basis function interpolation [18]. The coarse model of the flow uses 1 457 cells. The grid is refined



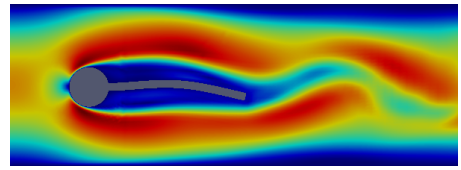
(a) FSI2: pressure contours



(b) FSI2: velocity field



(c) FSI3: pressure contours



(d) FSI3: velocity field

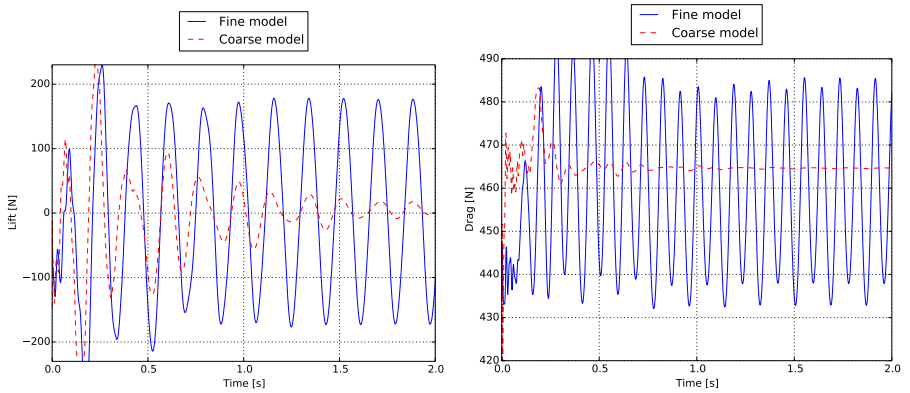
Figure 3.3.: Cylinder with an attached flap test case. Pressure and velocity contours of the FSI2 and FSI3 problems.

in each direction with a factor 4, resulting in 23 924 cells for the fine model of the fluid. The mesh of the structural model consists of 40 cells and is uniformly refined to form the fine mesh of the solid containing 328 cells. The computational cost of one coarse model evaluation is approximately 1.4 % of one fine model evaluation.

Pressure and velocity contours of the FSI2 and FSI3 problems are shown in Fig. 3.3. For the FSI3 case, the lift and drag are shown for the coarse model and the fine model in Fig. 3.4 for separate computations. The coarse model finds a steady state solution, whereas the fine model finds the periodic solution.

3.2.6. Numerical results

The numerical results for this test case are shortly discussed. First, the performance of the manifold mapping acceleration technique is compared with standard coupling techniques. Thereafter, the difference in coupling iterations between the serial and parallel coupling of fluid and solid solvers is discussed.



(a) Lift (b) Drag

Figure 3.4.: Cylinder with an attached flap test case FSI3. Lift and drag are shown for $t = 0\,\mathrm{s}$ until $t = 2\,\mathrm{s}$ for the fine model as well as the coarse model. The computations shown are executed separately. Note that the fluid domain is initialized with a fully developed flow.

Comparison of different coupling schemes in terms of sub-iterations

Table 3.4 and 3.5 list the number of coupling iterations per time steps for the fine and coarse model averaged over one periodic cycle. Again, the performance of the manifold mapping acceleration technique is compared with the quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, aggressive space mapping, and output space mapping. The initial solution of the displacement is determined by a state extrapolation from previous time steps for each numerical method under consideration.

Figure 3.5 shows a graph of the convergence history within one time-step for the different coupling algorithms considered. Information from previous time steps is not reused. The figure highlights the fast convergence of the manifold mapping and aggressive space mapping technique compared to IQN-ILS.

For the FSI2 benchmark, the number of fine model evaluations is reduced by approximately 49 % in comparison with the fine model only (IQN-ILS). The FSI3 test case shows the same reduction in fine model evaluations of 49 % in comparison with the IQN-ILS algorithm. With reuse of information

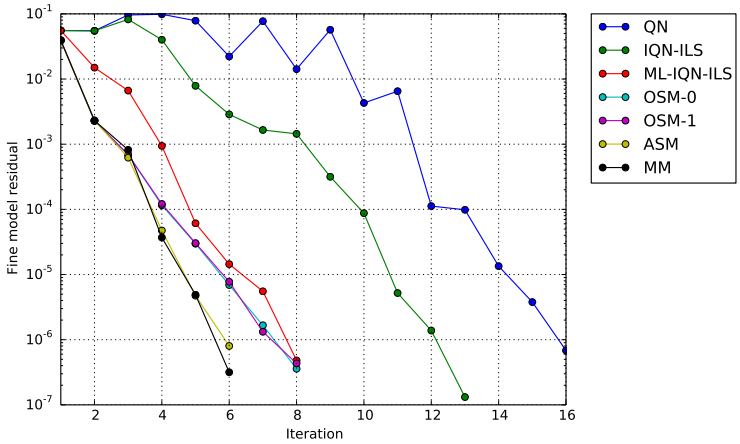


Figure 3.5.: Two dimensional flow over a fixed cylinder with an attached flexible flap FSI3. The convergence history of the different coupling algorithms is shown for one time step.

from previous time steps, the manifold mapping algorithm can reduce the number of iterations per time step to approximately two iterations for both cases.

Results for output space mapping with the mapping Jacobian included in the surrogate model (OSM-1), and without the mapping Jacobian (OSM-0) are shown in Table 3.4 and 3.5. For the FSI2 benchmark, the use of the Jacobian leads to an increase in fine model iterations, which indicates that the quality of the Jacobian is not sufficient to result in a speedup. However, for the FSI3 benchmark, the use of the mapping Jacobian results in a minor decrease of fine model evaluations compared to OSM-0. Also, the reuse of information from previous time steps does not influence the results significantly.

The aggressive space mapping algorithm is competitive to manifold mapping in case information from previous time steps is not included. However, with information from previous time steps, the decrease in the number of fine model iterations is less compared to manifold mapping.

For the FSI2 case, the ML-IQN-ILS shows similar iteration numbers compared to manifold mapping. For the FSI3 case with a strong coupling between the fluid and the solid domain, the differences between manifold

Table 3.4.: Two-dimensional flow over a fixed cylinder with an attached flexible flap FSI2. The influence of the reuse of information from previous time steps is studied for different coupling algorithms for the fluid-structure interaction problem. Manifold mapping (MM) is compared with quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, aggressive space mapping (ASM), and output space mapping (OSM).

Method	**Reuse 0**		**Reuse 4**		**Reuse 8**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
QN		10.6		6.3		5.7
IQN-ILS		8.6		3.6		2.9
ML-IQN-ILS	11.6	9.6	4.6	5.3	3.4	4.5
OSM-0	25.2	4.6	19.6	4.5	19.4	4.5
OSM-1	38.1	5.9	47.5	5.3	45.6	5.1
ASM	27.3	4.3	16.0	3.4	16.0	3.4
MM	24.3	4.4	5.8	2.1	5.4	2.0

mapping and ML-IQN-ILS are more substantial. The ML-IQN-ILS algorithm does not return to the coarse model after a fine model evaluation and performs fewer iterations with the coarse model as a result compared to manifold mapping, aggressive space mapping, and output space mapping.

Summarizing, the manifold mapping algorithm results in the largest speedup in terms of the fine model evaluations compared to other coupling algorithms. By including information from previous time steps, the number of iterations per time step is decreased substantially.

Serial and parallel execution of fluid and solid solvers

Table 3.6 shows the averaged iteration numbers for the manifold mapping algorithm and also for IQN-ILS and ML-IQN-ILS. Reuse of information from previous time steps is considered in order to accelerate the convergence of the different coupling schemes. Sequential execution (S) of the fluid and structure solvers is compared with the parallel case (P).

In case the sequentially coupled manifold mapping technique (S-MM) is used and information from previous time steps is not included, approxi-

Table 3.5.: Two-dimensional flow over a fixed cylinder with an attached flexible flap FSI3. The influence of the reuse of information from previous time steps is studied for different coupling algorithms for the fluid-structure interaction problem. Manifold mapping (MM) is compared with quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, aggressive space mapping (ASM), and output space mapping (OSM).

Method	**Reuse 0**		**Reuse 4**		**Reuse 8**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
QN		16.0		7.8		6.7
IQN-ILS		11.6		4.5		3.1
ML-IQN-ILS	13.5	8.9	4.9	3.7	4.3	3.3
OSM-0	43.1	7.4	43.3	7.4	44.1	7.4
OSM-1	53.3	7.0	62.7	5.9	59.0	5.8
ASM	37.0	6.0	27.0	4.9	28.3	4.9
MM	35.1	5.7	8.5	2.4	6.6	2.1

Table 3.6.: Two-dimensional flow over a fixed cylinder with an attached flexible flap FSI3. The influence of the reuse of information from previous time steps is studied for different coupling algorithms for the fluid-structure interaction problem. Manifold mapping (MM) is compared with IQN-ILS and ML-IQN-ILS. Two different cases are considered, the flow and the structure solver are executed sequentially (S) or in parallel (P). The iteration numbers are averaged over the complete simulation.

Method	**Reuse 0**		**Reuse 8**		**Reuse 16**		**Reuse 24**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
S-IQN-ILS		11.6		3.1		3.3		3.6
S-ML-IQN-ILS	13.5	8.9	4.3	3.3	4.5	3.7	4.6	3.7
S-MM	35.1	5.7	6.6	2.1	6.5	2.0	6.7	2.0
P-IQN-ILS		26.3		5.7		5.7		8.1
P-ML-IQN-ILS	28.8	20.6	10.8	12.7	13.3	18.6	14.4	20.8
P-MM	150.6	11.2	29.4	3.4	32.9	3.0	34.7	2.9

mately 5.7 fine model iterations are performed at every time step. IQN-ILS uses almost twice as many iterations per time step (11.6). With reuse of information from previous time steps, the least number of fine model iterations is 3.1 for IQN-ILS. However, this number increases in case information from more than eight time-steps is included. For the manifold mapping algorithm, the number of fine model iterations does not increase in case information from a large number of time steps is taken into account.

In case the fluid and the structure solver are executed in parallel, the number of fine model iterations increases to 11.2 for the manifold mapping technique. However, IQN-ILS uses 26.3 iterations per time step. In case information from previous time steps is reused by the different coupling algorithms, the number of fine model iterations decreases significantly. Again, it is important to not include information from a large number of time steps for IQN-ILS, since the number of iterations increases from 5.7 (8 or 16 time-steps reused) to 8.1 (24 time-steps reused). For the manifold mapping algorithm, an increase in fine model iterations is not observed in case the information from an increasing number of time steps is reused.

As observed before, the ML-IQN-ILS algorithm uses significantly less coarse model iterations in comparison to the manifold mapping technique since the method optimizes the coarse model only once per time step, whereas for the manifold mapping algorithm the coarse model is optimized at every manifold mapping iteration. Note, in case the fluid and the structure are coupled in parallel, the number of fine model iterations is increased significantly compared to the staggered execution of the fluid and the structure solver.

Table 3.7 shows the averaged iteration numbers for the different coupling algorithms in case the Jacobians of the different coupling algorithms are updated with the appropriate secant equation to take into account information from all previous time steps. If we compare these results to those of Table 3.6, Table 3.6 shows worse performance if no time steps are reused, but better performance if the optimal number of time steps are reused.

Table 3.7.: Two-dimensional flow over a fixed cylinder with an attached flexible flap FSI3. Manifold mapping (MM) is compared with IQN-ILS and ML-IQN-ILS. Two different cases are considered, the flow and structure solver are executed sequentially (S) or in parallel (P). The Jacobians of the different coupling algorithms are updated with the appropriate secant equation to take into account information from all previous time steps. The iteration numbers are averaged over the complete simulation.

Method	**Sequential**		**Parallel**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
IQN-ILS		4.4		7.6
ML-IQN-ILS	5.4	4.8	8.5	8.2
MM	14.2	3.2	49.9	5.5

3.3. Wave propagation in a three-dimensional elastic tube

The third example consists of a wave propagating in a straight, three-dimensional elastic tube [9, 34, 49]. The geometry of the fluid and the solid domain is shown in Fig. 3.6. The length of the tube is 0.05 m. The tube has a thickness of 0.001 m. The inner diameter of the tube is 0.01 m. Both ends of the tube are fixed. Starting from $t = 0\,\mathrm{s}$ until $t = 0.003\,\mathrm{s}$, the boundary condition for the gauge pressure at the inlet is set to the fixed value 1333.2 Pa. After that, the inlet gauge pressure is set to zero. At the outlet, the pressure is fixed at zero at every time instant. The pressure contours at different time instants are shown in Fig. 3.7.

3.3.1. Physical parameters

The fluid domain of the fine model is governed by the incompressible Navier-Stokes equations, as discussed in Sect. 3.2.1. The flow has a density of $10^3\,\mathrm{kg/m^3}$, and a dynamic viscosity of $3.0 \cdot 10^{-3}\,\mathrm{Pa \cdot s}$. The solid is assumed to be elastic and compressible, as described in Sect. 3.2.2. The density of elastic solid is $1.2 \cdot 10^3\,\mathrm{kg/m^3}$, the Young's modulus is $3.0 \cdot 10^5\,\mathrm{N/m^2}$, and Poisson's ratio is 0.3.

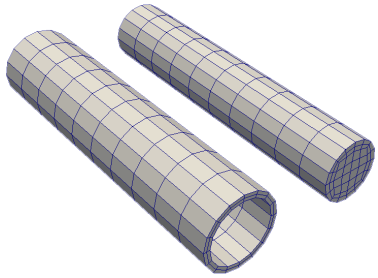
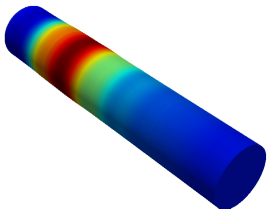
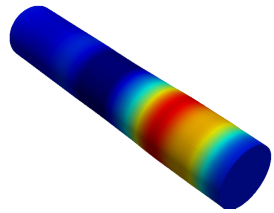


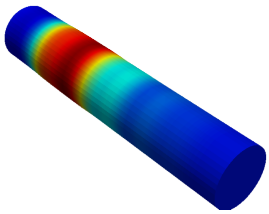
Figure 3.6.: Wave propagation in a straight elastic tube. The geometry and a very coarse mesh are shown for both the fluid and solid domain.



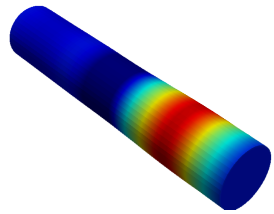
(a) $t = 0.0050\,\mathrm{s}$, fine model



(b) $t = 0.0100\,\mathrm{s}$, fine model



(c) $t = 0.0050\,\mathrm{s}$, coarse model



(d) $t = 0.0100\,\mathrm{s}$, coarse model

Figure 3.7.: Wave propagation in a straight tube. The pressure contours at different time instants on the fluid-structure interface. The solutions for the fine model are shown on the top, and solution for the coarse model are shown on the bottom. These simulations are executed separately.

The fluid domain of the coarse model is governed by the incompressible Euler equations, i.e. viscous effects are ignored. Another approach would be to use the one-dimensional solver described in Sect. 3.1 as the coarse model for this test case.

3.3.2. Discretization

The same solver as for the fixed cylinder with an attached flexible flap is used to simulate this problem. Thus, a finite volume solver based on a coupled solution algorithm is employed in combination with a second-order time integration scheme. The coarse model of the fluid uses 2 600 cells, and is refined in each direction with a factor 2, resulting in 20 800 cells for the fine model.

Two different cases are considered with regards to the coarse model of the structure. For the first example, the same structural model is used for the coarse model as well as the fine model, since the largest computational time is spent during the fluid evaluation (6 400 cells). The second example consists of a coarsened grid of the structural coarse model (800 cells). The initial solution of the displacement is determined by a state extrapolation from previous time steps for each numerical method under consideration. The computational cost of one coarse model evaluation is approximately 1 % of one fine model evaluation.

3.3.3. Numerical results

The numerical results for this test case are shortly discussed. As for the previous test case, the performance of the manifold mapping acceleration technique is compared with standard coupling techniques. Thereafter, the difference in coupling iterations between the serial and parallel coupling of fluid and solid solvers is discussed.

Comparison of different coupling schemes in terms of sub-iterations

Table 3.8 and 3.9 show the averaged iteration numbers for the manifold mapping algorithm, and also for several other coupling algorithms. Comparing Table 3.8 with 3.9 shows an increase in the number of fine model iterations for the multi-level algorithms when using a coarsened mesh for

the structure, as is to be expected. However, the ML-IQN-ILS technique shows an increase of 2.8 fine model iterations in case the structural mesh is coarsened and reuse of information from previous time steps is not considered. However, for manifold mapping, aggressive space mapping and output space mapping with the Jacobian information included (OSM-1), the number of fine model iterations increases with approximately one iteration.

The use of the manifold mapping algorithm results in a reduction in the number of fine model evaluations of approximately 60 % compared to IQN-ILS without reuse of information from previous time steps, and where the structural mesh is coarsened. With reuse of information, a reduction of 42 % compared to IQN-ILS with reuse is observed for the manifold mapping technique.

The use of the aggressive space mapping algorithm results in a comparable reduction of fine model iterations when information from previous time steps is not reused. Reuse of information does not lead to significant computational savings for both cases.

Output space mapping with the mapping Jacobian included in the surrogate model (OSM-1), and without the mapping Jacobian included (OSM-0) are both considered. Table 3.8 and 3.9 show that the use of the mapping Jacobian in the surrogate model does not result in a decrease in the number of fine model iterations, compared to OSM-0. On the contrary, the reuse of information from previous time steps leads to an increase in the number of fine model evaluations, which indicates that the quality of the Jacobian is not sufficient to result in a speedup.

The use of the multi-level IQN-ILS algorithm results in a significant acceleration of the high-fidelity model. This is also the case when information from previous time steps is included by the algorithm. However, the manifold mapping algorithm is able to reduce the number of fine model evaluations to a larger extent.

Concluding, manifold mapping shows the best performance in terms of the number of fine model evaluations in comparison with the quasi-Newton method based on the Broyden update, IQN-ILS, ML-IQN-ILS, aggressive space mapping, and output space mapping. Coarsening of the structural mesh results in an increase of only one fine model iteration. A significant speedup is observed when information from previous time steps is included.

Table 3.8.: Three-dimensional flow through a flexible tube. The influence of the reuse of information from previous time steps is studied for different coupling algorithms for the fluid-structure interaction problem. Manifold mapping (MM) is compared with aggressive space mapping (ASM), quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, and output space mapping (OSM). The number of coupling iterations per time step has been averaged over the complete simulation. The structural model used for the coarse and the fine model consists of the same number of finite volumes.

Method	**Reuse 0**		**Reuse 4**		**Reuse 8**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
QN		24.7		14.5		12.1
IQN-ILS		15.6		7.5		5.9
ML-IQN-ILS	17.2	6.5	8.8	4.0	7.2	3.9
OSM-0	40.9	5.1	33.1	5.1	31.2	5.1
OSM-1	63.4	7.0	88.2	7.8	82.1	7.7
ASM	42.9	5.0	35.4	5.0	33.8	5.0
MM	40.0	5.0	19.6	3.4	16.4	3.2

Table 3.9.: Three-dimensional flow through a flexible tube. The influence of the reuse of information from previous time steps is studied for different coupling algorithms for the fluid-structure interaction problem. Manifold mapping (MM) is compared with aggressive space mapping (ASM), quasi-Newton method based on the Broyden update (QN), IQN-ILS, ML-IQN-ILS, and output space mapping (OSM). The number of coupling iterations per time step has been averaged over the complete simulation. The structural domain of the coarse model is coarsened in comparison with the fine model.

Method	**Reuse 0**		**Reuse 4**		**Reuse 8**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
QN		24.7		14.5		12.1
IQN-ILS		15.6		7.5		5.9
ML-IQN-ILS	18.1	9.3	9.0	6.1	7.2	4.5
OSM-0	54.0	6.9	45.2	6.9	43.5	6.9
OSM-1	75.7	8.0	92.3	8.2	85.0	8.0
ASM	52.5	6.0	42.0	5.9	39.8	5.7
MM	51.0	6.3	21.8	3.7	17.6	3.4

Table 3.10.: Three-dimensional flow through a flexible tube. The influence of the reuse of information from previous time steps is studied for different coupling algorithms for the fluid-structure interaction problem. Manifold mapping (MM) is compared with IQN-ILS and ML-IQN-ILS. Two different cases are considered, the flow and the structure solver are executed sequentially (S), or in parallel (P). The iteration numbers are averaged over the complete simulation.

Method	**Reuse 0**		**Reuse 8**		**Reuse 16**		**Reuse 24**	
	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$
S-IQN-ILS		15.6		5.9		5.1		4.9
S-ML-IQN-ILS	18.1	9.3	7.2	4.5	6.2	4.4	5.8	4.6
S-MM	51.0	6.3	17.6	3.4	15.7	3.4	15.4	3.3
P-IQN-ILS		30.2		11.0		9.5		8.7
P-ML-IQN-ILS	34.0	14.6	12.2	7.6	10.2	7.0	9.3	6.9
P-MM	89.3	6.6	37.3	4.5	31.9	4.3	29.8	4.2

Serial and parallel execution of fluid and solid solvers

Table 3.10 shows the averaged iteration numbers. The use of the manifold mapping algorithm reduces the number of fine model iterations from 15.6 for IQN-ILS to 6.3 in case the fluid and structure are executed in a sequential fashion and information from previous time steps is not considered. In case information from previous time steps is reused, the number of fine model iterations for IQN-ILS reduces to 4.9, whereas the manifold mapping technique uses just 3.3 iterations per time step. The S-ML-IQN-ILS scheme outperforms IQN-ILS in terms of fine model iterations, though the manifold mapping technique is observed to use the least amount of fine model iterations. Note that viscous effects are ignored for the coarse model of the fluid domain.

When the fluid and structure solvers are coupled in parallel, the number of fine model iterations increases slightly to 6.6 for the manifold mapping technique, whereas the number of iterations for IQN-ILS is almost twice as high compared to the sequential coupling case. Including information from previous time steps accelerates the manifold mapping significantly to only 4.2 fine model iterations per time steps.

Table 3.11.: Three-dimensional flow through a flexible tube. Manifold mapping (MM) is compared with IQN-ILS and ML-IQN-ILS. Two different cases are considered, the flow and the structure solver are executed sequentially (S), or in parallel (P). The Jacobians of the different coupling algorithms are updated with the appropriate secant equation in order to take into account information from all previous time steps. The iteration numbers are averaged over the complete simulation.

Method	**Sequential**		**Parallel**	
	$\mathbf{n_f}$	$\mathbf{n_c}$	$\mathbf{n_f}$	$\mathbf{n_c}$
IQN-ILS		6.8		11.7
ML-IQN-ILS	7.4	6.7	12.0	10.8
MM	28.5	4.6	44.5	5.8

Table 3.11 shows the averaged iteration numbers for the same coupling algorithms in case the full Jacobians are updated with the appropriate secant equation in order to take into account the information from all previous time steps. Again, the sequential and the parallel coupling of the fluid and structure solvers is examined. The manifold mapping algorithm clearly outperforms IQN-ILS and the ML-IQN-ILS technique in terms of fine model iterations for both sequential and parallel cases. The number of fine model iterations increases with approximately one iteration for the parallel case compared to sequential coupling, whereas the ML-IQN-ILS and IQN-ILS use 3 and 4 extra iterations, respectively.

3.4. Steady state for fixed cylinder with an attached flexible flap case

Thus far unsteady simulations have been performed. Now we are considering a test case with a steady state solution. Steady-state solutions are of much importance for industrial test cases. For example, when studying the cruise flight condition of an airplane, the wings are deformed due to the aerodynamic loading. Hence, there is an interaction between the structure and the flow.

Turek and Hron [96] discuss a third setup of the fixed cylinder with an

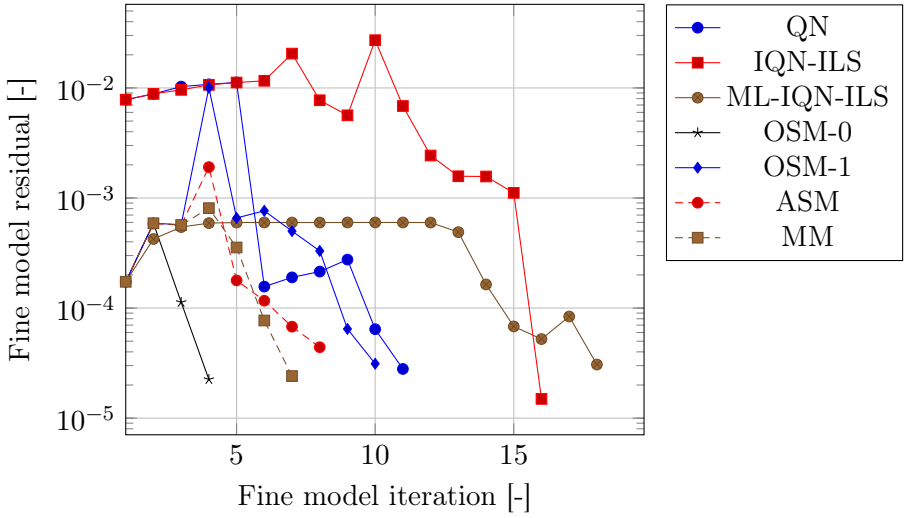


Figure 3.8.: Steady state solution of two dimensional flow over a fixed cylinder with an attached flexible flap FSI1. The convergence history of different coupling methods are shown.

attached flexible flap case which results in a steady state solution, named FSI1. A segregated solution algorithm is employed for the fluid domain to solve the incompressible Navier-Stokes equations. The fluid mesh is deformed with radial basis function interpolation. The coarse model of the flow uses 5 981 cells. After uniform refinement, a mesh is generated consisting of 23 924 cells for the fine model of the fluid. The mesh of the structural model consists of 82 cells and is uniformly refined to form the fine mesh of the solid containing 328 cells.

Figure 3.8 shows the convergence history of the quasi-Newton method using a Broyden update (QN), IQN-ILS method, ML-IQN-ILS method, output space mapping (OSM), aggressive space mapping (ASM) and manifold mapping (MM). The fluid and solid solvers are coupled in a serial manner. The figure highlights the fast and regular convergence of the manifold mapping algorithm compared to IQN-ILS. The use of manifold mapping results in a speedup of approximately 2.3 in terms of fine model iterations.

The ML-IQN-ILS method needs a lot of fine model iterations before the

method starts to converge. This is probably caused by the fact that the information from the coarse model is included directly in the approximation of the Jacobian of the fine model. Since the coarse model data is not accurate enough, these vectors actually hinder the convergence of the fine model.

The aggressive space mapping algorithm needs eight fine model iterations to converge but shows a smaller convergence rate during the last four iterations compared to manifold mapping.

Output space mapping with the mapping Jacobian included in the surrogate model (OSM-1) needs a significantly larger number of iterations to converge compared to without using the Jacobian. Output space mapping without the Jacobian outperforms manifold mapping for this example. The only difference between output space mapping without the mapping Jacobian and manifold mapping is the use of the mapping matrix T_k for manifold mapping. When T_k is kept constant as the identity matrix during the complete optimization process, it is equal to output space mapping. Therefore, it can be concluded that the mapping matrix T_k is not accurate enough during the first iterations for manifold mapping. The two algorithms show very similar convergence rates after the first couple of iterations.

3.5. Unsteady flow in a one-dimensional tube with three levels

A natural extension of the two-level manifold mapping algorithm is to use three or even more levels. Here, we study the use of three levels. The optimization of the coarse model is accelerated by the manifold mapping algorithm, with an even cheaper and less accurate low fidelity model. The earlier discussed unsteady flow in a flexible one-dimensional tube case is used again to study the use of three levels. From this point on, we only consider the manifold mapping technique as we have already established the good performance of the method compared to other coupling techniques in the previous sections.

Table 3.12 shows the iteration counts averaged over the complete simulation where two levels and three levels are considered with different mesh sizes and different settings for reuse of information from previous time

steps. The governing equations for each fluid model are linearized around the initial state, except for the fine fluid model. The fluid equations for the fine model are still the full non-linear equations, which are coupled to the solid solver. Hence, the fine model is accelerated by a substantially different coarse model. The mesh size of the models is indicated between the brackets. Thus, 2 (100, 1000) represents a two-level simulation with 100 cells for the coarse model, and 1000 cells for the fine model. 3 (20, 50, 1000) indicates a three-level simulation with 1000 cells for the fine model, and 20 and 50 cells for the coarse models.

The reuse of information from previous time steps results in the largest reduction in the number of sub-iterations per time step for the fine model, namely from 4.7 - 5.7 to 2.6 - 2.7 sub-iterations depending on the coarse model. The coarse model also benefits largely from reuse of information from previous time steps.

The addition of a third level results in a reduction of approximately 25 % - 27 % for the intermediate level (26.4 to 19.2 iterations, and 23.6 to 17.7 iterations) when information from previous time steps is not reused. The number of fine model iterations is not influenced significantly, which can be expected since the accuracy of the coarse models does not differ greatly, and a further speedup of the fine model is therefore not expected. In case information from previous time steps is reused, the addition of a third level results in a reduction of approximately 18 % - 23 % for the intermediate level (10.1 to 7.7 iterations, and 9.8 to 8.0 iterations).

For the 2 (50, 1000) and 3 (50, 100, 1000) level simulations, a new level is added in between the original coarse and the fine model with 100 cells. Thus, the accuracy of the coarse model is increased when compared to the two-level setup. One can expect a decrease in the number of fine model iterations for the three-level setup. The number of fine model iterations does decrease by approximately 8 %, but the number of iterations with the coarse model containing 50 cells is increased from 26.4 to 55.5. Thus, it is not necessarily the case that the addition of an extra model results in a speedup of the overall simulation.

Table 3.13 shows the number of iterations when the fluid and solid solvers are coupled in parallel. When information from previous time steps is not reused, the acceleration of the coarse model by a third level is substantially larger compared to the serial case, namely from 56.5 iterations to 15.6

Table 3.12.: One-dimensional flow through a flexible tube. The influence of three levels within the manifold mapping framework is studied. The number of coupling iterations per time step are shown. The information from previous time steps is reused (0, 4 and 8 time steps). The fluid and solid solver are coupled in a serial or staggered manner. The coarse models use a linearized fluid model instead of the full non-linear equations. Level 3 is the most accurate model, where level 1 and level 2 represent the coarse models.

Levels / Histories	**Reuse 0**			**Reuse 4**			**Reuse 8**		
	1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
2 (20, 1000)		27.9	5.7		10.0	2.8		9.8	2.7
2 (50, 1000)		26.4	5.3		9.6	2.7		10.1	2.7
2 (100, 1000)		23.6	4.7		9.5	2.6		9.8	2.6
3 (20, 50, 1000)	57.7	19.2	5.2	15.9	7.7	2.6	15.7	7.7	2.6
3 (20, 100, 1000)	51.6	16.8	4.7	16.1	8.0	2.6	15.8	7.8	2.6
3 (50, 100, 1000)	55.5	17.7	5.0	16.6	8.1	2.7	16.6	8.0	2.7

iterations for the coarse model with 100 cells (72 %), and 60.9 to 21.7 for the coarse model with 50 cells (64 %). However, when information from previous time steps is reused, it highly depends on the accuracy of the coarse model whether a speedup is observed.

3.6. Wave propagation in a three-dimensional elastic tube with three levels

A final example is considered where three levels are applied within the manifold mapping framework. The three-dimensional flow through a flexible tube example, as discussed in Sect. 3.3 is used to study the computational efficiency of the method when applied to three levels. Where previously a simplified one-dimensional model was used, now the fluid and solid domains are governed by the incompressible Navier-Stokes equations and the geometrically non-linear elasticity equations.

Table 3.13.: One-dimensional flow through a flexible tube. The influence of three levels within the manifold mapping framework is studied. The number of coupling iterations per time step is shown. The information from previous time steps is reused (zero, four and eight time-steps). The fluid and solid solver are coupled in a parallel manner. The coarse models use a linearized fluid model instead of the full nonlinear equations.

Levels / Histories	**Reuse 0**			**Reuse 4**			**Reuse 8**		
	1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
2 (20, 1000)		58.2	5.5		17.4	3.0		16.8	3.0
2 (50, 1000)		60.9	5.2		9.2	2.3		9.2	2.3
2 (100, 1000)		56.5	4.5		9.7	2.3		9.5	2.2
3 (20, 50, 1000)	100.0	21.7	6.2	23.2	9.3	3.1	22.6	9.2	3.0
3 (20, 100, 1000)	88.9	19.4	5.7	22.8	9.4	3.1	21.4	9.0	3.0
3 (50, 100, 1000)	85.8	15.6	4.5	15.6	7.0	2.3	14.9	6.7	2.2

3.6.1. Discretization

The same structural solver as in Sect. 3.3 is used. The fluid solver is also based on the finite volume method but uses an iterated PISO-like solver to solve the nonlinear incompressible Navier-Stokes equations. Again, a second order backward differencing time integration scheme is used for both the fluid as well as the structural domain.

The coarse models for the fluid domain are governed by the incompressible Euler equations. Hence, the coarse models actually simulate different physics. The meshes for the fluid and solid are refined uniformly. Refinement level 1 contains 2 600 cells for the fluid, and 800 cells for the solid. Refinement level 2 has 20 800 cells for the fluid, and 6 400 cells for the solid. Finally, the finest grids contain 166 400 cells for the fluid domain, and 51 200 cells for the solid.

3.6.2. Numerical results

Table 3.14 shows the iteration counts averaged over the complete simulation where the fluid and solid solvers are coupled in serial. One set of simulations does not reuse information from previous time steps, whereas

Table 3.14.: Three-dimensional flow through a flexible tube. The influence of three levels within the manifold mapping framework is studied. The number of coupling iterations per time step is shown. The information from previous time steps is reused (zero and four time-steps). The fluid and solid solver are coupled in a serial or staggered manner. For the fluid, the coarse models are governed by the incompressible Euler equations, where the fine model is governed by the incompressible Navier-Stokes equations. The values between parentheses indicate the refinement level of the mesh.

Levels / Histories	**Reuse 0**			**Reuse 4**		
	1	**2**	**3**	**1**	**2**	**3**
2 (1, 3)	40.4		4.6	14.9		2.4
2 (2, 3)		36.5	3.8		15.5	2.5
3 (1, 2, 3)	109.1	21.3	3.9	39.6	10.6	2.5

the other set reuses information from four previous time steps which is sufficient as shown in Sect. 3.3. The refinement level of the mesh for both the fluid and the solid domains is indicated between parentheses in Table 3.14. Thus simulation 2 (1, 3) indicates a setup with two levels, of which the coarse model uses refinement level 1, and the fine model uses refinement level 3.

When only two levels are considered, it can be observed that the manifold mapping method results in a small number of coupling iterations for the fine model, even when the coarse model is coarsened to the point where the computational costs are negligible (3.8 and 4.6 iterations without reuse, and 2.4 and 2.5 iterations with reuse). Note that the coarse model uses a different model for the fluid domain. The use of a more accurate coarse model decreases the number of fine model iterations in case information from previous time steps is not being reused.

However, when information from previous time steps is reused, the number of fine model iterations does not differ significantly between different setups. The use of three levels does decrease the number of coarse model iterations for refinement level 2 from 15.5 to 10.6 (32 %).

Table 3.15 shows the number of iterations per time steps when the fluid and solid solver are coupled in parallel. Similar conclusions from this table

Table 3.15.: Three-dimensional flow through a flexible tube. The influence of three levels within the manifold mapping framework is studied. The number of coupling iterations per time step is shown. The information from previous time steps is reused (zero and four time-steps). The fluid and solid solver are coupled in a parallel manner. For the fluid, the coarse models are governed by the incompressible Euler equations, where the fine model is governed by the incompressible Navier-Stokes equations. The values between parentheses indicate the refinement level of the mesh.

Levels / Histories	**Reuse 0**			**Reuse 4**		
	1	**2**	**3**	**1**	**2**	**3**
2 (1, 3)	87.6		6.0	32.3		3.7
2 (2, 3)		92.7	6.1		33.7	3.9
3 (1, 2, 3)	217.1	29.5	6.2	65.0	14.2	3.7

can be drawn as for the serial case, namely that the reuse of information from previous time steps results in the largest speedup. The use of three levels decreases the number of coarse model iterations for refinement level 2 from 33.7 to 14.2 in case information from previous time steps is reused.

Concluding, the use of three levels to accelerate the convergence of the coarse model is only beneficial if the computational costs of the coarse model are significant compared to the fine model. For this example, the most inaccurate model with negligible computational costs can already be applied to accelerate the convergence of the fine model to 2.4 iterations per time step showing the robustness of the manifold mapping technique.

Chapter 4.

Conclusions on multi-level acceleration techniques for strongly coupled fluid-structure interaction

> According to the theory of aerodynamics, as may be readily demonstrated through wind tunnel experiments, the bumblebee is unable to fly. This is because the size, weight, and shape of his body in relation to the total wingspread make flying impossible. But the bumblebee, being ignorant of these scientific truths, goes ahead and flies anyway-and makes a little honey every day.
>
> *(Anonymous)*

4.1. Conclusions

A new multi-level coupling algorithm for partitioned simulation of fluid-structure interaction has been proposed based on the manifold mapping algorithm. The main advantage of the manifold mapping algorithm is that it has provable convergence to the solution of the high-fidelity model under mild conditions, contrary to aggressive space mapping which does not always converge to the solution of the high-fidelity model as proven by Echeverria and Hemker [41] and Echeverria et al. [43]. Also, the technique gives complete freedom in the choice of the low-fidelity model. A coarser

mesh can be selected for the low-fidelity model, but also an engineering model can be used in combination with a finite volume or finite element approach for the high-fidelity model. Finally, the method is non-intrusive in the sense that only input-output information of the different models is considered.

The research question was formulated as follows: can manifold mapping be effectively applied to a partitioned fluid-structure interaction simulation, and does its use result in a significant decrease in the number of high fidelity coupling iterations? Numerical experiments showed the high potential of the coupling scheme, which can reduce the number of fine model evaluations of a partitioned fluid-structure interaction simulation by approximately 50 % by transferring most of the work to the coarse model. The convergence of the high-fidelity model is accelerated even further when information from previous time steps is reused. As mentioned, the manifold mapping algorithm has provable convergence to the solution of the high fidelity model.

The use of three levels to accelerate the convergence of the coarse model is only beneficial if the computational costs of the coarse model are significant compared to the fine model. For the examples considered here, the most inaccurate model with negligible computational costs can already be applied to accelerate the convergence of the fine model showing the robustness of the manifold mapping technique.

Judging from the numerous examples we considered, it can be expected that for more complex test cases such as the airflow around a wing in transonic conditions with shocks, similar speedups are observed when compared to the IQN-ILS method or Anderson acceleration technique. The most efficient setup is to use a relative residual convergence measure for the surrogate model, and also reuse information from previous time steps to achieve the lowest number of coupling iterations.

In case a test case with a steady state solution is considered, information from previous time steps cannot be reused. On the contrary, only information from previous iterations is available. Since the approximation changes rapidly during the first iterations, the approximate Jacobian used by the manifold mapping algorithm is not accurate enough for good convergence rates during those initial iterations. Therefore, it is advised to use the identity matrix as the mapping matrix during the initial iterations (i.e

output space mapping), and then enable the use of the mapping matrix T_k.

4.2. Recommendations

Two approaches exist to reach the smallest number of coupling iterations per time step. Specifically, (1) the user specifies the number of time steps of which information is reused, or (2) the mapping matrix T_k is stored in memory and a secant update is used to take into account information from all previous time steps.

The first approach has the advantage that the memory requirements are on the order of the number of interface points. However, the user needs to fine tune the variable which specifies the number of time steps of which information is reused. The second approach has the advantage that the user does not need to specify this parameter, but it results in large memory requirements. The size of the mapping matrix scales with n-squared, with n being the number of interface points.

One possible solution which bypasses both disadvantages is to observe that the mapping matrix has a low rank. Therefore, it can be approximated by a low-rank approximation based on a truncated singular value decomposition. The use of the low-rank approximation results in linear storage requirements for the mapping matrix, and the user does not need to specify the number of time steps of which information is reused. The question remains whether this approach results in a similar number of coupling iterations.

Part II.

Arbitrarily high-order time integration for partitioned fluid-structure interaction simulations using spectral deferred corrections

Chapter 5.

Higher order time integration schemes

> We must use time wisely and forever realize that the time is always ripe to do right.
>
> *(Nelson Mandela)*

We consider the application of arbitrarily high-order time integration using spectral deferred corrections (SDC), integral deferred corrections (IDC) and the Picard Integral Exponential Solver (PIES). The potential of delivering higher temporal accuracy with lower cost than lower order methods has been the incentive in their consideration. Algorithmic details of the solution procedure for spectral deferred corrections, integral deferred corrections, the Picard Integral Exponential Solver, and finally singly-diagonal implicit Runge-Kutta schemes are presented.

Parts of this chapter have been published in

- D. S. Blom et al. "On the application of spectral deferred corrections to incompressible flow on unstructured grids". In: *Journal of Computational Physics* under review (2016);
- D. S. Blom et al. "Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using integral deferred corrections". In: *Journal of Computational Physics* under review (2016).

5.1. Time integration

In this part, we use the method of lines paradigm. In other words, a partial differential equation is first discretized in space and then discretized in time. We study an initial value problem of the form:

$$\frac{\mathrm{d}}{\mathrm{d}t} u\left(t\right) = f\left(u\left(t\right), t\right), \qquad u(0) = u^0, \qquad t \in \left[0, T\right], \tag{5.1}$$

with initial solution u^0 on the time interval $t \in [0, T]$. As time integration schemes, we study spectral deferred corrections, integral deferred corrections, the Picard Integral Exponential Solver, and diagonally-implicit Runge-Kutta methods.

5.2. Spectral deferred corrections

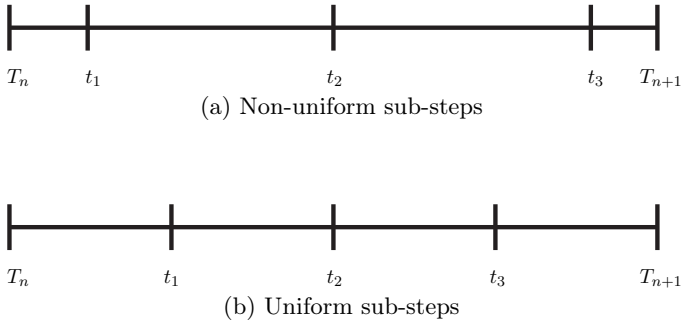
Spectral deferred corrections (SDC) were originally developed for ordinary differential equations (ODE's) by Dutt et al. [40]. Following the initial introduction of the method, an acceleration procedure has been proposed to accelerate the convergence of the SDC sweeps [62]. Also, the method has been applied to problems of reactive flow [20] and gas dynamics [72], among others. Here, we focus on the original formulation of the method as discussed by Dutt et al. [40]. The application of the acceleration procedure is left as future work. The notation of SDC by Speck et al. [92] is used throughout this thesis.

The SDC method starts from a standard initial value problem in integral form:

$$u\left(t\right) = u_0 + \int_0^t f\left(u\left(s\right), s\right) \mathrm{d}s, \tag{5.2}$$

where $t \in [0, T]$, u_0 represents the initial solution, $u\left(t\right) \in \mathbb{R}^N$, and the function $f\left(u\left(t\right), t\right) \in \mathbb{R}^N$. The time $t \in [0, T]$ is subdivided into time steps of size Δt. Each time step is divided further into M sub-steps based on a set of quadrature nodes. For example, Gauss-Lobatto, Gauss-Radau and Clenshaw-Curtis nodes can be used as quadrature nodes. However, uniform nodes are chosen in case of the integral deferred correction method, discussed in Sect. 5.3.

Figure 5.1a shows an example of one time-step with three intermediate



(a) Non-uniform sub-steps

(b) Uniform sub-steps

Figure 5.1.: Schematic of a time step Δt with non-uniform and uniform sub-steps.

quadrature nodes. Depending on the used quadrature nodes, the function value at the left end point T_n is included or not included in the integration in time. Where Gauss-Lobatto and Clenshaw-Curtis nodes do include the left-end point, Gauss-Radau nodes do not include the left-end point. Figure 5.1b gives an example of uniform sub-steps, as is the case for integral deferred corrections.

The collocation polynomial on $[T_n, T_{n+1}]$ is denoted with $u_p(t)$. Thus, we make the approximation $U_j = u_p(t_j) \approx u(t_j)$. The quadrature weights are defined as follows

$$q_{m,j} = \frac{1}{\Delta t} \int_{T_n}^{t_m} l_j(s)\,\mathrm{d}s, \quad m = 0, \ldots, M, \quad j = 0, \ldots, M, \tag{5.3}$$

where $T_n = t_0 < t_1 < \cdots < t_M = T_{n+1}$, and $(l_j)_{j=0,\ldots,M}$ are the Lagrange polynomials. The time step Δt is simply $\Delta t = T_{n+1} - T_n$. An approximation of the solutions $U_m, m = 0, \ldots, M$ is obtained with

$$U_m = u_0 + \Delta t \sum_{j=0}^{M} q_{m,j}\, f(U_j, t_j), \quad m = 0, \ldots, M. \tag{5.4}$$

The integration matrix q is defined to be $(M+1 \times M+1)$ with the matrix entries $q_{m,j}$. The solution vector $\mathbf{U}$ is represented with

$$\mathbf{U} = [U_0, \ldots, U_M], \tag{5.5}$$

and the vector $\mathbf{F}$ containing the function evaluations is defined as

$$\mathbf{F}(\mathbf{U}) = [F_0, \ldots, F_M] = [f(U_0, t_0), \ldots, f(U_M, t_M)]. \tag{5.6}$$

Besides these definitions, the matrix Q is introduced as $Q = q \otimes I_N$ with the identity matrix I and the Kronecker product $\otimes$. The set of equations as shown in (5.4) can now be written more compactly as:

$$\mathbf{U} = \mathbf{U_0} + \Delta t\, Q\, \mathbf{F}(\mathbf{U}, \mathbf{t}), \tag{5.7}$$

with $\mathbf{U_0}$ defined as $\mathbf{U}_0 = U_0 \otimes I_N$.

Equation (5.7) is a fully coupled system consisting of $M \times N$ degrees of freedom. We would like to decouple this system and apply an iterative scheme to achieve the same order of accuracy of Eqn. (5.7), but solve a system of N degrees multiple times instead.

Thus, we introduce the quadrature weights $s_{m,j}$, which apply to node-to-node integration, to decouple the fully coupled system as follows:

$$s_{m,j} = \frac{1}{\Delta t} \int_{t_{m-1}}^{t_m} l_j(s)\, \mathrm{d}s, \quad m = 1, \ldots, M. \tag{5.8}$$

Hence, integrals over $[t_{m-1}, t_m]$ are approximated. The integration matrix s consists of the quadrature weights $s_{m,j}$, and the matrix S is consequently defined as $S = s \otimes I_N$.

The implicit update equation is found as:

$$U_{m+1}^{k+1} = U_m^{k+1} + \Delta t_m \left[f\left(U_{m+1}^{k+1}, t_{m+1}\right) - f\left(U_{m+1}^{k}, t_{m+1}\right) \right] + \Delta t\, Z_m^k, \tag{5.9}$$

which can be interpreted as a backward Euler step with an extra source term:

$$\frac{U_{m+1}^{k+1} - U_m^{k+1}}{\Delta t_m} = f\left(U_{m+1}^{k+1}, t_{m+1}\right) - f\left(U_{m+1}^{k}, t_{m+1}\right) + \frac{\Delta t}{\Delta t_m} Z_m^k. \tag{5.10}$$

Z_m^k is defined as $Z_m^k = S\, \mathbf{F}\left(\mathbf{U}^k\right)$, and $\Delta t_m = t_m - t_{m-1}$. The residual of

the implicit system is simply defined as

$$\mathbf{r}^k_{Euler} = \frac{U^{k+1}_{m+1} - U^{k+1}_m}{\Delta t_m} - f\left(U^{k+1}_{m+1}, t_{m+1}\right) + f\left(U^k_{m+1}, t_{m+1}\right) - \frac{\Delta t}{\Delta t_m} Z^k_m. \tag{5.11}$$

Solving (5.9) at each quadrature node is named an SDC sweep or SDC iteration. One interesting property of the spectral deferred correction method is that each correction sweep increases the order of accuracy of the method, up to the order of accuracy of Eqn. (5.7). This is under the assumption that the implicit systems are solved sufficiently accurate. Also, the use of a first order integrator is assumed, i.e. backward Euler for implicit time stepping. On convergence of the SDC iterations, the method becomes equivalent to the collocation scheme as determined by the quadrature nodes. Hence, the order of convergence of the scheme depends on the chosen quadrature rule, the number of integration points, and the number of SDC sweeps.

Here, the implicit systems with residual (5.11) are solved up to a user specified tolerance τ_{Euler}. Instead of using a fixed number of SDC sweeps, the SDC residual is evaluated:

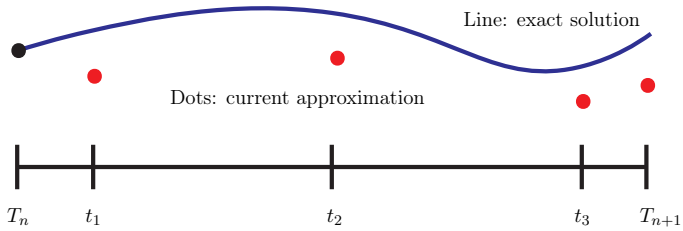
$$\mathbf{r}^k_{SDC} = \mathbf{U}_0 + \Delta t\, Q\, \mathbf{F}\left(\mathbf{U}^k\right) - \mathbf{U}^k. \tag{5.12}$$

Convergence of the SDC sweeps is assumed in case the L_2-norm of the residual scaled by the L_2-norm of $\Delta t\, Q\, \mathbf{F}\left(\mathbf{U}^k\right)$ is smaller than the tolerance τ_{SDC}:

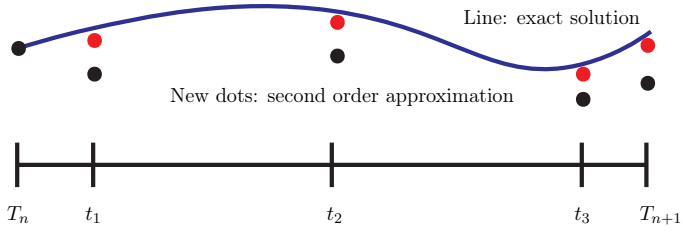
$$\frac{\left|\left|\mathbf{U}_0 + \Delta t\, Q\, \mathbf{F}\left(\mathbf{U}^k\right) - \mathbf{U}^k\right|\right|}{||\Delta t\, Q\, \mathbf{F}\left(\mathbf{U}^k\right)||} \leq \tau_{SDC}. \tag{5.13}$$

Figure 5.2 gives a schematic example of three sweeps. The solid line represents the exact solution which is approximated by the dots at the time instants t_1, t_2, t_3 and T_{n+1}. At each iteration, the accuracy of the approximate solution increases with approximately one order.

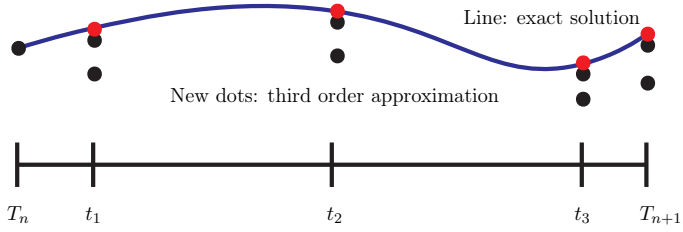
Emmett and Minion [45] have proposed to solve the sub-steps of the spectral deferred correction method in parallel and use a multi-grid solver in time to perform the least amount of iterations or sweeps on the fine level. The coarse grid performs the SDC sweeps in serial, whereas the fine grid solves each sub-step in parallel. The aim of the technique is to achieve



(a) Sweep 1



(b) Sweep 2



(c) Sweep 3

Figure 5.2.: Schematic of SDC sweeps. The red dots are the current approximations of the solution at the times t_1, t_2, t_3 and T_{n+1}. The blue line is the exact solution. The accuracy of the approximate solution increases quickly with each iteration.

better parallel scalability of an already existing code, which uses a domain decomposition to solve the spatial discretization in parallel.

We introduce the notation SDCxy, where x represents the number of quadrature nodes, and y the theoretical order of convergence. For example, SDC23 is a third order method, where one SDC sweep consists of two substeps.

5.3. Integral deferred corrections

Integral deferred corrections are closely related to spectral deferred corrections, namely uniform quadrature nodes are chosen, instead of for example Gauss-Lobatto, Gauss-Radau or Clenshaw-Curtis nodes. Additionally, in [29] it is proven, under mild conditions, that using an r^{th} order Runge-Kutta integrator to solve the update equation, the order of accuracy of the method increases by r orders at every correction sweep. This is not necessarily the case for the spectral deferred correction method [29]. Equation (5.9) is an example where the first-order implicit Euler integrator is used to solve the update equation.

The focus is on quadrature rules that do not depend on the left-hand endpoint, i.e. right-hand quadrature rules. These methods have good stability properties as shown by Hansen and Strain [59] and Layton and Minion [73], and are expected to be good candidates for the simulation of incompressible flows.

The notation IDCx is used throughout this thesis, where x represents the number of quadrature nodes which equals the order of accuracy.

5.4. Picard Integral Exponential Solver (PIES)

The Picard Integral Exponential Solver is closely related to SDC methods [53, 69]. However, its main difference with the SDC method is the assumption that the solution can be accurately represented by a combination of complex exponential functions. The selection of the exponential functions is based on a matrix skeletonization step and the solution of a linear least squares system. Highly accurate quadrature weights are computed on uniform quadrature nodes and can be determined off-line or once during every simulation. A short description of the method is given here.

For further details, the reader is referred to Glaser and Rokhlin [53] and Kushnir and Rokhlin [69].

First, the solution of an initial value problem is approximated by the summation of complex valued exponential functions:

$$u(t) \approx \sum_{j=1}^{n} \alpha_j \, e^{\lambda_j \, t}, \tag{5.14}$$

with the coefficients α_j and a-priori chosen complex numbers λ_j. A set $\lambda_1, \ldots, \lambda_n$ is chosen based on a numerical scheme called matrix skeletonization [27] (applied on (5.17)). When $\lambda_1, \ldots, \lambda_n$ are known, the integration coefficients ω for the numerical integration are computed based on a least squares approach. These coefficients are comparable to the coefficients $q_{m,j}$ of the spectral deferred correction method.

Furthermore, the set $\lambda_1, \ldots, \lambda_n$ is selected for which λ lies in the complex semi-disk S_ρ

$$S_\rho = \{\lambda \in \mathbb{C} | \mathrm{Re}(\lambda) \leq 0, |\lambda| \leq \rho\}, \tag{5.15}$$

with radius ρ. With Lemma 13 by Glaser and Rokhlin [53], it is concluded that it is sufficient to only discretize the boundary of the semi-disk S_ρ. Exponentials of the form $e^{\lambda t}$ up to precision δ are selected such that

$$\left| e^{\lambda t} - \sum_{i=1}^{n} c_i \, e^{\lambda_i \, t} \right| < \delta \qquad t \in [-1, 1], \tag{5.16}$$

with coefficients c_i.

Assume that $\lambda_1, \ldots, \lambda_N$ and $t_1, \ldots, t_M$ are sufficiently dense uniform discretization of the semi-disk S_ρ, and the time interval $[-1, 1]$. Then, the subset $\lambda_1, \ldots, \lambda_n$ is selected based on the skeleton of the matrix B

$$B = \begin{pmatrix} e^{\lambda_1 \, t_1} & e^{\lambda_2 \, t_1} & \ldots & e^{\lambda_N \, t_1} \\ e^{\lambda_1 \, t_2} & e^{\lambda_2 \, t_2} & \ldots & e^{\lambda_N \, t_2} \\ \vdots & \vdots & \ddots & \vdots \\ e^{\lambda_1 \, t_M} & e^{\lambda_2 \, t_M} & \ldots & e^{\lambda_N \, t_M} \end{pmatrix} \tag{5.17}$$

with an error term smaller than δ, which can be computed with a QR-decomposition. This results in n columns with indices $i_1, \ldots i_n$. The se-

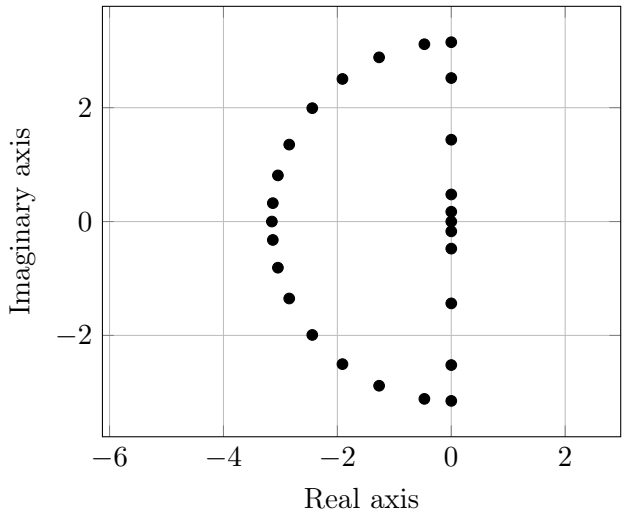


Figure 5.3.: Example of selection of complex numbers $\lambda_1, \ldots, \lambda_n$ on the boundary of the semi-disk with radius 3.15 and $\delta = 1 \cdot 10^{-12}$

lected complex numbers $\lambda_1, \ldots, \lambda_n$ span $\left\{e^{\lambda t} | \lambda \in \delta S_\rho\right\}$ on the boundary of the semi-disk up to precision δ and are therefore a suitable representation of $\left\{e^{\lambda t} | \lambda \in S_\rho\right\}$.

An example of the selected complex numbers $\lambda_1, \ldots, \lambda_n$ is shown in Fig. 5.3. The radius of the semi-circle ρ is set to 3.15 as suggested by Glaser and Rokhlin [53]. The semi-disk is discretized with 800 points, and the selection criterion δ is set to $1 \cdot 10^{-12}$. Another example is given in Fig. 5.4 where the selection criterion δ is set to $1 \cdot 10^{-5}$. As can be seen, a substantially smaller set complex numbers $\lambda_1, \ldots, \lambda_n$ is selected.

Similarly to spectral deferred correction methods, the numerical integration of one time-step can be evaluated by the quadrature rule

$$\int_{-1}^{t_j} u(s)\, \mathrm{d}s \approx \sum_{i=1}^{k} u(t_i)\, \omega_{ij}, \tag{5.18}$$

with the real valued weights $\omega_{ij} \in \mathbb{R}$. Note that the weight matrix $\omega =$

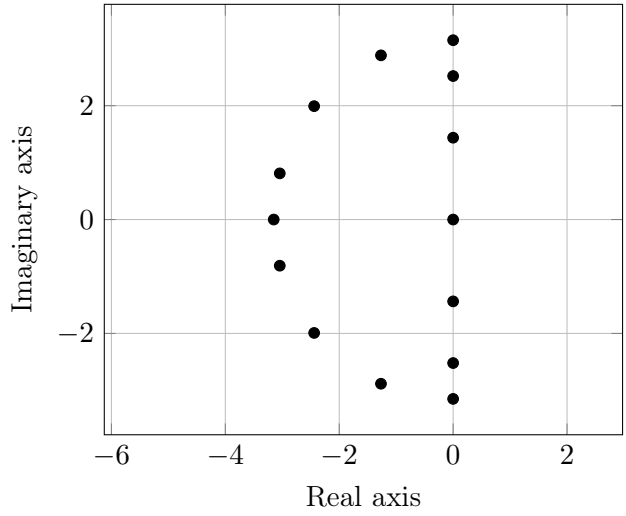


Figure 5.4.: Example of selection of complex numbers $\lambda_1, \ldots, \lambda_n$ on the boundary of the semi-disk with radius 3.15 and $\delta = 1 \cdot 10^{-5}$

$(\omega_{ij}) \in \mathcal{R}^{n \times k}$ has to satisfy the linear system

$$A\,\omega = b, \tag{5.19}$$

with the complex valued matrix $A \in \mathcal{C}^{n \times k}$

$$A = \begin{pmatrix} e^{\lambda_1 t_1} & e^{\lambda_1 t_2} & \ldots & e^{\lambda_1 t_k} \\ e^{\lambda_2 t_1} & e^{\lambda_2 t_2} & \ldots & e^{\lambda_2 t_k} \\ \vdots & \vdots & \ddots & \vdots \\ e^{\lambda_n t_1} & e^{\lambda_n t_2} & \ldots & e^{\lambda_n t_k} \end{pmatrix}, \tag{5.20}$$

and the complex valued matrix $b \in \mathcal{C}^{n \times k}$

$$b = \begin{pmatrix} \int_{-1}^{t_1} e^{\lambda_1 s} \mathrm{d}s & \int_{-1}^{t_2} e^{\lambda_1 s} \mathrm{d}s & \cdots & \int_{-1}^{t_k} e^{\lambda_1 s} \mathrm{d}s \\ \int_{-1}^{t_1} e^{\lambda_2 s} \mathrm{d}s & \int_{-1}^{t_2} e^{\lambda_2 s} \mathrm{d}s & \cdots & \int_{-1}^{t_k} e^{\lambda_2 s} \mathrm{d}s \\ \vdots & \vdots & \ddots & \vdots \\ \int_{-1}^{t_1} e^{\lambda_n s} \mathrm{d}s & \int_{-1}^{t_2} e^{\lambda_n s} \mathrm{d}s & \cdots & \int_{-1}^{t_k} e^{\lambda_n s} \mathrm{d}s \end{pmatrix}. \tag{5.21}$$

We use a singular value decomposition to solve the linear system (5.19) for the coefficients ω_{ij}, which have the same function as the coefficients $q_{m,j}$ of the spectral deferred correction method (see Eqn. (5.3)). The node-to-node integration coefficients $\omega_{s_{ij}}$ can be computed with a different definition of the matrix b_s:

$$b_s = \begin{pmatrix} 0 & \int_{t_1}^{t_2} e^{\lambda_1 s}\mathrm{d}s & \cdots & \int_{t_{k-1}}^{t_k} e^{\lambda_1 s}\mathrm{d}s \\ 0 & \int_{t_1}^{t_2} e^{\lambda_2 s}\mathrm{d}s & \cdots & \int_{t_{k-1}}^{t_k} e^{\lambda_2 s}\mathrm{d}s \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \int_{t_1}^{t_2} e^{\lambda_n s}\mathrm{d}s & \cdots & \int_{t_{k-1}}^{t_k} e^{\lambda_n s}\mathrm{d}s \end{pmatrix}. \tag{5.22}$$

The rest of the procedure is the same as the spectral deferred correction method. So, the update equation (5.9) is now solved with the newly obtained coefficients $\omega_{s_{ij}}$.

Kushnir and Rokhlin [69] show that when the left-hand endpoint t_n is excluded, an L-stable method is found. Since uniform sub-steps are assumed, the convergence rate of the non-linear solver will not differ significantly from one sub-step to the next, just as the IDC technique. We use the notation PIESx, where x denotes the number of sweeps. This is equal to the expected order of accuracy under the assumption that the systems solved accurately enough. Unless explicitly stated otherwise, δ is set to $\delta = 1 \cdot 10^{-13}$.

Summarizing, the procedure to determine the sub-steps and coefficients with the Picard Integral Exponential Solver is as follows:

1. Create a fine discretization of the boundary of the semi-disk S_ρ with radius ρ;

2. Create a fine discretization of the time interval $t \in [\,-1, 1\,]$;

3. Compute the matrix B as shown in Eqn. (5.17), copied here for completeness:

$$B = \begin{pmatrix} e^{\lambda_1 t_1} & e^{\lambda_2 t_1} & \cdots & e^{\lambda_N t_1} \\ e^{\lambda_1 t_2} & e^{\lambda_2 t_2} & \cdots & e^{\lambda_N t_2} \\ \vdots & \vdots & \ddots & \vdots \\ e^{\lambda_1 t_M} & e^{\lambda_2 t_M} & \cdots & e^{\lambda_N t_M} \end{pmatrix}, \tag{5.23}$$

Table 5.1.: Butcher tableau of a SDIRK method

c_1	γ	0	0	0
c_2	a_{21}	γ	0	0
$\vdots$	$\vdots$	$\ddots$	$\ddots$	0
c_s	a_{s1}	$\dots$	a_{ss-1}	γ
	a_{s1}	$\dots$	a_{ss-1}	a_{ss}

and perform the skeletonization process up to precision δ to select the subset $\lambda_1, \dots, \lambda_n$;

4. Then, solve for the node-to-note coefficients ω_s as in Eqn. (5.19) by using the definitions of A and b_s in Eqn. (5.20) and (5.22), respectively. Thereafter, the coefficients can be divided by two in order to scale the coefficients to the unit time interval $t \in [\,0, 1\,]$.

In Table A.1 the coefficients are given for the node-to-node integration with $\rho = 3.15$ and $\delta = 1 \cdot 10^{-2}$. Coefficients for schemes with smaller values of δ are not shown, due to a large number of coefficients.

5.5. Implicit Runge-Kutta

We consider the singly-diagonally-implicit Runge-Kutta (SDIRK) method, which subdivides a time step into s stages. SDIRK schemes are of the form

$$k_i = f\left(u_n + \Delta t \sum_{j=1}^{s} a_{ij}\, k_j, t_n + c_i\, \Delta t\right), \quad i, \dots, s, \tag{5.24}$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{s} b_i\, k_i, \tag{5.25}$$

where the coefficients a_{ij}, b_i, c_i depend on the chosen scheme, and are chosen such that some order conditions are satisfied in order to reach the specified order of accuracy. Also, the coefficients can be found in a Butcher tableau, as shown in Table 5.1.

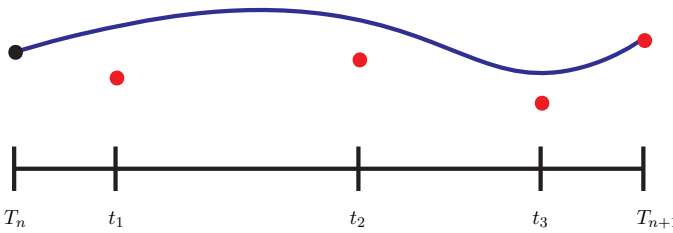


Figure 5.5.: Schematic of one time step solved with a SDIRK scheme. The red dots are the current approximations of the solution at the times t_1, t_2, t_3 and T_{n+1}. The blue line is the exact solution. Only the final solution at T_{n+1} has a high order of accuracy.

Here, we only consider stiffly-accurate SDIRK schemes, i.e. the last row of the Butcher tableau is identical to $\mathbf{b}^T$ which is advantageous in solving stiff problems [58]. In other words, the solution at the next time step equals to the last sub-step solution. Just as the integral deferred correction schemes, each sub-step or stage corresponds to solving an implicit Euler time step with an extra source term. However, it is a direct method contrasting the IDC technique. Thus, when all the stages are solved, the solution at the next time step is found. The scheme moves then to the next time step. A schematic of one time step solved with the SDIRK scheme is shown in Fig. 5.5.

The explicit-singly-diagonally-implicit Runge-Kutta (ESDIRK) method is closely related the SDIRK method. The a_{11} entry of the Butcher tableau equals zero for an ESDIRK scheme, resulting in an explicit first stage. This is in contrast to the SDIRK method for which every stage is implicit. Therefore, ESDIRK schemes need the function value f at the start of the time interval. Since the first stage is explicit, the implicit stages can be constructed to be second order accurate instead of first order accurate for the SDIRK scheme.

The Butcher tableau's of the considered time integration schemes can be found in: Ellsiepen [44] (SDIRK2), Cash [25] (SDIRK3, SDIRK4), Kennedy and Carpenter [67] (ESDIRK3, ESDIRK4, ESDIRK5), Rang [84] (SDIRK2PR) and Rang [83] (ESDIRK53PR, ESDIRK63PR, ESDIRK74PR). For convenience and fast look up, the Butcher tableau's are also included

in Appendix B and C.

This concludes the description of the time integration methods. The finite volume method used to discretize the Navier-Stokes equations is discussed in the following chapter.

Chapter 6.

Spatial and temporal discretization of the incompressible Navier-Stokes equations

> The fundamental laws necessary for the mathematical treatment of a large part of physics and the whole of chemistry are thus completely known, and the difficulty lies only in the fact that application of these laws leads to equations that are too complex to be solved.
>
> *(Paul Dirac)*

Where the previous chapter focused on several higher order time integration schemes, here we show the solution procedure of the incompressible Navier-Stokes equations. Details of the numerical method are shown which preserves the theoretical orders of accuracy in time, including the appropriate treatment of the velocity flux (time-consistent Rhie-Chow interpolation), and use of an iterative time advancing algorithm (iterated PISO) to minimize the influence of iterative errors on temporal order.

Parts of this chapter have been published in D. S. Blom et al. "Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using integral deferred corrections". In: *Journal of Computational Physics* under review (2016).

6.1. Introduction

We consider the incompressible Navier-Stokes equations, which consist of the momentum and continuity equations:

$$\dot{\mathbf{u}} + \nabla\mathbf{u}\cdot\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = 0, \tag{6.1}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{6.2}$$

$\mathbf{u}$ and p denote the velocity and pressure, and ν represents the kinematic viscosity. We use the iterated PISO algorithm [63] to solve the Navier-Stokes equations with a finite volume method on unstructured grids. Also, the momentum interpolation of Lien and Leschziner [74] (later also published in Yu et al. [100]) is used in order to obtain a temporally consistent formulation. First, the solution procedure for static meshes is discussed. After that, the changes necessary for simulations on deforming meshes are shown.

6.2. Static grids

By applying the finite volume method, the momentum equation per control volume is:

$$\frac{\mathrm{d}}{\mathrm{d}\,t}\int_{V_C}\mathbf{u}\,\mathrm{d}V + \oint_{S_C}\mathbf{n}\cdot\mathbf{u}\,\mathbf{u}\,\mathrm{d}S - \int_{V_C}\nabla\cdot(\nu\nabla\mathbf{u})\,\mathrm{d}V = -\int_{V_C}\nabla p\,\mathrm{d}V. \tag{6.3}$$

We introduce a control volume V_C, cell surface area S_C, and $\mathbf{n}$ as the normal vector to the cell surface. The discrete form of the continuity equation is written as

$$\sum_f \mathbf{u}_f\cdot\mathbf{n}_f S_f = 0, \tag{6.4}$$

where we introduce the velocity $\mathbf{u}_f$ which is defined on the face centers, and can be used to compute the mass flux through the face with $\phi_f = \mathbf{u}_f\cdot\mathbf{n}_f S_f$. The subscript f defines a variable which is defined on the face center. The discrete form of the momentum equation is given as

$$V\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \sum_f \phi\,\mathbf{u} + \nu\sum_f(\nabla\mathbf{u})_f\cdot\mathbf{n}_f S_f = -\sum_f(p)_f\mathbf{n}_f S_f. \tag{6.5}$$

After dividing by the volume V, and choosing appropriate schemes for the spatial operators, Eqn. (6.5) can be written as

$$\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}t} + a\,\mathbf{u} + B\,\mathbf{u} = \mathbf{q} - \nabla p, \tag{6.6}$$

with the diagonal a and off diagonal B coefficients of the discretization matrix. $\mathbf{q}$ consists of the remaining source terms. The $\mathbf{H}$ operator is defined in the following way:

$$\mathbf{H}\left(\mathbf{u}\right) = \mathbf{q} - B\,\mathbf{u}. \tag{6.7}$$

Note that the volume V can only be assumed to be constant for static grids, therefore Eqn. (6.6) is not valid for moving grids.

To obtain a method which is consistent in time, the $\mathbf{H}$ operator needs to contain contributions only from the spatial discretization. Hence, the momentum equation can be simplified to

$$\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}t} + a\,\mathbf{u} = \mathbf{H}\left(\mathbf{u}\right) - \nabla p. \tag{6.8}$$

By applying implicit Euler time integration we find

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\Delta t} + a\,\mathbf{u}^{n+1} = \mathbf{H}\left(\mathbf{u}^{n+1}\right) - \nabla p^{n+1}. \tag{6.9}$$

Hence, the residual vector $\mathcal{F}$ is defined as follows:

$$\mathcal{F} = -a\,\mathbf{u}^{n+1} + \mathbf{H}\left(\mathbf{u}^{n+1}\right) - \nabla p^{n+1}. \tag{6.10}$$

Finally, we arrive at the following initial value problem which can be integrated in time using one of the chosen time integration schemes:

$$\frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}t} = \mathcal{F}\left(\mathbf{u}, p, t\right). \tag{6.11}$$

6.3. Face-velocity interpolation

To avoid pressure-velocity decoupling, we follow the solution algorithm of Rhie and Chow [85]. Thus, a fully discretized momentum equation for the

face velocities is applied. The method of Rhie and Chow [85] does not result in a consistent time integration. Thus the momentum interpolation of Lien and Leschziner [74] is used to obtain a temporally consistent formulation.

Equivalent to the cell centered formulation of the momentum equation (see Eqn. 6.8), we can write the momentum equation for the face centers as follows:

$$\frac{\mathrm{d}\,\mathbf{u}_f}{\mathrm{d}t} + \overline{a}\,\mathbf{u}_f = \overline{\mathbf{H}}_f - (\nabla p)_f, \tag{6.12}$$

with $(\nabla p)_f$ as the face pressure gradient normal to the surface. The notation $\overline{\mathbf{H}}$ and $\overline{a}$ indicates a linear interpolation from the cell centers to the cell face of the $\mathbf{H}$ and a operators. After discretization in time with the implicit Euler method we find

$$\frac{\mathbf{u}_f^{n+1} - \mathbf{u}_f^n}{\Delta t} + \overline{a}\,\mathbf{u}_f^{n+1} = \overline{\mathbf{H}}_f - \left(\nabla p^{n+1}\right)_f. \tag{6.13}$$

Equation (6.13) is slightly rewritten to find the face velocity

$$\mathbf{u}_f^{n+1} = \frac{1}{1 + \Delta t\,\overline{a}} \left(\Delta t\,\overline{\mathbf{H}}_f - \Delta t \left(\nabla p^{n+1}\right)_f + \mathbf{u}_f^n\right). \tag{6.14}$$

After recalling the continuity equation

$$\nabla \cdot \mathbf{u} = \sum_f \mathbf{u}_f \cdot \mathbf{n}_f S_f = \sum_f \phi_f = 0, \tag{6.15}$$

Equation (6.14) is substituted into the continuity equation, which yields the Laplace equation for the pressure:

$$\sum_f \left(\frac{\Delta t \left(\nabla p^{n+1}\right)_f}{1 + \Delta t\,\overline{a}}\right) \cdot \mathbf{n}_f S_f = \sum_f \left(\frac{\Delta t\,\overline{\mathbf{H}}_f + \mathbf{u}_f^n}{1 + \Delta t\,\overline{a}}\right) \cdot \mathbf{n}_f S_f. \tag{6.16}$$

6.4. Solution algorithm

Since a segregated solution procedure is used to find the velocity and pressure, and the nonlinear convection term is treated implicitly, an iterative solution procedure is applied. Thus, for each sub-step of the integral deferred correction scheme, or for each stage of the SDIRK method, a

backward Euler step with extra source term $\mathbf{q}_t$ is solved in the following manner.

1. Solve the discretized momentum equation for the velocity vector $\mathbf{u}_{r+1}^{n+1}$ with outer iteration counter r:

$$\frac{\mathbf{u}_{r+1}^{n+1} - \mathbf{u}^n}{\Delta t} + a\,\mathbf{u}_{r+1}^{n+1} = \mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right) - \nabla p_r + \mathbf{q}_t. \tag{6.17}$$

   The pressure of the previous time step is used as initial guess for ∇p_r.

2. Calculate the intermediate mass flux with

$$\phi_f^\star = \left(\frac{\Delta t\, \overline{\mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right)}_f + \mathbf{u}_f^n + \Delta t\, \mathbf{q}_{tf}}{1 + \Delta t\, \overline{a}} \right) \cdot \mathbf{n}_f S_f. \tag{6.18}$$

   Note the linear interpolation from the cell centers to the cell faces for $\overline{\mathbf{H}}_f$ and $\overline{a}$, as indicated by the overline notation $\overline{}$. The term $\mathbf{q}_{tf}$ is the higher order source term of the time derivative of the face velocity.

3. Solve for the pressure at the next iteration $r+1$ with

$$\sum_f \left(\frac{\Delta t \left(\nabla p_{r+1}^{n+1}\right)_f}{1 + \Delta t\, \overline{a}} \right) \cdot \mathbf{n}_f S_f = \sum_f \phi_f^\star. \tag{6.19}$$

4. Correct the flux and velocities defined in the cell centers and face centers as follows:

$$\begin{aligned} \mathbf{u}_{r+1}^{n+1} &= \frac{\Delta t\, \mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right) - \Delta t \left(\nabla p_{r+1}^{n+1}\right) + \mathbf{u}^n + \Delta t\, \mathbf{q}_t}{1 + \Delta t\, a}, \\ \mathbf{u}_{r+1_f}^{n+1} &= \frac{\Delta t\, \overline{\mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right)}_f - \Delta t \left(\nabla p_{r+1}^{n+1}\right)_f + \mathbf{u}_f^n + \Delta t\, \mathbf{q}_{tf}}{1 + \Delta t\, \overline{a}}, \\ \phi_f &= \mathbf{u}_f \cdot \mathbf{n}_f S_f. \end{aligned} \tag{6.20}$$

We use the non-linear residual defined on the cell centers to monitor the convergence of the outer iterations:

$$\mathbf{R} = \frac{\mathbf{u}_{r+1}^{n+1} - \mathbf{u}^n}{\Delta t} + a\,\mathbf{u}_{r+1}^{n+1} - \mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right) + \nabla p_{r+1} - \mathbf{q}_t. \tag{6.21}$$

When the norm of the non-linear residual falls below a certain threshold, the procedure moves to the next sub step or time step. Otherwise, the algorithm restarts at step 1 with updated velocity and pressure solutions.

6.5. Moving grids

For deforming domains, the Arbitrary Lagrangian Eulerian (ALE) framework is applied onto the incompressible Navier-Stokes equations. Thus, the discretized momentum equation reads as follows:

$$\frac{\mathrm{d}\,\mathbf{u}\,V}{\mathrm{d}t} + \sum_f \left(\phi - \phi_{mesh}\right)\mathbf{u} + \nu \sum_f \left(\nabla \mathbf{u}\right)_f \cdot \mathbf{n}_f S_f = -\sum_f p_f \mathbf{n}_f S_f, \tag{6.22}$$

with the mesh velocity flux ϕ_{mesh}, and the inclusion of the volumes in the time derivative. After discretizing in space, and applying the implicit Euler time integration scheme, the following form of the momentum equation is found:

$$\frac{\mathbf{u}^{n+1}\,V^{n+1} - \mathbf{u}^n\,V^n}{\Delta t} + a\,\mathbf{u}^{n+1}\,V^{n+1} = \mathbf{H}\left(\mathbf{u}^{n+1}\right)\,V^{n+1} - \sum_f p_f\,\mathbf{n}_f^{n+1} S_f^{n+1}. \tag{6.23}$$

Note that the equation is not divided by the volume V, due to the fact that the volumes are not constant during the simulation. The residual vector $\mathcal{F}$ follows as:

$$\mathcal{F} = -a\,\mathbf{u}^{n+1}\,V^{n+1} + \mathbf{H}\left(\mathbf{u}^{n+1}\right)\,V^{n+1} - \sum_f p_f\,\mathbf{n}_f^{n+1} S_f^{n+1}. \tag{6.24}$$

Hence, the following initial value problem is found:

$$\frac{\mathrm{d}\,\mathbf{u}\,V}{\mathrm{d}t} = \mathcal{F}\left(\mathbf{u}, p, t, V\right). \tag{6.25}$$

The momentum equation for the face center velocity $\mathbf{u}_f$ also includes the contributions of the changing volumes:

$$\frac{\mathbf{u}_f^{n+1}\,\overline{V^{n+1}} - \mathbf{u}_f^{n}\,\overline{V^{n}}}{\Delta t} + \overline{a}\,\mathbf{u}_f^{n+1}\,\overline{V^{n+1}} = \overline{\mathbf{H}}_f\,\overline{V^{n+1}} - \sum_f p_f\,\mathbf{n}_f^{n+1} S_f^{n+1}. \quad (6.26)$$

After solving for the face center velocity $\mathbf{u}_f$ we find:

$$\mathbf{u}_f^{n+1} = \frac{1}{1+\Delta t\,\overline{a}}\left(\Delta t\,\overline{\mathbf{H}}_f - \Delta t\left(\nabla p^{n+1}\right)_f + \mathbf{u}_f^{n}\frac{\overline{V^{n}}}{\overline{V^{n+1}}}\right), \quad (6.27)$$

which is sufficient to implement a time consistent solver for the incompressible Navier-Stokes on moving meshes.

For completeness, the solution algorithm is laid out here including higher order source terms $\mathbf{q}_t$ and $\mathbf{q}_{tf}$. These source terms have the unit $\mathrm{m}^4/\mathrm{s}^2$, as it is an acceleration term multiplied by a volume term.

1. Solve the discretized momentum equation for the velocity vector $\mathbf{u}_{r+1}^{n+1}$ at iteration $r+1$:

$$\frac{\mathbf{u}_{r+1}^{n+1}\,V^{n+1} - \mathbf{u}_{r+1}^{n}\,V^{n+1}}{\Delta t} + a\,\mathbf{u}^{n+1}\,V^{n+1} = \mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right)\,V^{n+1} - \sum_f p_r\,\mathbf{n}_f^{n+1} S_f^{n+1} + \mathbf{q}_t. \quad (6.28)$$

   As mentioned before, the pressure of the previous time step is used as initial guess for $\sum_f p_r\,\mathbf{n}_f^{n+1} S_f^{n+1}$, i.e. $p_0^{n+1} = p^n$.

2. Calculate the intermediate mass flux $\phi_f^\star$ by taking the change in volume into account:

$$\phi_f^\star = \left(\frac{\Delta t\,\overline{\mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right)}_f + \mathbf{u}_f^{n}\frac{\overline{V^{n}}}{\overline{V^{n+1}}} + \frac{\Delta t\,\mathbf{q}_{tf}}{\overline{V^{n+1}}}}{1+\Delta t\,\overline{a}}\right)\cdot \mathbf{n}_f^{n+1} S_f^{n+1}. \quad (6.29)$$

   Linear interpolation of the volumes from the cell centers to the face centers is necessary to take the change in volume into account. Note

that the volumes are first interpolated to the faces, and thereafter divided: $\frac{\overline{V^n}}{V^{n+1}}$ instead of $\overline{\left(\frac{V^n}{V^{n+1}}\right)}$.

3. Solve for the pressure with

$$\sum_f \left(\frac{\Delta t}{1 + \Delta t\, \overline{a}} (\nabla p_{r+1})^{n+1} \right) \cdot \mathbf{n}_f^{n+1} S_f^{n+1} = \sum_f \phi_f^\star. \tag{6.30}$$

4. Correct the flux and velocities defined in the cell centers and face centers as follows:

$$\begin{aligned} \mathbf{u}_{r+1}^{n+1} &= \frac{\Delta t\, \mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right) - \Delta t\left(\nabla p_{r+1}^{n+1}\right) + \mathbf{u}^n \frac{V^n}{V^{n+1}} + \frac{\Delta t\, \mathbf{q}_t}{V^{n+1}}}{1 + \Delta t\, a}, \\ \mathbf{u}_{r+1_f}^{n+1} &= \frac{\Delta t\, \overline{\mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right)}_f - \Delta t\left(\nabla p_{r+1}^{n+1}\right)_f + \mathbf{u}_f^n \frac{\overline{V^n}}{\overline{V^{n+1}}} + \frac{\Delta t\, \mathbf{q}_{t\,f}}{\overline{V^{n+1}}}}{1 + \Delta t\, \overline{a}}, \\ \phi_f &= \mathbf{u}_f \cdot \mathbf{n}_f S_f. \end{aligned} \tag{6.31}$$

Similar to the static grid case, we use the nonlinear residual as defined on the cell centers to monitor the convergence of the outer iterations:

$$\begin{aligned} \mathbf{R} = &\frac{\mathbf{u}_{r+1}^{n+1} V^{n+1} - \mathbf{u}_{r+1}^{n} V^{n+1}}{\Delta t} + a\, \mathbf{u}^{n+1} V^{n+1} \\ &- \mathbf{H}\left(\mathbf{u}_{r+1}^{n+1}\right) V^{n+1} + \sum_f p_{r+1}\, \mathbf{n}_f^{n+1} S_f^{n+1} - \mathbf{q}_t. \end{aligned} \tag{6.32}$$

When the norm of the non-linear residual falls below a certain threshold, the procedure moves to the next sub step or time step. Otherwise, the algorithm performs step 1 through 4 again with updated velocity and pressure solutions.

6.6. Higher order time integration

So far, we only considered the implicit Euler scheme (or only included the source term $\mathbf{q}_t$ in the formulation), which is first order accurate. To obtain a higher order approximation in time, we apply spectral deferred corrections, the integral deferred correction scheme, the Picard Integral

Exponential Solver, and the diagonally implicit Runge-Kutta method. The time derivative occurs in the cell centered momentum equation, i.e. face centered momentum equation, and in the Laplace equation for the pressure.

When the spectral or integral deferred correction scheme is applied to the cell centered momentum equation, we find the following equation:

$$\begin{aligned}\frac{\mathbf{u}_{m+1}^{k+1}\, V_{m+1}^{k+1} - \mathbf{u}_{m}^{k+1}\, V_{m}^{k+1}}{\Delta t} &= \mathcal{F}\left(u_{m+1}^{k+1}, p_{m+1}^{k+1}, V_{m+1}^{k+1}, t_{m+1}\right) \\ &\quad - \mathcal{F}\left(U_{m+1}^{k}, p_{m+1}^{k}, V_{m+1}^{k}, t_{m+1}\right) + \frac{\Delta t}{\Delta t_m}\, Z_m^k.\end{aligned} \tag{6.33}$$

The residual vector $\mathcal{F}$ is defined as in Eqn. (6.24). The equations for the face centered velocity, and the Laplace equation for the pressure follow in a similar fashion.

The solution procedure for the Navier-Stokes equations used in this thesis only results in an explicit function operator for the cell centered terms, but not for the face value terms. Also, the function value $\mathcal{F}$ cannot be evaluated straightforward at the left end point (start of the time interval) for the Discrete Geometric Conservation Law (see Sect. 6.7). Therefore, the left end point is not included in the quadrature nodes to not further complicate the implementation of the complete method.

Thus, at the end of each implicit solve, the function $\mathcal{F}$ is determined by rewriting the update Equation (6.33) instead of performing a separate function call:

$$\begin{aligned}\mathcal{F}\left(u_{m+1}^{k+1}, p_{m+1}^{k+1}, V_{m+1}^{k+1}, t_{m+1}\right) &= \frac{\mathbf{u}_{m+1}^{k+1}\, V_{m+1}^{k+1} - \mathbf{u}_{m}^{k+1}\, V_{m}^{k+1}}{\Delta t} \\ &\quad + \mathcal{F}\left(U_{m+1}^{k}, p_{m+1}^{k}, V_{m+1}^{k}, t_{m+1}\right) - \frac{\Delta t}{\Delta t_m}\, Z_m^k.\end{aligned} \tag{6.34}$$

6.7. Discrete Geometric Conservation Law

The mesh fluxes ϕ_m are determined with the Discrete Geometric Conservation Law (DGCL), and are consequently based on the swept volumes. Violation of the DGCL results in spurious oscillations due to additional

sources and sinks of mass [39]. The DGCL states that the change in volume of a moving control volume must be equal to the sum of the total volume swept by the faces that enclose the volume. When applying the backward Euler (BE) method, we find the mesh fluxes ϕ_{mesh} as follows:

$$\phi_{mesh,\,BE} = \frac{\Delta V_f^{n+1}}{\Delta t}, \tag{6.35}$$

with the swept volumes of a face ΔV_f. When applying the integral deferred correction scheme, we find:

$$\phi_{mesh,\,IDC} = f_{\Delta V}\left(\Delta V_{m+1}^{k+1}, t_{m+1}\right) - f_{\Delta V}\left(\Delta V_{m+1}^{k}, t_{m+1}\right) + \frac{\Delta t}{\Delta t_m} Z_m^k, \tag{6.36}$$

with the function $f_{\Delta V}$ defined as

$$f_{\Delta V} = \frac{\Delta V_f}{\Delta t}. \tag{6.37}$$

6.8. Boundary conditions at the fluid-structure interface

At the fluid-structure interface $\Gamma^{f\,s}$, the balance of stresses (i.e. dynamic boundary condition) is enforced through

$$\boldsymbol{\sigma}^f \boldsymbol{n} = \boldsymbol{\sigma}^s \boldsymbol{n} \qquad \text{on } \Gamma^{f\,s}, \tag{6.38}$$

with the unit vector $\boldsymbol{n}$ normal to the fluid-structure interface $\Gamma^{f\,s}$, and the stress tensors $\boldsymbol{\sigma}^f$ and $\boldsymbol{\sigma}^s$. Also, at the fluid-structure interface the velocities $\boldsymbol{v}^f$ and $\boldsymbol{v}^s$ must be equal (i.e. kinematic boundary condition):

$$\boldsymbol{v}^f = \boldsymbol{v}^s \qquad \text{on } \Gamma^{f\,s}. \tag{6.39}$$

6.9. Integral deferred corrections applied to fluid-structure interaction simulations

We consider the partitioned fluid-structure interaction setup because highly specialized solvers have already been developed for fluid as well as structure

simulations. It is desired to reuse these applications for fluid-structure interaction computations where separate fluid and structure solvers need to be coupled. We shortly discuss the used coupling method and convergence measures which should be used when using an integral deferred correction method.

6.9.1. Anderson acceleration combined with higher order time integration

Chapter 2 discusses the Anderson acceleration method, which is used in this part as well to accelerate the convergence of the sub-iterations within one sub-step. Note that special care has to be taken when combining the Anderson acceleration method with a high-order time integration method. With uniform sub-steps, information from previous sub-steps and previous time steps can easily be reused to speed up the convergence of the coupling iterations in the current sub-step.

However, in the case of spectral deferred corrections or the SDIRK method, non-uniform sub-steps are applied. In this case, it is necessary to scale the values of the interface Jacobian by the used sub-step. Otherwise, the change in sub-step size will be incorporated in the approximation of the interface Jacobian possibly increasing the number of coupling iterations instead of accelerating the method.

6.9.2. Convergence measures for partitioned fluid-structure interaction

As mentioned in Chap. 2, a standard way to measure the convergence of the coupling iterations is to use a relative convergence criterion based on input and output information with a tolerance τ:

$$\frac{||\boldsymbol{R}(\boldsymbol{x})||_2}{||\boldsymbol{x}||_2 + \varepsilon} \leq \tau_f, \tag{6.40}$$

with $\varepsilon = \sqrt{2^{-52}} = 2^{-26}$ when computing with double precision floats, and the fluid-structure interface residual $\boldsymbol{R}$. This is the preferred way to measure convergence for the SDIRK time integration scheme or backward differencing. However, in case of the integral deferred correction method,

each sweep increases the accuracy of the solution, and it is not necessary to solve the FSI problem up to a strict tolerance τ. Following the ideas of Speck et al. [93], the norm of the residual is scaled by the norm of the first residual of each sub step and each correction sweep:

$$\frac{||\boldsymbol{R}(\boldsymbol{x})||_2}{||\boldsymbol{R}_0||_2 + \varepsilon} \leq \tau. \tag{6.41}$$

Now, at each sweep the norm of $\boldsymbol{R}_0$ decreases, and convergence is assumed in case the SDC residual is small enough. Thus, it is sufficient to set the tolerance τ to 10^{-1} or 10^{-2}. This approach can be combined with an absolute convergence criterion for the interface residual to avoid oversolving. Concluding, the convergence of each sub-step is assumed when the absolute convergence criteria are met, or in case (6.41) is satisfied.

Chapter 7.

Numerical results for incompressible flows

> I write by hand, making many, many corrections. I would say I cross out more than I write. I have to hunt for words when I speak, and I have the same difficulty when writing.
>
> *(Italo Calvino)*

In this chapter, we consider the application of arbitrarily high-order time integration using spectral deferred corrections (SDC), integral deferred corrections (IDC) and the Picard Integral Exponential Solver (PIES) to cell-centered collocated finite volume discretization of the unsteady incompressible Navier-Stokes equations. By considering some test cases, the theoretical order of convergence is confirmed for both the velocity and pressure. Furthermore, the accuracy and computational efficiency of SDC, IDC, and PIES compared to diagonally implicit Runge-Kutta methods and second order backward differencing scheme are investigated. The numerical results demonstrate a clear advantage over second order backward differencing regarding efficiency for a given temporal accuracy. For moderate accuracies, diagonally implicit Runge-Kutta methods prove to be competitive to SDC, IDC, and PIES. For high accuracies, it is certainly beneficiary to increase the order to five or higher.

The next chapter investigates several numerical examples when the higher

Parts of this chapter have been published in D. S. Blom et al. "On the application of spectral deferred corrections to incompressible flow on unstructured grids". In: *Journal of Computational Physics* under review (2016).

order schemes are applied to incompressible flows on moving meshes and partitioned fluid-structure interaction simulations.

7.1. Travelling waves

First, the well-known two-dimensional traveling waves test case is considered. An exact analytical solution of the incompressible Navier-Stokes equations exists [80]. The analytical solution is used as the initial solution of the simulations:

$$u\left(x,y,t\right) = 0.75 + 0.25\cos\left(2\pi\left(x-t\right)\right)\sin\left(2\pi\left(y-t\right)\right)\mathrm{e}^{-8\pi^2 t/\mathrm{Re}}, \quad (7.1)$$

$$v\left(x,y,t\right) = 0.75 - 0.25\sin\left(2\pi\left(x-t\right)\right)\cos\left(2\pi\left(y-t\right)\right)\mathrm{e}^{-8\pi^2 t/\mathrm{Re}}, \quad (7.2)$$

$$p\left(x,y,t\right) = -\frac{1}{64}\left(\cos\left(4\pi\left(x-t\right)\right) - \cos\left(4\pi\left(y-t\right)\right)\right)\mathrm{e}^{-16\pi^2 t/\mathrm{Re}}. \quad (7.3)$$

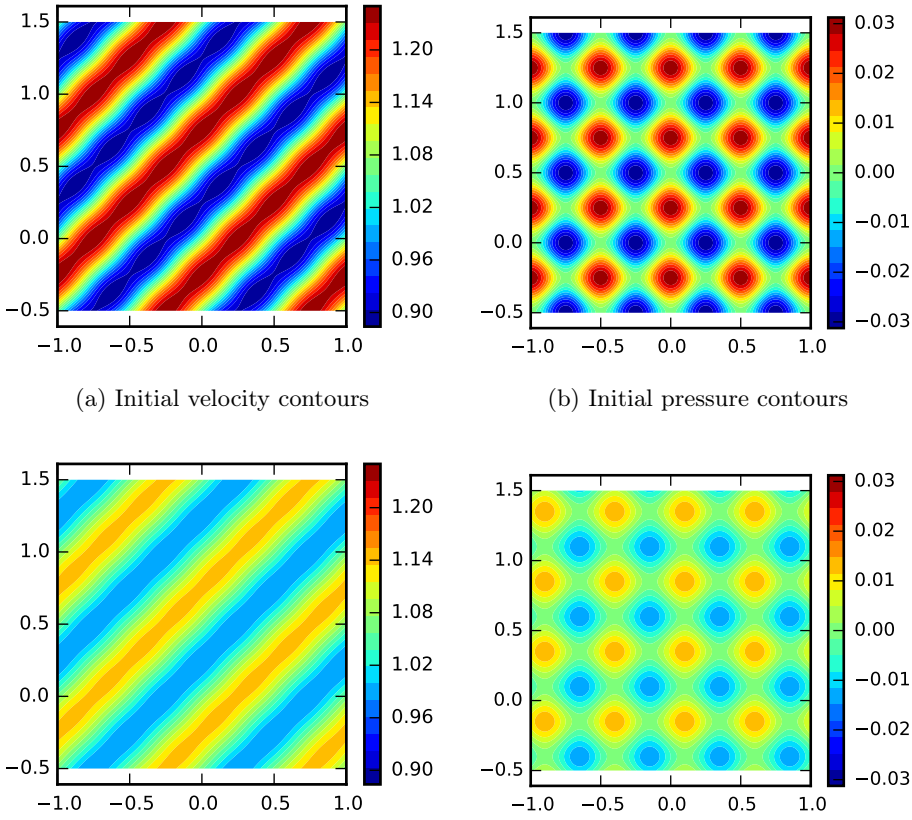
Periodic boundary conditions are used for the velocity as well as the pressure. The Reynolds number is set to $Re = 10$. The size of the square domain is two meter in both directions, with x in [-1,1] and y in [-0.5,1.5]. A uniform grid with size 50×50 is used. The simulations are carried out until $t = 0.05\,\mathrm{s}$. The convection and diffusion terms of the Navier-Stokes equations are discretized with second order central schemes. Figure 7.1 shows the pressure and velocity contours at $t = 0\,\mathrm{s}$ and at $t = 0.05\,\mathrm{s}$.

To minimize the computational costs of the simulations, the nonlinear systems are solved based on the convergence criterion:

$$\frac{\mathbf{r}_{\textit{non-linear}}}{\mathbf{r}_{\textit{non-linear,initial}}} < \tau_{Euler}, \quad (7.4)$$

with $\tau_{Euler} = 1.0 \cdot 10^{-2}$, and the initial momentum residual $\mathbf{r}_{\textit{non-linear,initial}}$ evaluated at the start of each non-linear solve. The tolerance τ_{SDC} is set to $1.0 \cdot 10^{-13}$. The reference solution is obtained with the SDC611 scheme with $\Delta t = 0.00125\,\mathrm{s}$. The relative error is defined as the L_2 norm of the difference between the approximation and reference solution scaled by the L_2 norm of the reference solution.

Figure 7.2 shows the convergence study for both the velocity and pressure field when using the spectral deferred correction method. Note that both the velocity and the pressure show the expected convergence order



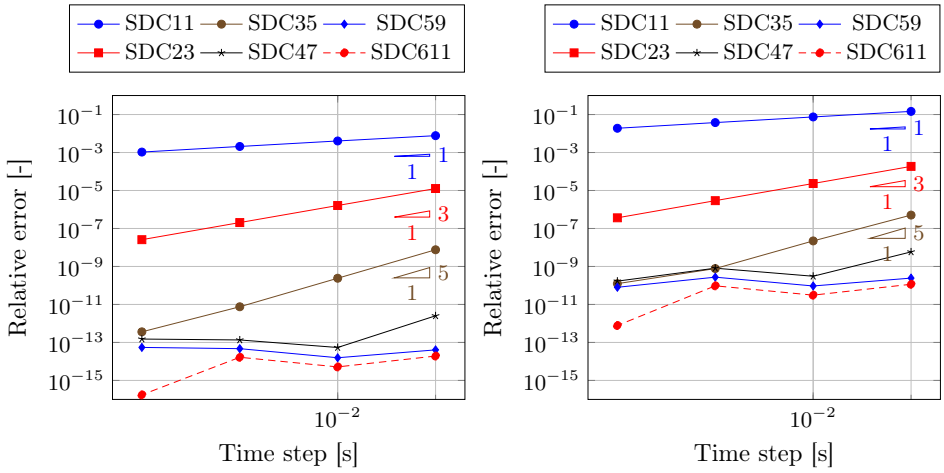
(a) Initial velocity contours

(b) Initial pressure contours

(c) Final velocity contours

(d) Final pressure contours

Figure 7.1.: Travelling waves test case: pressure and velocity contours of the initial and final solution.



(a) Velocity error for SDC schemes

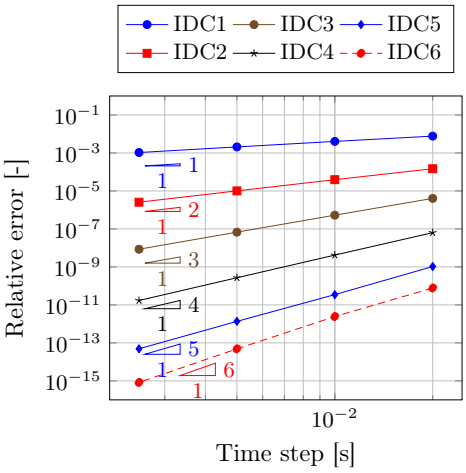
(b) Pressure error for SDC schemes

Figure 7.2.: Travelling waves test case: time step study for the SDC scheme. The temporal L_2 error norm of the velocity and pressure fields are shown.

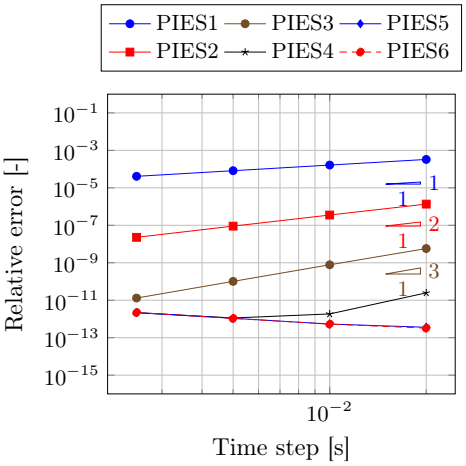
when decreasing the time step. Below an error of 10^{-12}, the numerical round-off error starts dominating the temporal error. Due to the high order of accuracy of the SDC scheme, machine precision is quickly reached for the SDC47, SDC59, and SDC611 methods.

Figure 7.3 shows the results of the time step studies for the IDC, PIES, SDIRK, ESDIRK and BDF methods. The results for the mentioned schemes show the expected order behavior. For the ESDIRK schemes, the ESDIRK5 reaches the highest level of accuracy.

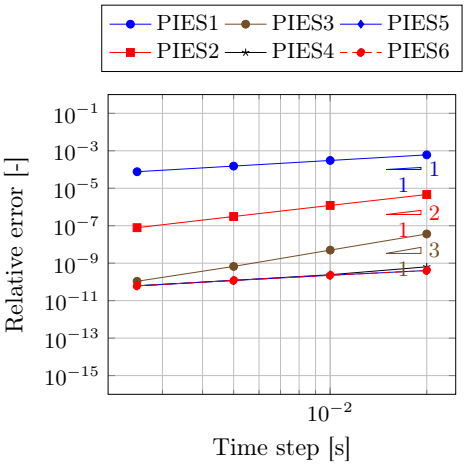
Figure 7.3b and 7.3c show the convergence plots when the PIES method is used with $\delta = 1 \cdot 10^{-13}$ and $\delta = 1 \cdot 10^{-5}$, respectively. When the selection criterion is set to $\delta = 1 \cdot 10^{-5}$, the accuracy of the interpolation with the exponential functions is decreased. This can also be observed in the plots, as the convergence plots level off quicker compared to Fig. 7.3b.



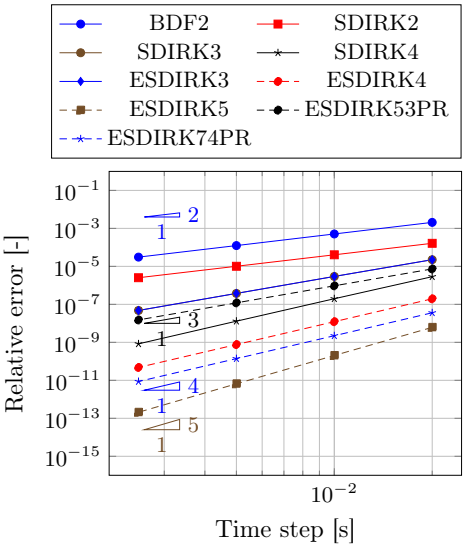
(a) Velocity error IDC schemes



(b) Velocity error PIES schemes, $\delta = 1{\cdot}10^{-13}$



(c) Velocity error PIES schemes, $\delta = 1 \cdot 10^{-5}$



(d) Velocity error SDIRK, ESDIRK and BDF schemes

Figure 7.3.: Travelling waves test case: time step study for IDC, PIES, SDIRK, ESDIRK and BDF schemes. The temporal L_2 error norm of the velocity field is shown.

7.2. Lid-driven cavity flow

The lid-driven cavity flow has been used by other researchers to study different time integration schemes [8, 66]. A two-dimensional flow is considered with Reynolds number $Re = 10$. The computational domain consists of a unit square with solid boundaries. At the top, a Dirichlet boundary condition is applied to the velocity in the x-direction with the value $1\,\mathrm{m/s}$. A uniform mesh of size 50×50 is used. A zero gradient boundary condition is used for the pressure, and the no-slip boundary condition is applied on the solid walls for the velocity. Second order centered schemes are used for the spatial discretization.

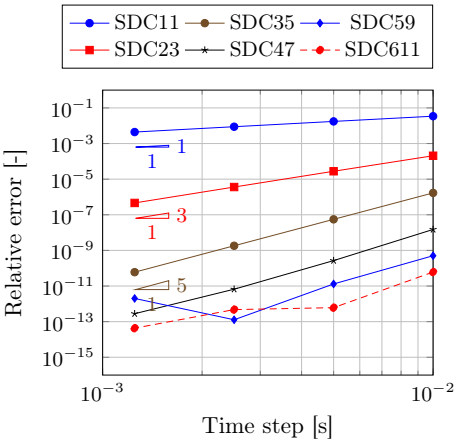
For the time step study, the initial transient of the simulation is considered. The flow is initially at rest, and the simulations are carried out until $t = 0.05\,\mathrm{s}$. The reference solution is obtained with the SDC815 scheme with $\Delta t = 0.000625\,\mathrm{s}$.

Figure 7.4 shows the temporal error of the velocity field against time step size for the SDC, IDC, SDIRK, ESDIRK and BDF schemes. The orders of accuracy for the different methods are confirmed. Again, due to the high order of accuracy of the SDC method with Gauss-Radau nodes, machine precision is quickly reached for the SDC611 method.

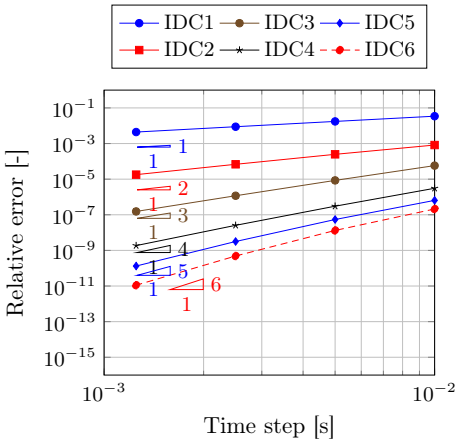
For the PIES method, an accuracy of 10^{-12} is reached due to the tolerance settings for computing the coefficients. The error constant of the PIES method is significantly smaller compared to the other time integration methods, due to the relatively large number of sub-steps with $\delta = 1.0 \cdot 10^{-13}$, namely 27. When the selection criterion is set to $\delta = 1.0 \cdot 10^{-5}$, as shown in Fig. 7.5, the convergence plots level off at an accuracy of 10^{-10}.

Figure 7.6 shows temporal error versus computational time for the same computations. As the order for the SDC method increases, the efficiency of the method increases substantially until machine precision is reached. This effect is not that large for the IDC method since the order of the method increases only by one when a sub-step is added. Hence, the SDC schemes outperform the IDC methods due to a higher order of convergence for the same number of quadrature nodes.

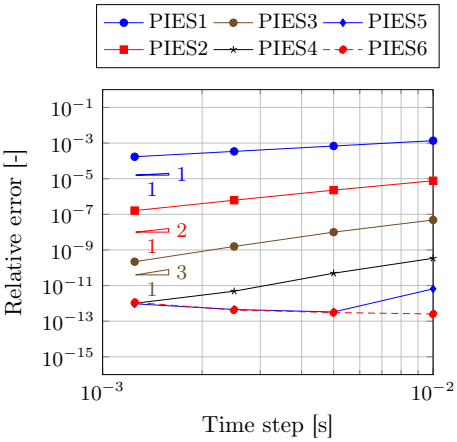
The PIES methods become more efficient with an increasing number of sweeps and show a relatively large increase in costs when decreasing the time step compared to the other methods. This is again explained by



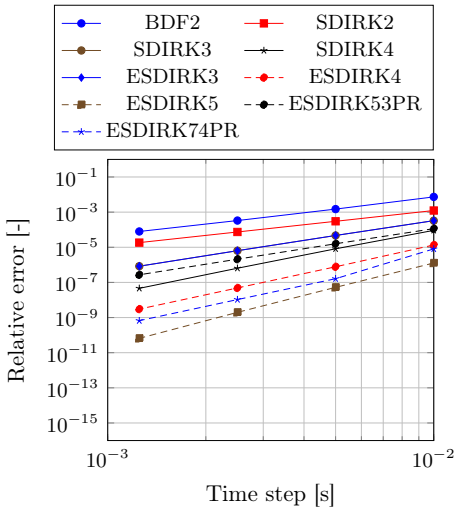
(a) Velocity error for SDC schemes



(b) Velocity error IDC schemes



(c) Velocity error PIES schemes, $\delta = 1 \cdot 10^{-13}$



(d) Velocity error SDIRK, ESDIRK and BDF schemes

Figure 7.4.: Lid-driven cavity flow: time step study for SDC, IDC, PIES, SDIRK, ESDIRK and BDF schemes. The temporal L_2 error norm of the velocity field is shown.

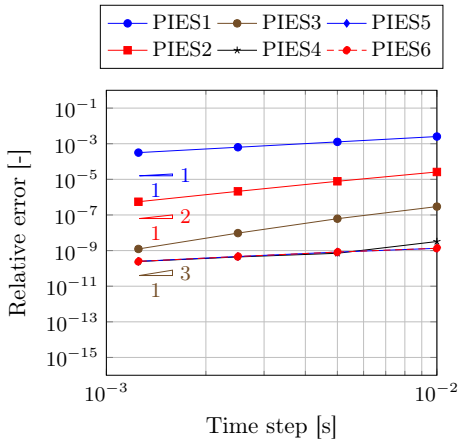
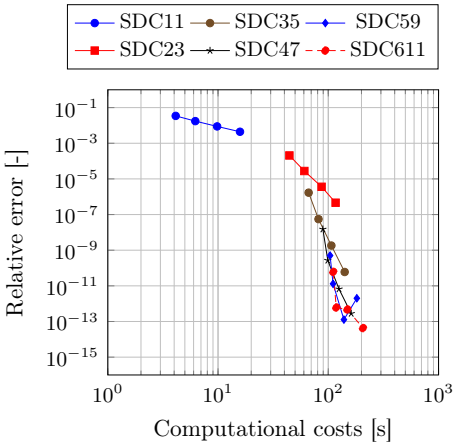


Figure 7.5.: Lid-driven cavity flow: time step study for the PIES scheme with $\delta = 1 \cdot 10^{-5}$. The temporal L_2 error norm of the velocity field is shown.

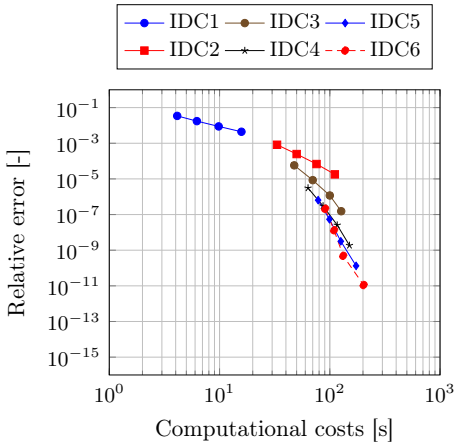
the large number of sub-steps selected. Note that the coefficients of the method are computed off-line, and the corresponding computational times are therefore not included in this plot. Judging from Fig. 7.7 ($\delta = 1 \cdot 10^{-5}$ and $\delta = 1 \cdot 10^{-13}$), the same observations can be made. Note though that due to the smaller number of sub-steps, the computational costs are not decreased but approximately at the same level as for the simulations with $\delta = 1 \cdot 10^{-13}$. The only difference between the two sets of simulations is a difference in the number of sub-steps. The order of convergence is the same, thus selecting more sub-steps only results in a smaller error constant.

The difference in computational cost for the BDF and ESDIRK methods are relatively small compared to the SDC, IDC and PIES methods due to the lower orders of accuracy, i.e. 1 to 5. It can be concluded that increasing the order is beneficial and enhances both efficiency and accuracy. Note that the fourth order ESDIRK74PR method is competitive to the ESDIRK5 method.

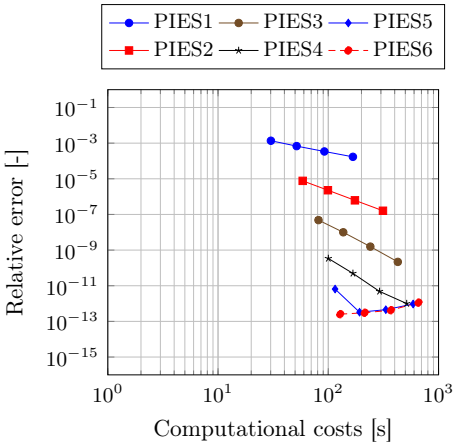
Figure 7.8 summarizes the results of Fig. 7.6 by using a linear fit through the best data points, thus only differentiating between SDC, IDC, PIES, BDF, SDIRK and ESDIRK methods. The result of the largest time step of the PIES simulations is used in Fig. 7.8. The SDIRK and ESDIRK



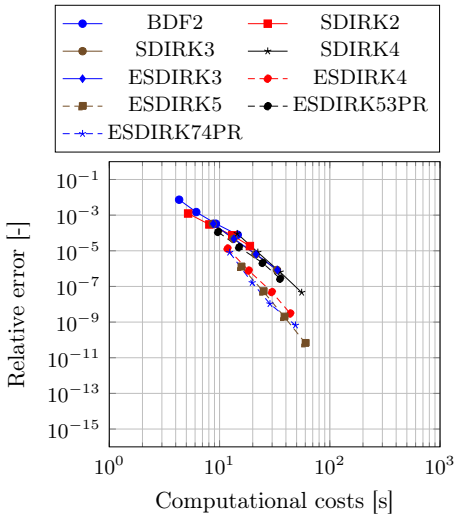
(a) Velocity error for SDC schemes



(b) Velocity error IDC schemes



(c) Velocity error PIES schemes



(d) Velocity error SDIRK, ESDIRK and BDF schemes

Figure 7.6.: Lid-driven cavity flow: comparison of computational efficiency versus temporal error in velocity for the SDC, IDC, PIES, SDIRK, ESDIRK and BDF schemes.

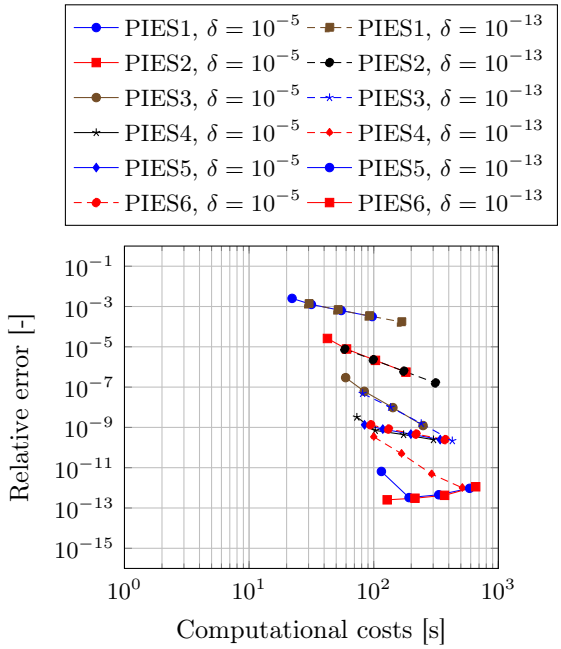


Figure 7.7.: Lid-driven cavity flow: comparison of computational efficiency versus temporal error in velocity for the PIES scheme with $\delta = 1 \cdot 10^{-5}$ and $\delta = 1 \cdot 10^{-13}$.

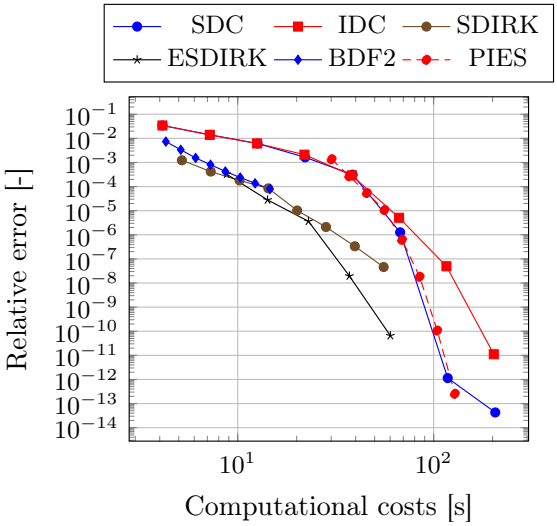


Figure 7.8.: Lid-driven cavity flow: smoothed computational costs versus error for SDC, IDC and ESDIRK.

schemes are computationally more efficient than the SDC, IDC and PIES methods for the accuracy range shown. The PIES method is competitive to SDC and becomes more efficient when the number of sweeps is increased.

Note that in contrast to the other time integration schemes, the ESDIRK methods need the function values f at the left end point of the time domain. Comparing the results of PIES and SDC with SDIRK, it is expected that they will outperform the SDIRK scheme for high accuracies.

7.3. Lid-driven cavity flow with time dependent boundary conditions

The third case consists again of a lid-driven cavity flow. Now an oscillating boundary condition is applied on the top wall [66] to study the behavior for the different time integration schemes with a time dependent boundary condition. Hence, a non-autonomous system is considered, with the initial

value problem defined as

$$\frac{\mathrm{d}}{\mathrm{d}t} u\left(t\right) = f\left(u\left(t\right), t\right), \qquad u(0) = u^0, \qquad t \in \left[0, T\right], \tag{7.5}$$

in contrast to an autonomous system, for which the function does not depend on time:

$$\frac{\mathrm{d}}{\mathrm{d}t} u\left(t\right) = f\left(u\left(t\right)\right), \qquad u(0) = u^0, \qquad t \in \left[0, T\right]. \tag{7.6}$$

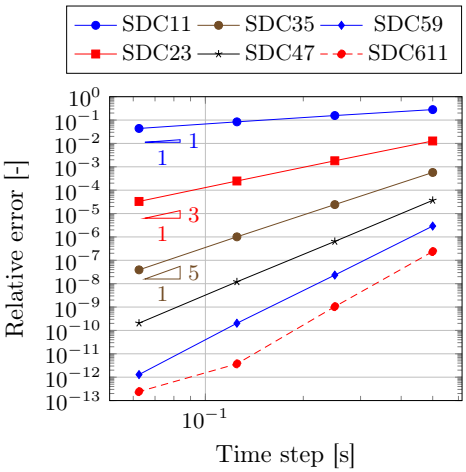
A sinusoidal forcing is imposed on the top wall:

$$u_x = \sin\left(\omega\, t\right), \tag{7.7}$$

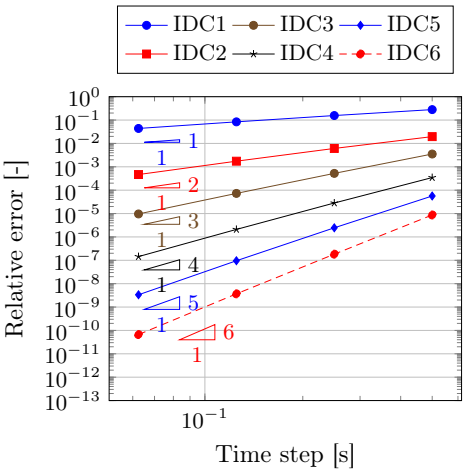
with the velocity $\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$, the frequency of the oscillations ω, and the time t. The Reynolds number is set to $Re = 400$, and the frequency of oscillations to $\omega = 0.5\,\pi$. A uniform grid is used with 129^2 cells. After the initial transient, the flow reaches a periodic state of which one period is simulated with the different time integration schemes.

Figure 7.9, 7.10, 7.11 and 7.12 show the results of the performed simulations. The expected order of accuracy of the spectral deferred correction, integral deferred correction schemes, and PIES is confirmed for this test case when a time dependent boundary condition is imposed. However, the SDIRK4 and ESDIRK5 methods reduce to third order accuracy. The ESDIRK74PR results in the lowest temporal error compared to the other (E)SDIRK methods.

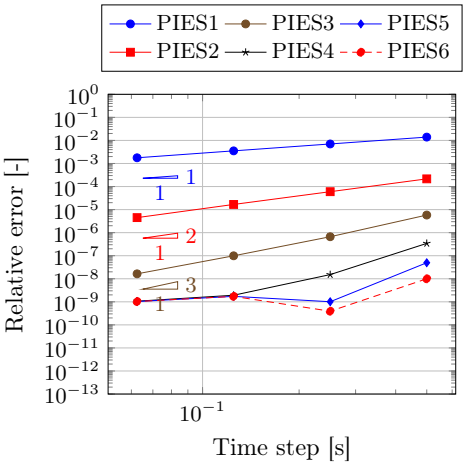
The efficiency of the SDC, IDC, PIES and ESDIRK methods is shown in Fig. 7.13 where the results are summarized with a linear fit through the best data points of Fig. 7.11. The (E)SDIRK schemes prove to be competitive to SDC and IDC for the accuracy range shown. IDC and SDC exhibit very similar computational cost for a given accuracy. The PIES method shows similar computation times to SDC for accuracies between 10^{-4}–10^{-8}.



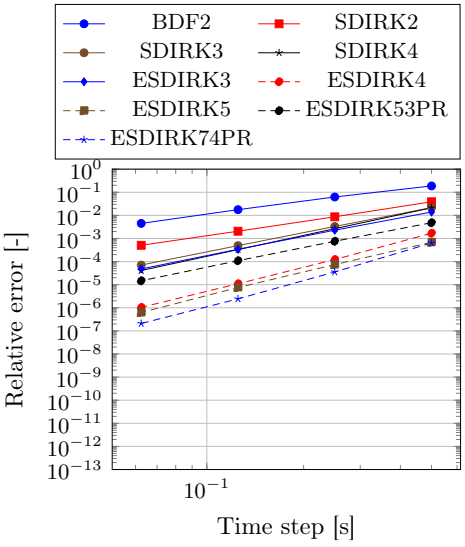
(a) Velocity error for SDC schemes



(b) Velocity error for IDC schemes



(c) Velocity error for PIES schemes, $\delta = 1 \cdot 10^{-13}$



(d) Velocity error for SDIRK, ESDIRK and BDF schemes

Figure 7.9.: Lid-driven cavity flow with time dependent boundary condition: time step convergence study. The temporal L_2 error norm of the velocity field is shown.

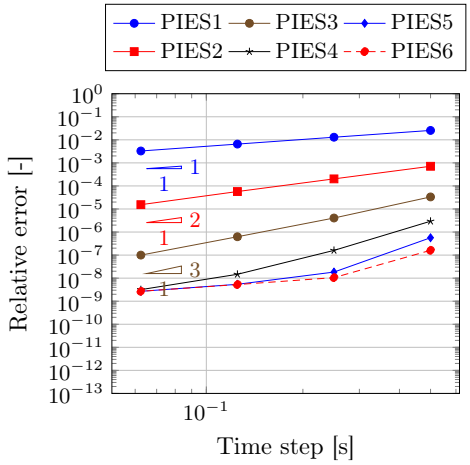


Figure 7.10.: Lid-driven cavity flow with time dependent boundary condition: time step convergence study for PIES with $\delta = 1 \cdot 10^{-5}$. The temporal L_2 error norm of the velocity field is shown.

7.4. Flow around a cylinder

Finally, a more complex benchmark problem of the flow around a cylinder is considered [64, 89]. John [64] has also studied this test case with different time integration schemes, namely the second order Crank-Nicolson scheme and the fractional-step θ-scheme. The influence of the time integration scheme on the lift coefficient is considered here.

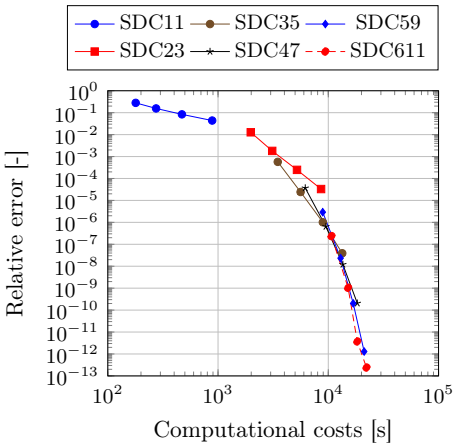
The length of the domain is 2.2 m, and the height H is 0.41 m. The diameter of the cylinder is 0.1 m, with the center of the cylinder located at $(0.2, 0.2)$, where the origin of the domain is located at the bottom left corner. A schematic of the test case is shown in Fig. 7.14.

A time-dependent inflow profile

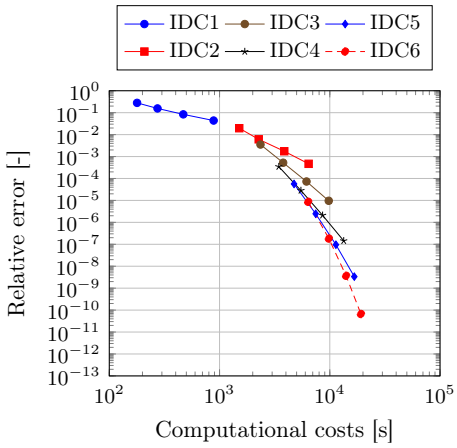
$$u_x(y,t) = \frac{6\,y}{H^2}\,(H-y)\sin\left(\frac{\pi t}{8}\right) \qquad 0 \le y \le H \tag{7.8}$$

is prescribed for $0 \le t \le 8\,\mathrm{s}$. The mean inflow velocity equals $u_x\,(t) = \sin\left(\frac{\pi\,t}{8}\right)\mathrm{m/s}$. Therefore, the maximum inlet velocity is $u_{\max} = 1\,\mathrm{m/s}$.

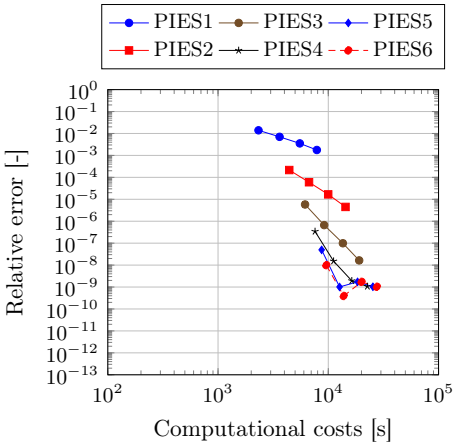
For the velocity, zero gradient boundary conditions are applied at the



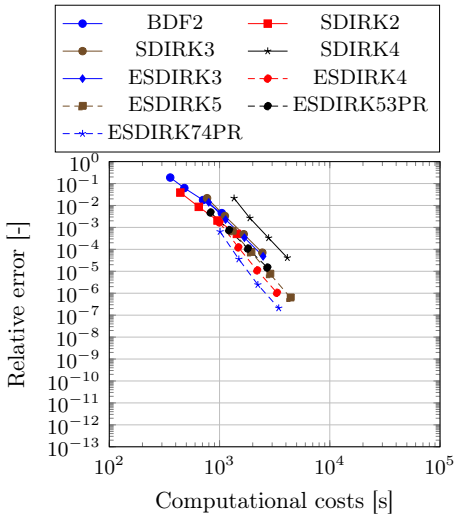
(a) Velocity error for SDC schemes



(b) Velocity error IDC schemes



(c) Velocity error PIES schemes, $\delta = 1 \cdot 10^{-13}$



(d) Velocity error SDIRK, ESDIRK and BDF schemes

Figure 7.11.: Lid-driven cavity flow with time dependent boundary condition: comparison of computational efficiency versus error in velocity.

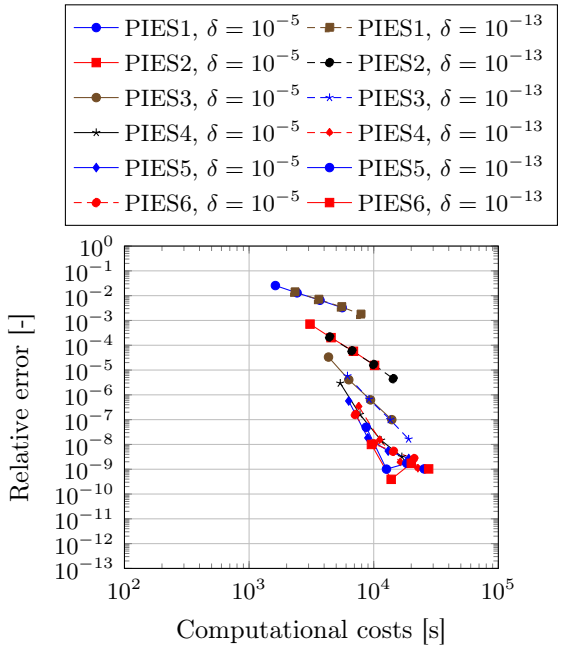


Figure 7.12.: Lid-driven cavity flow with time dependent boundary condition: comparison of computational efficiency versus error in velocity for PIES with $\delta = 1 \cdot 10^{-5}$ and $\delta = 1 \cdot 10^{-13}$.

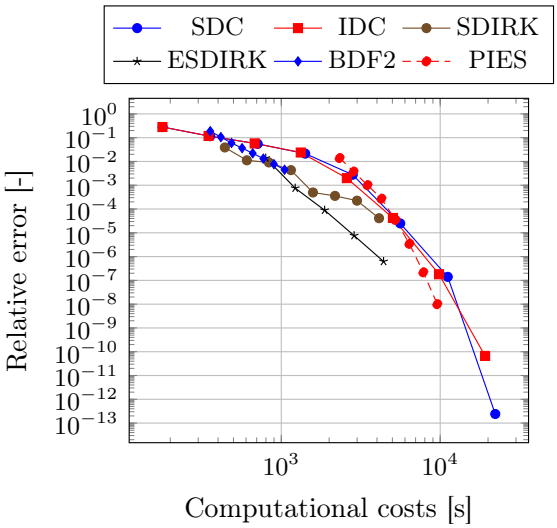


Figure 7.13.: Lid-driven cavity flow with time dependent boundary condition: smoothed computational costs versus error for SDC, IDC and ESDIRK.

Figure 7.14.: Schematic of the domain for the flow around a cylinder test case

outlet and at the other boundaries the no-slip condition is imposed. At the outlet, the average pressure is set to zero, and the zero gradient boundary condition is applied on the remaining boundaries. The kinematic viscosity is $\nu = 1.0 \cdot 10^{-3}\,\mathrm{m}^2/\mathrm{s}$, and the fluid density is $\rho = 1.0\,\mathrm{kg}/\mathrm{m}^3$. A zero initial solution is used for both the velocity and pressure. The mesh consists of 38 156 control volumes. Based on the mean inflow velocity, the Reynolds number varies between zero and 100.

A second order upwind scheme is used for the convection term of the Navier-Stokes equations. The diffusion term is discretized with a second order centered scheme including a non-orthogonality correction. A time step of 0.01 s is used for all time integration schemes.

Figure 7.15 shows the lift coefficient c_l, drag coefficient c_d and the difference of the pressure between the front and back of the cylinder Δp for a subset of the studied time integration schemes, namely SDC, ESDIRK and BDF schemes. The lift and drag coefficient are defined as:

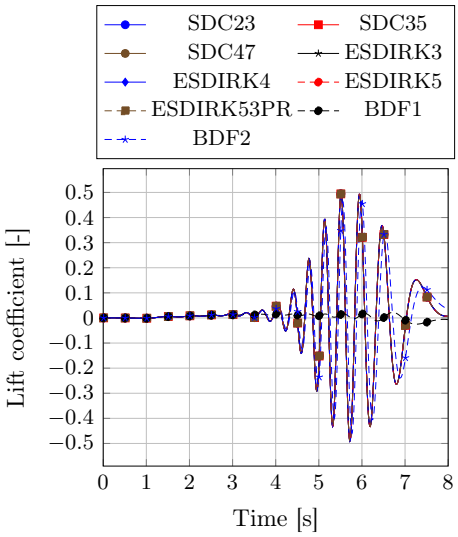
$$c_l\,(t) = -\frac{2\,L\,(t)}{\rho\,{u_{\max}}^2\,S}, \tag{7.9}$$

$$c_d\,(t) = \frac{2\,D\,(t)}{\rho\,{u_{\max}}^2\,S}, \tag{7.10}$$

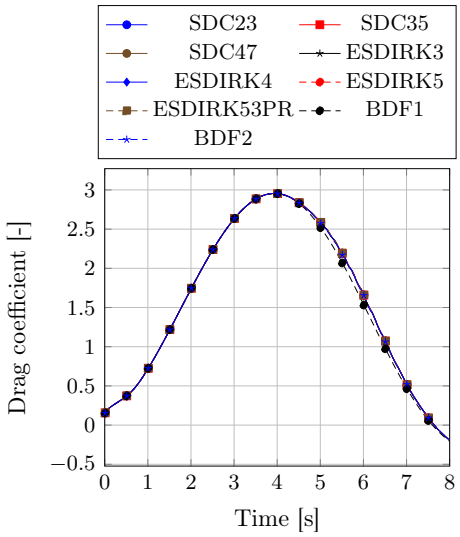
with the lift force L, drag force D, and reference area S (0.1 m). The lift coefficient is the most sensitive to the used time discretization scheme.

Also, it is immediately clear that the backward Euler method produces inaccurate results for c_l. BDF2 follows the general trend of the lift history but introduces a phase shift in time. Fig. c shows a zoom of the lift coefficient. Note that the ESDIRK4, ESDIRK5, SDC35, and SDC47 methods produce identical results indicating that the temporal discretization error is not dominating for these simulations. The third order SDC23 and ESDIRK53PR lie close to the reference values.

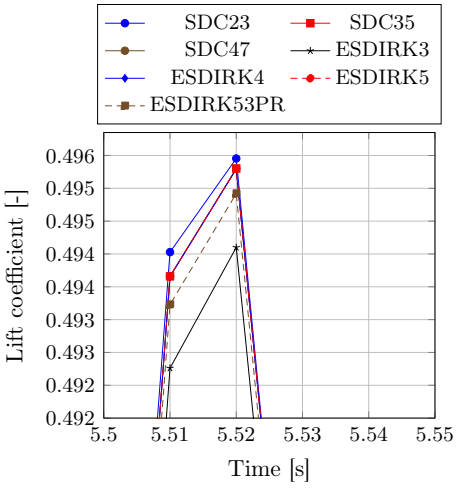
Table 7.1 shows the computational times, relative error of the lift coefficient for $0 \leq t \leq 8$ for a second set of simulations. Namely, the length of a sub-step or stage of each method is approximately the same for each time integration scheme. The relative error of the lift coefficient is determined by using a cubic interpolation of the lift coefficient in the complete time interval to the time instants of the reference solution, thereafter computing the L_2-norm of the difference between the approximation and reference



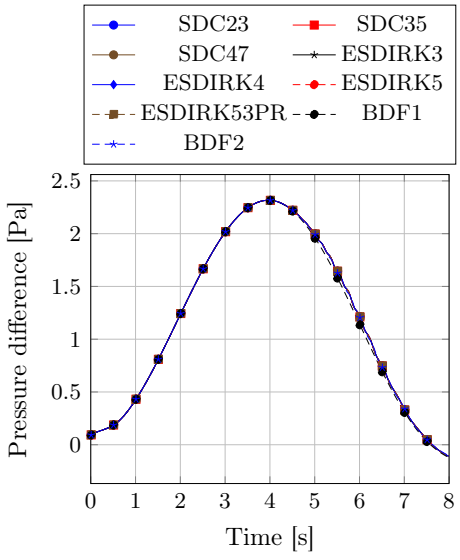
(a) Lift coefficient



(b) Drag coefficient



(c) Zoom of lift coefficient



(d) Pressure difference

Figure 7.15.: Lift, drag coefficient, and pressure difference for SDC, ESDIRK and BDF time integration schemes

Table 7.1.: Flow around a cylinder test case: time step versus computational time for different time discretization schemes

Method	Time step [s]	Computational time [h]	Relative error
SDC47	0.04	33.1	$1.10 \cdot 10^{-3}$
SDC35	0.03	28.3	$1.55 \cdot 10^{-3}$
SDC23	0.02	24.4	$9.76 \cdot 10^{-3}$
ESDIRK53PR	0.04	13.5	$8.73 \cdot 10^{-2}$
ESDIRK5	0.07	14.9	$2.03 \cdot 10^{-2}$
ESDIRK4	0.05	14.1	$2.59 \cdot 10^{-2}$
ESDIRK3	0.03	14.5	$1.23 \cdot 10^{-1}$
BDF2	0.01	10.6	$6.16 \cdot 10^{-1}$
BDF1	0.01	13.8	$1.01 \cdot 10^{+0}$

solution, and scaling by the L_2-norm of the reference solution.

The ESDIRK schemes have similar computational costs but show a clear increase in accuracy for higher orders. The SDC schemes have higher timings, however also smaller errors for the lift coefficient. It is interesting to point out that large errors for the BDF1 and BDF2 methods occur, where the error of BDF1 is larger than 100% for this simulation run.

Similar conclusions for the ESDIRK and SDC methods can be drawn from this case, namely that ESDIRK proves to be an efficient time integration method for moderate accuracies. SDC schemes of higher order are good candidates for even higher accuracies.

Chapter 8.

Numerical results for partitioned fluid-structure interaction simulations using integral deferred corrections

> Before I came here I was confused about this subject. Having listened to your lecture I am still confused. But on a higher level.
>
> *(Enrico Fermi)*

Now, we consider arbitrarily high-order time integration schemes using integral deferred corrections for partitioned fluid-structure interaction simulations. The integral deferred correction method is applied for the first time to a fluid-structure interaction problem. By partitioning the computational domain into a fluid and solid domain, a partitioning error is introduced. Furthermore, the use of a segregated solution technique (iterated PISO) for the incompressible Navier-Stokes equations introduces a second splitting error. By using sub-iterations accelerated by Anderson acceleration, the partitioning error is minimized.

In this chapter, we apply the IDC method which uses uniform sub-steps. With uniform sub-steps, information from previous sub-steps and previous time steps can easily be reused by the Anderson acceleration technique to

Parts of this chapter have been published in D. S. Blom et al. "Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using integral deferred corrections". In: *Journal of Computational Physics* under review (2016).

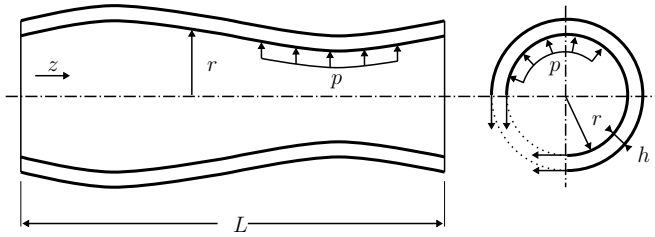


Figure 8.1.: Geometry of the one-dimensional model for the unsteady, incompressible flow through an elastic tube. The length L, the inner radius r, the pressure p acting on the inner wall of the tube, and the wall thickness h is shown. Deformation of the tube walls in radial direction is caused by the fluid pressure p acting on the inner tube walls. The figure is taken from Mehl et al. [76].

speed up the convergence of the coupling iterations in the current sub-step. In the case of spectral deferred corrections or the SDIRK method, non-uniform sub-steps are applied. Then, it is necessary to scale the values of the interface Jacobian by the used sub-step. Otherwise, the change in sub-step size will be incorporated in the approximation of the interface Jacobian possibly increasing the number of coupling iterations instead of accelerating the method.

By considering several test cases, the theoretical order of convergence is confirmed for the velocity as well as the pressure. The accuracy of the integral deferred method is compared to singly-diagonally-implicit Runge-Kutta methods (SDIRK). The numerical results show an order reduction for the SDIRK methods for which the coefficients do not satisfy order conditions derived from the Prothero-Robinson example, whereas the integral deferred correction achieves the expected order of accuracy. The use of different time integration schemes for the fluid and solid domain shows that the IDC method can be effectively combined with a second-order time integration scheme for the structural domain.

8.1. Unsteady flow in a one-dimensional tube

The first case consists of an incompressible, unsteady flow in a one-dimensional flexible tube [34, 36]. The tube has a circular cross-section and length L,

as shown in Fig. 8.1. This example is straightforward to implement and is a good example of a strongly coupled fluid-structure interaction problem. Two different structural models are considered based on a Hookean constitutive equation, and a generalized string model based on linear elasticity theory where an inertia term is included.

8.1.1. Governing equation for the fluid model

The flow is governed by the equations for conservation of mass and conservation of momentum written in conservative form as

$$\frac{\rho_f \partial a}{\partial t} + \frac{\rho_f \partial a v}{\partial z} = 0, \tag{8.1}$$

and

$$\frac{\partial a v}{\partial t} + \frac{\partial a v^2}{\partial z} + \frac{1}{\rho_f}\left(\frac{\partial a p}{\partial z} - p\frac{\partial a}{\partial z}\right) = 0, \tag{8.2}$$

with the coordinate z along the axis of the elastic tube, the cross-sectional area a and the velocity v in z-direction. r is the inner radius of the tube, the time is denoted by t, p represents the pressure, and ρ_f is the density of the fluid. Gravity and viscosity are not taken into account by the model.

8.1.2. Governing equation for the structural model

A Hookean constitutive relation describes the behavior of the elastic tube wall. The inertia of the tube wall is neglected with respect to the inertia of the fluid, resulting in a structure without mass. The stress acting in circumferential direction $\sigma_{\phi\phi}$ is approximated with

$$\sigma_{\phi\phi} = E\frac{r - r_0}{r_0} + \sigma_{\phi\phi_0} \tag{8.3}$$

for a given Young's modulus E and radius r_0 for which $\sigma_{\phi\phi} = \sigma_{\phi\phi_0}$. Only radial motion of the tube wall is allowed. The forces acting on the fluid-structure interface are in balance, thus

$$rp = \sigma_{\phi\phi}\, h. \tag{8.4}$$

The following relation can be found after substitution of the constitutive

equation (8.3) into (8.4)

$$rp = \frac{Eh}{\rho r_0}(r - r_0) + r_0\, p_0, \tag{8.5}$$

which can also be written as

$$a = a_0 \left(\frac{\frac{p_0}{2\,\rho_f} - c_{MK}^2}{\frac{p}{2\,\rho_f} - c_{MK}^2} \right)^2, \tag{8.6}$$

given the Moens-Korteweg wave speed c_{MK}

$$c_{MK} = \sqrt{\frac{E\,h}{2\,\rho_f\, r_0}}. \tag{8.7}$$

8.1.3. Generalized string model for the structural model

Since the previously discussed structural model does not take inertia into account and hence does not include a time derivative, we consider a second structural model. The model is based on linear elasticity theory for a cylindrical tube with a small thickness. The axial and circumferential displacement of the wall are not taken into account. The governing equation for the structure is as follows:

$$\rho_s\, h\, \frac{\partial^2 r}{\partial t^2} - \kappa\, G\, h\, \frac{\partial^2 r}{\partial z^2} + \frac{E\,h}{1-\nu^2}\, \frac{r - r_0}{r_0^2} - \gamma\, \frac{\partial^3 r}{\partial^2 z\, \partial t} = p, \tag{8.8}$$

with the inner radius r, the structural density ρ_s, and the thickness of the wall h. The shear modulus G, Young's modulus E, and Poisson's coefficient ν are introduced as well. The model is further simplified by neglecting the viscoelastic term, i.e. $\gamma = 0$ [52]. The Timoshenko shear corrector factor κ is related to Poisson's coefficient with [30]

$$\kappa = \frac{2\,(1+\nu)}{4 + 3\,\nu}. \tag{8.9}$$

8.1.4. Discretization

The tube is discretized with N cells of length Δz. The velocity and pressure of the flow are determined in the cell centers. A central discretization is employed for the continuity and momentum equation. However, the convective term of the momentum equation is discretized with a first-order upwind scheme.

The velocity at the inlet is calculated as

$$v_{in} = v_0 + \frac{v_0}{10} \sin^2 \left(\frac{\pi \, v_0 \, t}{L} \right). \tag{8.10}$$

The problem can be described with two dimensionless parameters, namely the dimensionless stiffness κ

$$\kappa = \frac{c_0}{v_0} = \frac{\sqrt{\frac{Eh}{2\rho_f r_0} - \frac{p_0}{2\rho_f}}}{v_0}, \tag{8.11}$$

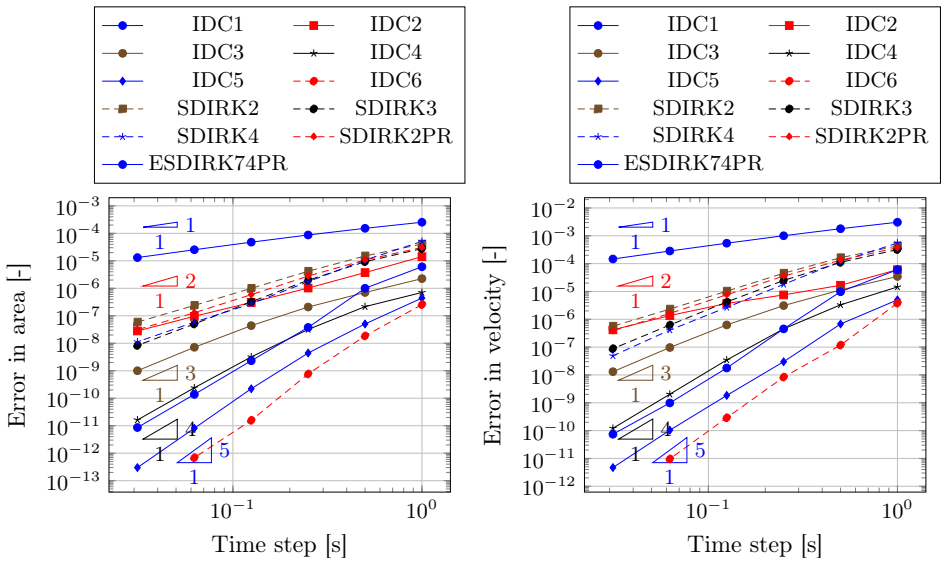
and the dimensionless time step $\tau = \frac{v_0 \Delta t}{L}$. For the computations shown in this section, κ is set to $\kappa = 10$, and τ is varied to show the order of convergence of the time stepping schemes. This results in a strong coupling between the fluid and the structure [36]. The initial conditions are the dimensionless velocity $\frac{v_0}{c_0} = 0.1$, the dimensionless pressure $\frac{p}{\rho_f c_0^2} = 0$, and the dimensionless cross-sectional area $\frac{a}{a_0} = 1$.

8.1.5. Numerical results

Three separate simulation sets are used to compare integral deferred corrections with implicit Runge-Kutta methods. First, we use a structural model without inertia or time-derivative term. Then, we introduce a time derivative into the structural equals and compare IDC with SDIRK. Finally, we look at combining different time integration schemes for the fluid and solid domains.

Results for integral deferred corrections and SDIRK

Figure 8.2 shows the convergence study for both the velocity and cross-sectional area of the tube. Both the cross-sectional area and the velocity



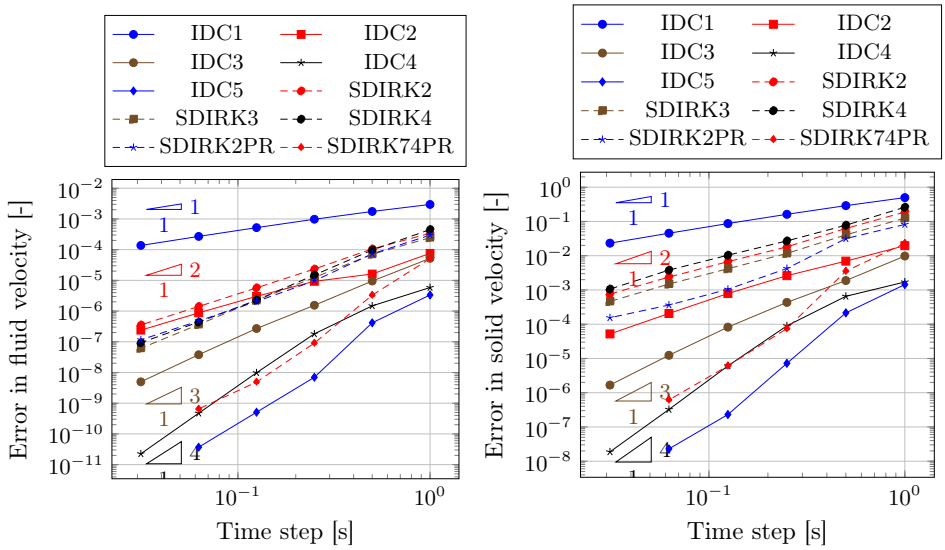
(a) Area (b) Velocity

Figure 8.2.: Unsteady flow in a one-dimensional tube: order of convergence confirmed for the IDC method when applied to a fluid-structure interaction problem. Order reduction observed for the fourth order SDIRK4 method.

show the expected order of convergence for the IDC method when increasing the order from 1 to 6. However, for the SDIRK method, an order reduction to third order is apparent for SDIRK4 as shown in Fig. 8.2a. An order reduction is also observed for the velocity (Fig. 8.2b).

Results for IDC and SDIRK with inertia for the solid

Figure 8.3 shows the results of the time step study for the fluid-structure interaction problem with the second structural model, i.e. an inertia term for the solid is added. The figure shows the relative error for the velocity of the fluid and the velocity of the structure. The observed order of accuracy for both the SDIRK3 and SDIRK4 methods is reduced to second order, as shown in Fig. 8.3b. However, the IDC method still shows the expected orders of accuracy.



(a) Velocity of the fluid

(b) Velocity of the structure

Figure 8.3.: Unsteady flow in a one-dimensional tube with inertia for the solid: order of convergence confirmed for the IDC method when applied to a fluid-structure interaction problem. Again an order reduction observed for the fourth order SDIRK4 method, and now also for the SDIRK3 method.

Note that the ESDIRK74PR method does not suffer from an order reduction for this fluid-structure interaction problem. The coefficients for the method have been derived by Rang [83] by analyzing the Prothero-Robinson example and are of interest when index-2 differential algebraic equations (DAE) are considered [57] such as the incompressible Navier-Stokes equations. However, the simulation with the smallest time step did not finish, due to convergence problems of the coupling iterations.

The SDIRK2PR also satisfies the new order conditions derived from the Prothero-Robinson example and shows the expected second order of accuracy. The SDIRK3 and SDIRK4 methods do not satisfy those extra order conditions and do show an order reduction to second order, as mentioned earlier.

When the integral deferred correction method is used, the theoretical orders of convergence are observed. Also, convergence problems for the coupling iterations did not occur. IDC5 with time step $\Delta t = 0.03125\,\mathrm{s}$ is used as the reference simulation, and is therefore not shown in the convergence plots.

Different time integration schemes for the fluid and solid domains

When considering the partitioned setup of the fluid-structure interaction simulation, it can be expected that the dedicated solvers for the different domains use different time integration methods. Here, we consider combining the IDC method for the fluid with the second order backward differencing method (BDF2) for the solid. Naturally, other time integration schemes can also be used for the solid mechanics solver. Different choices can be made how the fluid and solid solvers are coupled in time. One approach would be to only exchange information at t^{n+1}. For this example, we have chosen to perform sub-iterations at each sub-step and keep the solutions of each sub-step in memory for the solid solver for robustness and accuracy reasons. Figure 8.4 shows the procedure for one sweep.

Figure 8.5 shows the results of the time step study when changing the order of accuracy of IDC in the fluid domain, while keeping the second order accurate BDF2 scheme in the structural domain. As can be seen in the figure, for large time steps the accuracy of the overall simulation using a third or higher order method is higher compared to IDC2-BDF2. However, for the velocity of the solid and for small time steps the accuracy

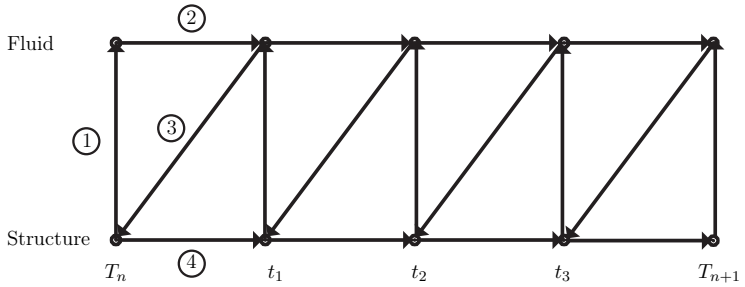


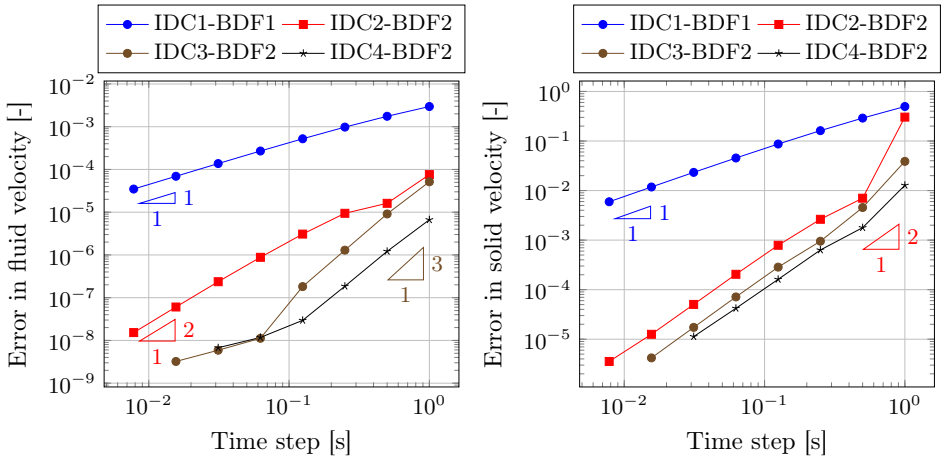
Figure 8.4.: Schematic of the conventional staggered procedure when combined with the integral deferred correction method.

is limited by the solid domain as can be expected.

So, it can be concluded that it is beneficiary to use a higher order time integration scheme for the fluid domain to reach a high accuracy for the fluid solution assuming that the accuracy of the structure does not dominate the overall solution. Generally, the time scales of the fluid domain are much smaller than the time scales in the structural domain, which is the reason why some authors use sub-cycling [82] for those type of conditions, and why one expects a benefit using a high order of accuracy in the fluid domain and a lower order approximation for the structure.

8.2. Lid-driven cavity flow with a moving mesh

The lid-driven cavity flow has been used by other researchers to study different time integration schemes [8, 66]. A two-dimensional incompressible flow is considered with Reynolds number $Re = 10$. The computational domain consists of a unit square with solid boundaries. At the top, a Dirichlet boundary condition is applied to the velocity in the x-direction with the value $1\,\mathrm{m/s}$. A zero-gradient boundary condition is used for the pressure, and the no-slip boundary condition is applied on the solid walls for the velocity. Second order centered schemes are used for the spatial discretization.



(a) Velocity of the fluid

(b) Velocity of the structure

Figure 8.5.: Unsteady flow in a one-dimensional tube with inertia for the solid: BDF1/BDF2 time stepping for the solid domain, and IDC time stepping for the fluid domain. The accuracy of the overall simulation for large time steps using a third or higher order method is more accurate than the IDC2-BDF combination.

8.2.1. Lid-driven cavity flow with internal moving mesh

The internal mesh is deformed during the simulation as a first step to show the order of convergence for the velocity and pressure with a moving mesh. For the time step study, the initial transient of the simulation is considered. The flow is initially at rest, and the simulations are carried out until $t = 0.05\,\mathrm{s}$. The reference solution is obtained with the IDC6 scheme with $\Delta t = 0.00015625\,\mathrm{s}$. A uniform mesh of size 25×25 is used.

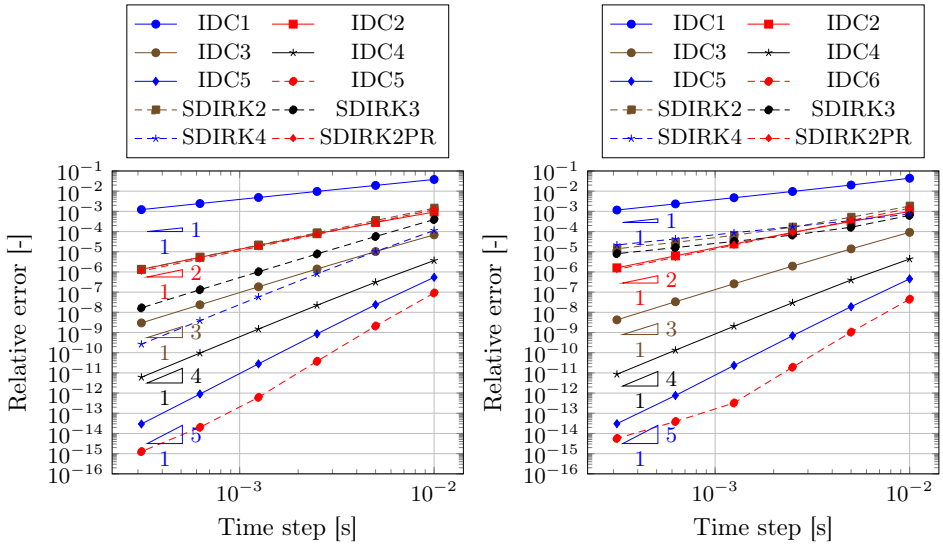
Figure 8.6 shows the temporal error of the velocity and pressure fields against the time step size for the IDC and SDIRK schemes. The theoretical order of accuracy for the velocity as well as for the pressure is confirmed for the integral deferred correction scheme. However, the SDIRK3 and SDIRK4 methods suffer from an order reduction to two for the pressure. Note that the SDIRK2PR shows the expected orders of convergence for the velocity, as well as for the pressure. This can be explained by the fact that the coefficients of the method satisfy extra order conditions as derived in Rang [84] by considering the Prothero-Robinson example, as mentioned in the previous section.

8.2.2. Lid-driven cavity flow with moving block

The complexity of the lid-driven cavity flow is increased by including a block in the middle of the computation domain which follows a prescribed rigid body motion. The width and height of the block are 0.1 m. The flow is initially at rest, and the simulations are carried out until $t = 0.05\,\mathrm{s}$. The reference solution is obtained with the IDC6 scheme with $\Delta t = 0.00015625\,\mathrm{s}$. A uniform mesh of size 24×24 is used. Figure 8.7 shows the pressure and vorticity contours at the end of the simulation.

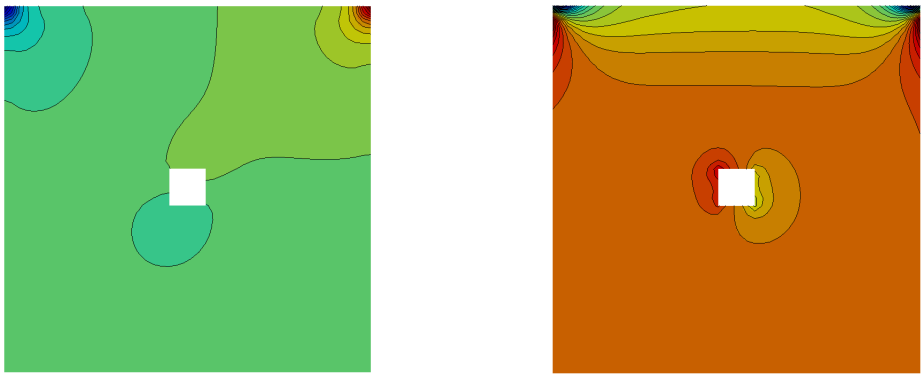
Figure 8.8 shows the temporal error of the velocity and pressure fields against the time step size for the IDC and SDIRK methods. The theoretical order of accuracy for the velocity as well as for the pressure is confirmed for the integral deferred correction scheme. SDIRK2, SDIRK3, and SDIRK4 reduce to first-order accuracy in time for the pressure, whereas the SDIRK2PR method shows second order accuracy for both the velocity and pressure.

The pressure shows a reduction for the IDC5 and IDC6 methods when using small time steps. This can be explained by the fact that a zero-



(a) Velocity (b) Pressure

Figure 8.6.: Lid driven cavity with internal moving mesh: time step study for the IDC and SDIRK time integration schemes. The temporal L_2 norm of the velocity and pressure error norm is shown. The SDIRK3 and SDIRK4 schemes suffer from an order reduction for the pressure. The integral deferred correction method performs as expected.



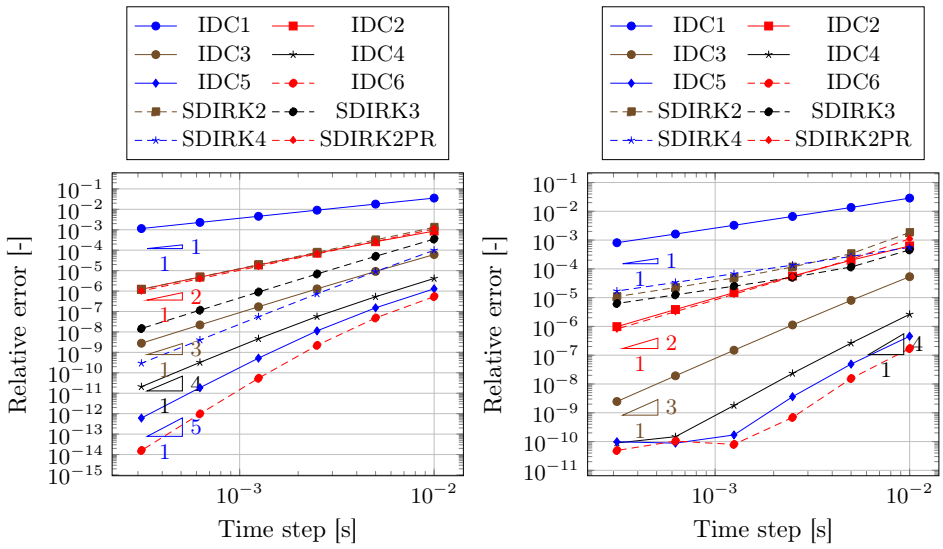
(a) Pressure contours (b) Vorticity contours

Figure 8.7.: Lid-driven cavity flow with moving block: pressure and vorticity contours at the end of the time interval

gradient boundary condition is used for the pressure at the (moving) walls. A more consistent boundary condition is proposed by Gresho and Sani [54] which evaluates the momentum equation on the boundary to obtain the pressure boundary condition. However, this is a non-trivial task for unstructured collocated grids and is therefore left to be assessed in future studies. Therefore, in this study, the often applied zero-gradient condition is used.

8.3. Three-dimensional flow over an elastic structure

A three-dimensional incompressible flow is considered over a flexible beam fixed to a wall, discussed in [86]. The problem is considered to be symmetric in the x/y plane. Thus, the simulation is only performed in one-half of the domain. The width and height of the computational domain are 0.4 m, where the length is 1.5 m. The structure consists of a rectangular block with a width and height of 0.2 m, and a length of 0.1 m. The mesh of the flow domain is shown in Fig. 8.9.



(a) Velocity (b) Pressure

Figure 8.8.: Lid-driven cavity flow with moving block: time step study for integral deferred corrections. The temporal L_2 norm of the velocity and pressure error norm is shown.

8.3.1. Governing equation for the structural model

The configuration of the structure domain is described by the displacement $\boldsymbol{u}^s$. The structure is assumed to be elastic and compressible, and the governing equation is given by the balance of momentum

$$\rho^s \frac{\partial \boldsymbol{v}^s}{\partial t} + \rho^s \left(\nabla \boldsymbol{v}^s\right) \boldsymbol{v}^s = \nabla \cdot \boldsymbol{\sigma}^s + \rho^s \boldsymbol{g} \qquad \text{in } \Omega_t^s. \tag{8.12}$$

Eqn. (8.12) is modified to use the total Lagrangian description, i.e. with respect to the initial reference state Ω^s, resulting in

$$\rho^s \frac{\partial^2 \boldsymbol{u}^s}{\partial t^2} = \nabla \cdot \left(J \boldsymbol{\sigma}^s \boldsymbol{F}^{-T}\right) + \rho^s \boldsymbol{g} \qquad \text{in } \Omega^s, \tag{8.13}$$

where the deformation gradient tensor $\boldsymbol{F}$ is defined as $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}^s$, and the Jacobian J is the determinant of the deformation gradient tensor $\boldsymbol{F}$. By applying the constitutive law for the St. Venant-Kirchhoff material, the Cauchy stress tensor $\boldsymbol{\sigma}^s$ is found by applying

$$\boldsymbol{\sigma}^s = \frac{1}{J} \boldsymbol{F} \left(\lambda^s \left(\operatorname{tr} \boldsymbol{E}\right) \boldsymbol{I} + 2\mu^s \boldsymbol{E}\right) \boldsymbol{F}^T, \tag{8.14}$$

with $\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I}\right)$, and the shear modulus μ^s [24].

8.3.2. Numerical results

At the inflow boundary a parabolic velocity profile is imposed as a Dirichlet boundary condition with peak velocity $v_{\max} = 0.3\,\mathrm{m/s}$. On the outflow boundary, the zero gradient condition is imposed on the velocity and the pressure has a Dirichlet boundary condition with zero pressure. A no-slip condition is used on the remaining boundaries. The incompressible fluid has a density $\rho = 10^3\,\mathrm{kg/m^3}$, and kinematic viscosity $\nu = 10^{-3}\,\mathrm{m^2/s}$. The density of the solid is $10^3\,\mathrm{kg/m^3}$, the Poisson ratio $\nu^s = 0.4$ and the Young's modulus E is set to $10^4\,\mathrm{N/m^2}$, which is a smaller value as in [86] resulting in a larger displacement of the beam. Radial basis function interpolation is used to deform the fluid mesh. Anderson acceleration is used to perform the coupling iterations on the FSI interface. Matching grids are used at the FSI interface.

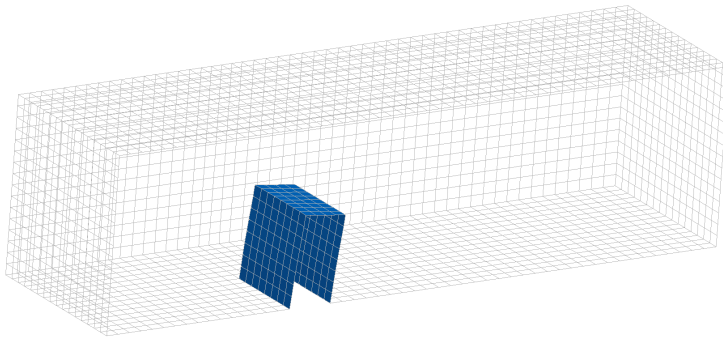


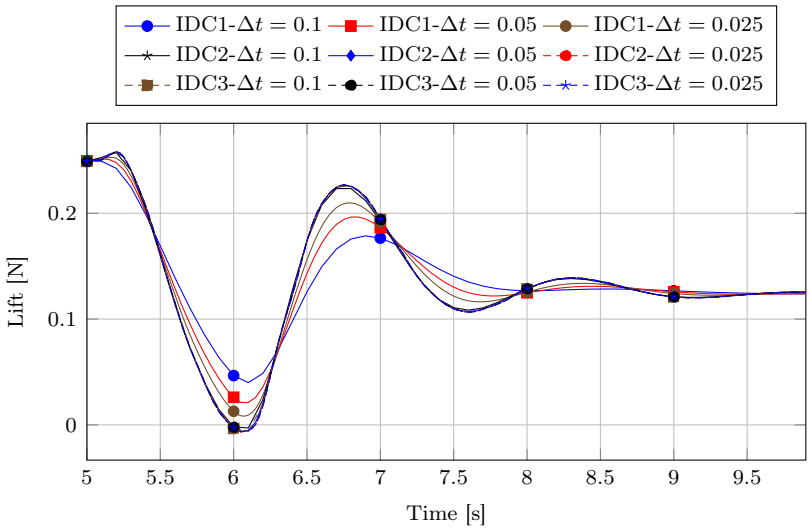
Figure 8.9.: Three-dimensional flow over an elastic structure. Computational grid of the flow domain.

From $t = 0\,\mathrm{s}$ until $t = 5\,\mathrm{s}$, the solid is assumed to be fixed. After that, a starting procedure is applied to use a smooth increase of the applied pressure and viscous force on the solid in time as

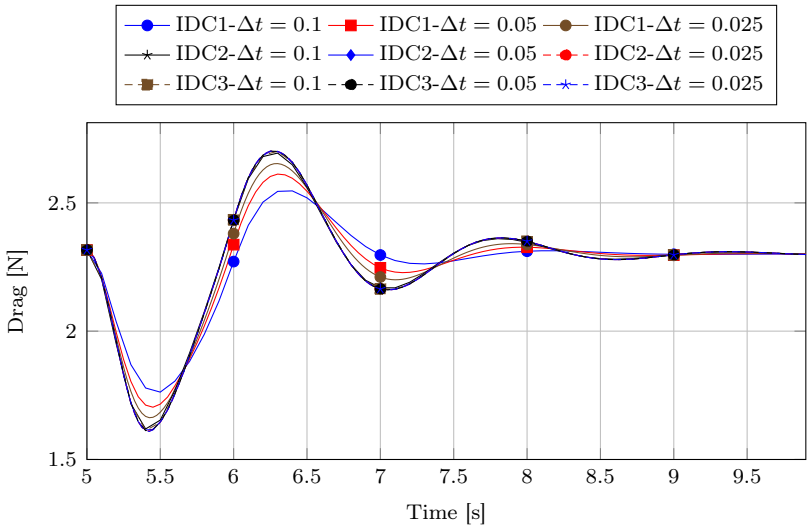
$$F(t) = \begin{cases} 0 & 0 < t <= 5 \\ F\left[0.5 - 0.5\cos\left(\pi\left(t-5\right)\right)\right] & 5 < t <= 6 \\ F & t > 6 \end{cases}, \tag{8.15}$$

where F represents the applied force on the structure.

Figure 8.10 shows the lift and drag over time for the different computations performed, and Fig. 8.11 shows the displacement of the top center point of the beam. As is to be expected, a large difference in lift and drag is observed when comparing the first and higher order time integration schemes. Those large differences highlight the importance of using a higher order time integration scheme for fluid-structure interaction simulations to accurately predict important parameters as lift, drag, and displacements.

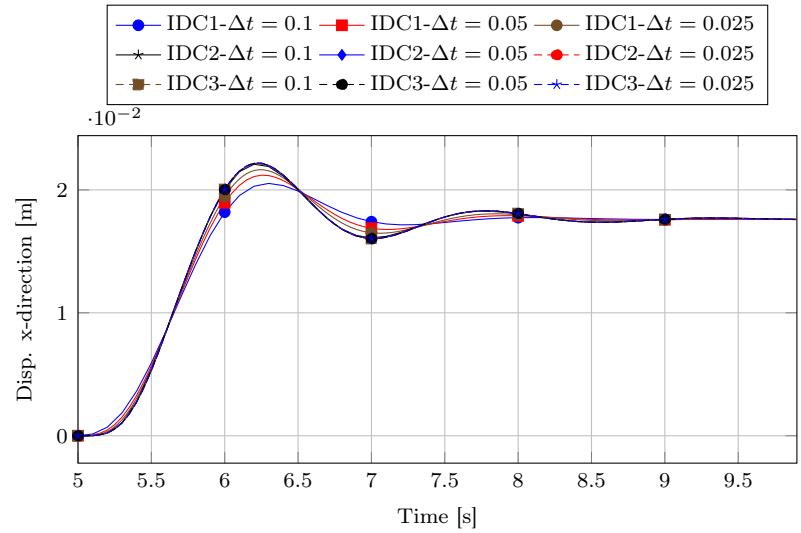


(a) Lift

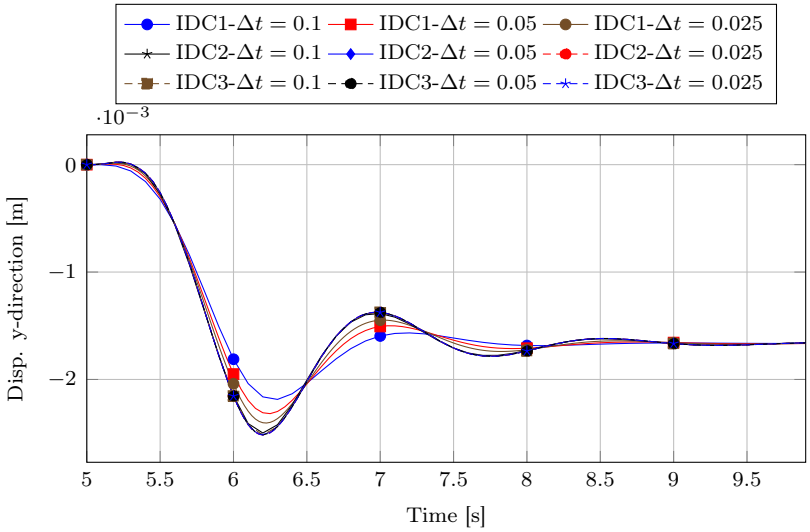


(b) Drag

Figure 8.10.: Lift and drag results for the flexible beam in a channel obtained with first, second and third order time integration schemes. Results for the time steps $\Delta t = [0.1, 0.05, 0.025]$ are shown.



(a) Displacement in x-direction



(b) Displacement in y-direction

Figure 8.11.: Displacement of the top-center point for the flexible beam in a channel obtained with first, second and third order time integration schemes

Chapter 9.

Conclusions and recommendations for arbitrarily high order time integration

> When everything seems to be going against you, remember that the airplane takes off against the wind, not with it.
>
> *(Henry Ford)*

9.1. Incompressible flow

Conclusions We demonstrated the use of arbitrarily high order time discretization schemes for the incompressible Navier-Stokes equations. The iterated PISO algorithm on collocated grids is used to solve the non-linear system of equations. The temporal derivative in the momentum equation is discretized with the method of lines using the chosen time integration schemes. A temporally consistent approximation is found by making a distinction between the spatial contribution and temporal contribution of the velocity flux.

The accuracy and efficiency of the SDC, IDC, and PIES methods have been demonstrated for several numerical examples. The main advantage of the deferred correction schemes over diagonally-implicit Runge-Kutta schemes is the fact that the coefficients necessary to reach a desired order of accuracy are easily obtained by using the quadrature integration rules, instead of manually deriving the correct coefficients by hand for the diagonally-implicit Runge-Kutta schemes. Note that when the left-end point is not included, and uniform sub-steps are applied (IDC), the

stability of the time integration is always guaranteed (L-stability).

Due to the iterative nature of the spectral deferred correction method, it is not necessary to solve the inner non-linear and linear systems to a high accuracy. One can apply a relative residual approach or a fixed number of iterations such that the current approximation is improved to some extent. In this case, the SDC loop controls the overall accuracy of the solution procedure, and the inner systems are not over-solved.

For moderate accuracy, the (E)SDIRK methods proved to be competitive to SDC and IDC. The flow around a cylinder case showed the high accuracy of the SDC method.

Recommendations One way to further increase the efficiency of the current implementation is to use a higher order predictor. For example, a second order backward differencing scheme, or even a third or fourth order (E)SDIRK scheme to decrease the number of correction sweeps to converge to a higher order solution. Another improvement would be to interweave the DIRK and SDC methods as proposed in Weiser [98] to reach faster convergence of the correction sweeps.

An extension of SDC time integration is parallel in time computation, as discussed by Emmett and Minion [45]. Since the SDC method is iterative by nature, it can be combined with a multi-grid approach in time. The coarse grid problems in time can work on lower order approximations to decrease the computational costs of the prediction and correction sweeps.

One important property of the integral and deferred spectral method is the fact that a continuous representation of the solution in between the sub-steps is readily available by using the interpolating polynomials. In the case of a partitioned simulation consisting of multiple domains which are only coupled in one direction, this property can be exploited. One example where this is advantageous is where a fluid simulation is coupled to an acoustics solver at the boundaries of the fluid domain. Generally, the acoustics solver imposes very strict time restrictions on the used time step in order to resolve the sound waves. Therefore, a standard approach is to employ sub-cycling to be able to use a relatively large time step for the fluid domain, and a small time step for the acoustics. This methodology has the disadvantage that the order of convergence in time is reduced to one since the boundary values are kept constant during the sub-cycling

process. However, when a continuous representation in time is available, which is the case for the spectral deferred correction scheme, higher order of convergence can be recovered.

9.2. Partitioned fluid-structure interaction simulations

Conclusions The integral deferred correction method has been applied to partitioned fluid-structure interaction problems, and incompressible flows with a moving object or moving grids. Since higher order time integration schemes deliver much higher accuracy compared to first or second order methods, we have shown the solution procedure for partitioned fluid-structure interaction problems.

To preserve uniform flow with a deforming mesh, we show how the Geometric Conservation Law (GCL) can be discretized with the IDC or SDC method. The incompressible Navier-Stokes equations are solved with the finite volume method on a collocated grid and coupled to the finite element solver for the solid domain. The theoretical orders of accuracy are achieved for the velocity as well as the pressure in case of the incompressible Navier-Stokes equations.

The numerical results show an order reduction for the SDIRK methods for which the coefficients do not satisfy order conditions derived from the Prothero-Robinson example, whereas the integral deferred correction achieves the expected order of accuracy. Concluding, integral deferred corrections are a good candidate for partitioned fluid-structure interaction simulations. The integral deferred correction method can be combined with a lower order scheme for the structure resulting in an accurate solution for the flow and sufficient accuracy for the structure.

Recommendations As recommended in Sect. 9.1, one way to further increase the efficiency of the current implementation is to use a higher order predictor. An idea one might have to further enhance the computational efficiency of a simulation is to combine the manifold mapping coupling scheme with the multigrid SDC time integration scheme. So instead of using a multigrid solver, the manifold mapping scheme can also be used.

Appendix A.

Coefficients for the Picard Integral Exponential Solver

Table A.1 shows the coefficients for the node-to-node integration of the Picard Integral Exponential Solver with $\rho = 3.15$ and $\delta = 1 \cdot 10^{-2}$. Coefficients for schemes with smaller values of δ are not shown, due to a large number of coefficients.

Table A.1.: Coefficients for the node-to-node integration of the Picard Integral Exponential Solver with $\rho = 3.15$ and $\delta = 1 \cdot 10^{-2}$.

(0,0) =	6.93889390390723e-18
(0,1) =	6.16911633400483e-01
(0,2) =	-1.67658786215060e+00
(0,3) =	3.10782377858074e+00
(0,4) =	-3.40459689015166e+00
(0,5) =	2.25518192294052e+00
(0,6) =	-7.32065915952810e-01
(1,0) =	-6.50521303491303e-18
(1,1) =	5.03446636091372e-02
(1,2) =	1.93708907900272e-01
(1,3) =	-1.64811006852359e-01
(1,4) =	1.50424890532760e-01
(1,5) =	-9.02230964025978e-02
(1,6) =	2.72223078794539e-02
(2,0) =	-3.68628738645072e-18
(2,1) =	-1.87209635497346e-03
(2,2) =	6.60817094651747e-02
(2,3) =	1.37492583518608e-01
(2,4) =	-5.48383646374622e-02
(2,5) =	2.71779234565757e-02
(2,6) =	-7.37508878125622e-03
(3,0) =	-2.58040117051550e-17
(3,1) =	5.07189797651921e-04
(3,2) =	-6.13558903373280e-03
(3,3) =	8.13118797030042e-02
(3,4) =	1.07698709926562e-01
(3,5) =	-2.14482006501728e-02
(3,6) =	4.73267692335439e-03
(4,0) =	-4.49943901581484e-18
(4,1) =	-3.25469363461508e-04
(4,2) =	3.24312071484101e-03
(4,3) =	-1.59089595464406e-02
(4,4) =	1.00430941478203e-01
(4,5) =	8.62718683987311e-02
(4,6) =	-7.04483501520615e-03
(5,0) =	-1.25767452008319e-17
(5,1) =	4.84478024852252e-04
(5,2) =	-4.39804440645361e-03
(5,3) =	1.77912895249151e-02
(5,4) =	-4.43686744251683e-02
(5,5) =	1.32325906189536e-01
(5,6) =	6.48317117589854e-02

Appendix B.

Butcher tableaus for the SDIRK methods

The used coefficients for the SDIRK2, SDIRK3, SDIRK4, and SDIRK2PR schemes can be found in Tables B.1, B.2, B.3 and B.4, respectively.

Table B.1.: Butcher tableau for the method of Ellsiepen [44] (SDIRK2), where $\alpha = 1 - \sqrt{2}/2$, $\hat{\alpha} = 2 - \frac{5}{4}\sqrt{2}$ and $\alpha - \hat{\alpha} = -1 + \frac{3}{4}\sqrt{2}$.

α	α	0
1	$1-\alpha$	α
b_i	$1-\alpha$	α
$\hat{b_i}$	$1-\hat{\alpha}$	$\hat{\alpha}$
$b_i - \hat{b_i}$	$\hat{\alpha} - \alpha$	$\alpha - \hat{\alpha}$

Table B.2.: Butcher tableau for SDIRK3, as introduced by Cash [25]

c_1	0.435866521508458	0	0
c_2	0.282066739245771	0.435866521508458	0
c_3	1.20849664917601	-0.644363170684469	0.435866521508458
b_i	1.20849664917601	-0.644363170684469	0.435866521508458
$\hat{b}_i$	0.772630127667551	0.227369872332449	0

Table B.3.: Butcher tableau for SDIRK4, as introduced by Cash [25]

c_1	0.435866521508	0	0	0	0
c_2	-1.1358665215	0.435866521508	0	0	0
c_3	1.08543330679	-0.721299828287	0.435866521508	0	0
c_4	0.416349501547	0.190984004184	-0.118643265417	0.435866521508	0
c_5	0.896869652944	0.0182725272734	-0.0845900310706	-0.266418670647	0.435866521508
b_i	0.896869652944	0.0182725272734	-0.0845900310706	-0.266418670647	0.435866521508
$\hat{b}_i$	0.77669193291	0.0297472791484	-0.0267440239074	0.220304811849	0

Table B.4.: Butcher tableau for SDIRK2PR, as introduced by Rang [84]

c_1	0.237286219578241	0	0
c_2	0.762713780421759	0.237286219578241	0
c_3	0.655553908732991	0.107159871688768	0.237286219578241
b_i	0.655553908732991	0.107159871688768	0.237286219578241
$\hat{b}_i$	0.762713780421759	0.237286219578241	0

Appendix C.

Butcher tableaus for the ESDIRK methods

The used coefficients for the ESDIRK3, ESDIRK4, ESDIRK5, ESDIRK53PR, ESDIRK63PR and ESDIRK74PR schemes can be found in Tables C.1, C.2, C.3, C.5 and C.6, respectively.

Table C.1.: Butcher tableau for ESDIRK3, as introduced by Kennedy and Carpenter [67]

c_1	0	0	0	0
c_2	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
c_3	$\frac{2746238789719}{10658868560708}$	$-\frac{640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
c_4	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{1126623926642 8}{1159328672282 1}$	$\frac{1767732205903}{4055673282236}$
b_i	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{1126623926642 8}{1159328672282 1}$	$\frac{1767732205903}{4055673282236}$
$\hat{b}_i$	$\frac{2756255671327}{12835298489170}$	$-\frac{1077155257357 5}{2220195857719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

Table C.2.: Butcher tableau for ESDIRK4, as introduced by Kennedy and Carpenter [67]

c_1	0	0	0	0	0	0
c_2	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
c_3	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$\frac{1}{4}$	0	0	0
c_4	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
c_5	$\frac{15267082809}{155376265600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
c_6	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
b_i	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
$\hat{b}_i$	$\frac{4586570599}{29645900160}$	0	$\frac{178811875}{945068544}$	$\frac{814220225}{1159782912}$	$-\frac{3700637}{11593932}$	$\frac{61727}{225920}$

Table C.3.: Butcher tableau for ESDIRK5, as introduced by Kennedy and Carpenter [67]

c_1	0	0	0	0	0	0	0
c_2	$\frac{41}{200}$	$\frac{41}{200}$	0	0	0	0	0
c_3	$\frac{41}{200}$	$-\frac{567603406766}{11931857230679}$	$\frac{41}{200}$	0	0	0	0
c_4	$\frac{683785636431}{9252920307686}$	0	$-\frac{110385047103}{1367015193373}$	$\frac{41}{200}$	0	0	0
c_5	$\frac{3016520224154}{10081342136671}$	0	$\frac{3058625980659}{12414158314087}$	$-\frac{2276050940356}{11113319521817}$	$\frac{41}{200}$	0	0
c_6	$\frac{2188664479029}{1489978393911}$	0	$\frac{638256894668}{5436446318841}$	$-\frac{1179710474555}{5321154724896}$	$-\frac{60928119172}{8023461067671}$	$\frac{41}{200}$	0
c_7	$\frac{10200423063}{571567683565}$	0	$\frac{25762820946817}{25263940353407}$	$-\frac{2161375909145}{9755907335909}$	$-\frac{211217309593}{5846859502534}$	$-\frac{4269925059573}{7827059040749}$	$\frac{41}{200}$
c_8	$-\frac{872700587467}{9133579230613}$	0	0	$\frac{2234821806326}{9555858737531}$	$-\frac{1143369518992}{8141816002931}$	$-\frac{3937952678962}{1901852630454}$	$\frac{327273823243}{429000448657}$
b_i	$-\frac{872700587467}{9133579230613}$	0	0	$\frac{2234821806326}{9555858737531}$	$-\frac{1143369518992}{8141816002931}$	$-\frac{3937952678962}{1901852630454}$	$\frac{327273823243}{429000448657}$
$\hat{b}_i$	$-\frac{97546191856}{9796059967033}$	0	0	$\frac{7807052710429}{3243259014707}$	$-\frac{5483825808}{3424219808633}$	$-\frac{3343884032128}{1559475310547}$	$\frac{362980080159}{465618377360}$

c_1	0
c_2	0
c_3	0
c_4	0
c_5	0
c_6	0
c_7	0
c_8	$\frac{41}{200}$
b_i	$\frac{41}{200}$
$\hat{b}_i$	$\frac{4035322873751}{18575991585200}$

Table C.4.: Butcher tableau for ESDIRK53PR, as introduced by Rang [83]

c_1	0	0	0	0	0
c_2	0.277777777777778	0.277777777777778	0	0	0
c_3	0.345655248351927	0.168174031571773	0.277777777777778	0	0
c_4	0.39656430472574	0.100115440493253	0.125542477003229	0.277777777777778	0
c_5	0.248147982878014	0.213947358893596	1.2062742392674	-0.946147358816787	0.277777777777778
b_i	0.248147982878014	0.213947358893596	1.2062742392674	-0.946147358816787	0.277777777777778
$\hat{b}_i$	0.444553753271355	-0.1065203443759	0.253312906975529	0.5	-0.091346315870985

Table C.5.: Butcher tableau for ESDIRK63PR, as introduced by Rang [83]

c_1	0						0
c_2	0.41666666666666667	0.41666666666666667					0
c_3	0.36404739157230 4	-0.041898861353313	0.41666666666666667				0
c_4	-2.894969214392781	-22.563417180646589	25.341719728372709	0.41666666666666667			0
c_5	0.230955102278 21	-1.849667242832423	2.197073089164931	0.00497238472261 5	0.41666666666666667		0
c_6	0.305496837846611	4.057983152922798	-2.202162095667 91	0.133348442927354	-1.711333004695519	0.41666666666666667	
b_i	0.305496837846611	4.057983152922798	-2.202162095667 91	0.133348442927354	-1.711333004695519	0.41666666666666667	0
$\hat{b}_i$	0.230955102278 21	-1.849667242832423	2.197073089164931	0.00497238472261 5	0.41666666666666667	0	

Table C.6.: Butcher tableau for ESDIRK74PR, as introduced by Rang [83]

c_1	0	0	0	0	0	0	0
c_2	0.166666666666667	0.166666666666667	0	0	0	0	0
c_3	0.041666666666667	-0.041666666666667	0.166666666666667	0	0	0	0
c_4	-1.5	-1.333333333333333	3.333333333333333	0.166666666666667	0	0	0
c_5	-1.580729166666667	-1.349609375	3.47265625	0.041015625	0.166666666666667	0	0
c_6	-2.005366150605651	-1.768688648609954	4.34126929534569	0.023261694346106	0.1	0.166666666666667	0
c_7	0.168485426780582	0.750108089883184	-0.225584388968693	-0.91344215042674	1.618140253772232	-0.564373897707231	0.166666666666667
b_i	0.168485426780582	0.750108089883184	-0.225584388968693	-0.91344215042674	1.618140253772232	-0.564373897707231	0.166666666666667
$\hat{b}_i$	-0.393018246175173	0.1	0.991634640557547	0	-0.251123215852894	0.439391281049749	0.113115540420771

Bibliography

[1] B. Alessandrini and G. Delhommeau. "Simulation of 3-Dimensional Unsteady Viscous Free-Surface Flow around a Ship Model". In: *International Journal for Numerical Methods in Fluids* 19.4 (1994), pp. 321–342. DOI: 10.1007/s00773-008-0012-7.

[2] D. G. Anderson. "Iterative Procedures for Nonlinear Integral Equations". In: *Journal of the ACM* 12.4 (1965), pp. 547–560. DOI: 10.1145/321296.321305.

[3] S. Badia, F. Nobile, and C. Vergara. "Fluid-structure partitioned procedures based on Robin transmission conditions". In: *Journal of Computational Physics* 227.14 (2008), pp. 7027–7051. DOI: 10.1016/j.jcp.2008.04.006.

[4] J. W. Bandler, R. M. Biernacki, S. H. Chen, R. H. Hemmers, and K. Madsen. "Electromagnetic Optimization Exploiting Aggressive Space Mapping". In: *IEEE Transactions on Microwave Theory and Techniques* 43.12 (1995), pp. 2874–2882. DOI: 10.1109/22.475649.

[5] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard. "Space mapping: the state of the art". In: *Microwave Theory and Techniques, IEEE Transactions on* 52.1 (2004), pp. 337–361. DOI: 10.1109/TMTT.2003.820904.

[6] J. Bandler, R. Biernacki, Shao Hua Chen, P. Grobelny, and R. Hemmers. "Space mapping technique for electromagnetic optimization". In: *IEEE Transactions on Microwave Theory and Techniques* 42.12 (1994), pp. 2536–2544. DOI: 10.1109/22.339794.

[7] A. T. Barker and X.-C. Cai. "Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling". In: *Journal of Computational Physics* 229.3 (2010), pp. 642–659. DOI: 10.1016/j.jcp.2009.10.001.

[8] F. Bassi, A. Crivellini, D. A. Di Pietro, and S. Rebay. "An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows". In: *Computers and Fluids* 36.10 (2007), pp. 1529–1546. DOI: 10.1016/j.compfluid.2007.03.012.

[9] K. J. Bathe and G. A. Ledezma. "Benchmark problems for incompressible fluid flows with structural interactions". In: *Computers and Structures* 85.11-14 (2007), pp. 628–644. DOI: 10.1016/j.compstruc.2007.01.025.

[10] K. J. Bathe and H. Zhang. "A mesh adaptivity procedure for CFD and fluid-structure interactions". In: *Computers and Structures* 87.11-12 (2009), pp. 604–617. DOI: 10.1016/j.compstruc.2009.01.017.

[11] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. R. Hughes. "Isogeometric Fluid-structure Interaction Analysis with Applications to Arterial Blood Flow". In: *Computational Mechanics* 38.4-5 (2006), pp. 310–322. DOI: 10.1007/s00466-006-0084-3.

[12] H. Bijl, M. H. Carpenter, V. N. Vatsa, and C. A. Kennedy. "Implicit Time Integration Schemes for the Unsteady Compressible Navier-Stokes Equations: Laminar Flow". In: *Journal of Computational Physics* 179 (2002), pp. 313–329. DOI: 10.1006/jcph.2002.7059.

[13] D. S. Blom, B. Uekermann, M. Mehl, A. H. van Zuijlen, and H. Bijl. "Multi-Level Acceleration of Parallel Coupled Partitioned Fluid-Structure Interaction with Manifold Mapping". In: *Lecture Notes in Computational Science and Engineering*. Vol. 105. Springer International Publishing, 2015, pp. 135–150. DOI: 10.1007/978-3-319-22997-3_8.

[14] D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 296 (2015), pp. 211–231. DOI: 10.1016/j.cma.2015.08.004.

[15] D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Acceleration of strongly coupled fluid-structure interaction with manifold mapping". In: *Proceedings of WCCM XI, Proceedings of ECCM V, Proceedings of ECFD VI* WCCM XI (2014), pp. 1–12.

[16] D. S. Blom, V. Kazemi-Kamyab, T. Gillebaart, A. H. van Zuijlen, and H. Bijl. "On the application of spectral deferred corrections to incompressible flow on unstructured grids". In: *Journal of Computational Physics* under review (2016).

[17] D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using integral deferred corrections". In: *Journal of Computational Physics* under review (2016).

[18] A. de Boer, A. H. van Zuijlen, and H. Bijl. "Radial Basis Functions for Interface Interpolation and Mesh Deformation". In: *Advanced Computational Methods in Science and Engineering*. Ed. by B. Koren and K. Vuik. Springer Berlin Heidelberg, 2009, pp. 143–178. DOI: 10.1007/978-3-642-03344-5_6.

[19] A. E. J. Bogaers, S. Kok, B. D. Reddy, and T. Franz. "Quasi-Newton methods for implicit black-box FSI coupling". In: *Computer Methods in Applied Mechanics and Engineering* 279 (2014), pp. 113–132. DOI: 10.1016/j.cma.2014.06.033.

[20] A. Bourlioux, A. T. Layton, and M. L. Minion. "High-order multi-implicit spectral deferred correction methods for problems of reactive flow". In: *Journal of Computational Physics* 189.2 (2003), pp. 651–675. DOI: 10.1016/S0021-9991(03)00251-1.

[21] E. H. van Brummelen. "Added Mass Effects of Compressible and Incompressible Flows in Fluid-Structure Interaction". In: *Journal of Applied Mechanics* 76.March 2009 (2009), p. 021206. DOI: 10.1115/1.3059565.

[22] E. H. van Brummelen. "Partitioned iterative solution methods for fluid-structure interaction". In: *International Journal for Numerical Methods in Fluids* 65.1-3 (2011), pp. 3–27. DOI: 10.1002/fld.2465.

[23] E. van Brummelen, K. van der Zee, and R. de Borst. "Space/time multigrid for a fluid-structure-interaction problem". In: *Applied Numerical Mathematics* 58.12 (2008), pp. 1951–1971. DOI: 10.1016/j.apnum.2007.11.012.

[24] P. Cardiff, A. Karac, and A. Ivankovic. "A large strain finite volume method for orthotropic bodies with general material orientations". In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014), pp. 318–335. DOI: 10.1016/j.cma.2013.09.008.

[25] J. R. Cash. "Diagonally Implicit Runge-Kutta formulae with error estimates". In: *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)* 24.3 (1979), pp. 293–301. DOI: 10.1093/imamat/24.3.293.

[26] M. Cervera, R. Codina, and M. Galindo. "On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems". In: *Engineering Computations* 13.6 (1996), pp. 4–30. DOI: 10.1108/02644409610128382.

[27] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin. "On the Compression of Low Rank Matrices". In: *SIAM Journal on Scientific Computing* 26.4 (2005), pp. 1389–1404. DOI: 10.1137/030602678.

[28] A. Christlieb, W. Guo, M. Morton, and J. M. Qiu. "A high order time splitting method based on integral deferred correction for semi-Lagrangian Vlasov simulations". In: *Journal of Computational Physics* 267 (2014), pp. 7–27. DOI: 10.1016/j.jcp.2014.02.012.

[29] A. Christlieb, B. Ong, and J.-M. Qiu. "Integral deferred correction methods constructed with high order Runge-Kutta integrators". In: *Mathematics of Computation* 79.270 (2009), pp. 761–783. DOI: 10.1090/S0025-5718-09-02276-5.

[30] G. R. Cowper. "The Shear Coefficient in Timoshenko's Beam Theory". In: *Journal of Applied Mechanics* 33.2 (1966), pp. 335–340. DOI: 10.1115/1.3625046.

[31] M. Darwish, I. Sraj, and F. Moukalled. "A coupled finite volume solver for the solution of incompressible flows on unstructured grids". In: *Journal of Computational Physics* 228.1 (2009), pp. 180–201. DOI: 10.1016/j.jcp.2008.08.027.

[32] J. Degroote. "Partitioned Simulation of Fluid-Structure Interaction". In: *Archives of Computational Methods in Engineering* 20.3 (2013), pp. 185–238. DOI: 10.1007/s11831-013-9085-5.

[33] J. Degroote, K.-J. Bathe, and J. Vierendeels. "Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction". In: *Computers & Structures* 87.11-12 (2009), pp. 793–801. DOI: 10.1016/j.compstruc.2008.11.013.

[34] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. "Stability of a coupling technique for partitioned solvers in FSI applications". In: *Computers & Structures* 86.23-24 (2008), pp. 2224–2234. DOI: 10.1016/j.compstruc.2008.05.005.

[35] J. Degroote, M. Hojjat, E. Stavropoulou, R. W\"{u}chner, and K.-U. Bletzinger. "Partitioned solution of an unsteady adjoint for strongly coupled fluid-structure interactions and application to parameter identification of a one-dimensional problem". In: *Structural and Multidisciplinary Optimization* 47.1 (2013), pp. 77–94. DOI: 10.1007/s00158-012-0808-2.

[36] J. Degroote and J. Vierendeels. "Multi-level quasi-Newton coupling algorithms for the partitioned simulation of fluid-structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 225-228 (2012), pp. 14–27. DOI: 10.1016/j.cma.2012.03.010.

[37] J. Degroote and J. Vierendeels. "Multi-solver algorithms for the partitioned simulation of fluid-structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 200.25-28 (2011), pp. 2195–2210. DOI: 10.1016/j.cma.2011.03.015.

[38] B. Delinchant, D. Lahaye, F. Wurtz, and J.-L. Coulomb. "Manifold mapping optimization with or without true gradients". In: *Mathematics and Computers in Simulation* 90 (2013), pp. 256–265. DOI: 10.1016/j.matcom.2012.09.005.

[39] I. Demirdzic and M. Peric. "Space conservation law in finite volume calculations of fluid flow". In: *International Journal for Numerical Methods in Fluids* 8.9 (1988), pp. 1037–1050. DOI: 10.1002/fld.1650080906.

[40] A. Dutt, L. Greengard, and V. Rokhlin. "Spectral deferred correction methods for ordinary differential equations". In: *BIT Numerical Mathematics* 40.2 (2000), pp. 241–266. DOI: 10.1023/A:1022338906936.

[41] D. Echeverria and P. W. Hemker. "Manifold mapping: a two-level optimization technique". In: *Computing and Visualization in Science* 11.4-6 (2008), pp. 193–206. DOI: 10.1007/s00791-008-0096-y.

[42] D. Echeverria, D. Lahaye, L. Encica, E. Lomonova, P. Hemker, and A. Vandenput. "Manifold-mapping optimization applied to linear actuator design". In: *IEEE Transactions on Magnetics* 42.4 (2006), pp. 1183–1186. DOI: 10.1109/TMAG.2006.870969.

[43] D. Echeverria, D. Lahaye, and P. W. Hemker. "Space Mapping and Defect Correction". In: *Model Order Reduction: Theory, Research Aspects and Applications*. Ed. by W. H. A. Schilders, H. A. van der Vorst, and J. Rommes. Vol. 13. Springer Berlin Heidelberg, 2008, pp. 157–176. DOI: 10.1007/978-3-540-78841-6_8.

[44] P. Ellsiepen. "Zeits- und ortsadaptive Verfahren angewandt auf Mehrphasenprobleme poroser Medien". PhD thesis. University of Stuttgart, 1999.

[45] M. Emmett and M. L. Minion. "Toward an efficient parallel in time method for partial differential equations". In: *Communications in Applied Mathematics and Computational Science* 7.1 (2012), pp. 105–132. DOI: 10.2140/camcos.2012.7.105.

[46] H. R. Fang and Y. Saad. "Two classes of multisecant methods for nonlinear acceleration". In: *Numerical Linear Algebra with Applications* 16.3 (2009), pp. 197–221. DOI: 10.1002/nla.617.

[47] C. Farhat and M. Lesoinne. "Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems". In: *Computer Methods in Applied Mechanics and Engineering* 182.3-4 (2000), pp. 499–515. DOI: 10.1016/S0045-7825(99)00206-6.

[48] C. A. Felippa, K. C. Park, and C. Farhat. "Partitioned analysis of coupled mechanical systems". In: *Computer Methods in Applied Mechanics and Engineering* 190.24-25 (2001), pp. 3247–3270. DOI: 10.1016/S0045-7825(00)00391-1.

[49] M. A. Fernandez and M. Moubachir. "A Newton method using exact jacobians for solving fluid-structure coupling". In: *Computers & Structures* 83.2-3 (2005), pp. 127–142. DOI: 10.1016/j.compstruc.2004.04.021.

[50] V. Ganine, N. J. Hills, and B. L. Lapworth. "Nonlinear acceleration of coupled fluid-structure transient thermal problems by Anderson mixing". In: *International Journal for Numerical Methods in Fluids* 71.8 (2013), pp. 939–959. DOI: 10.1002/fld.3689.

[51] M. W. Gee, U. Kuttler, and W. A. Wall. "Truly monolithic algebraic multigrid for fluid-structure interaction". In: *International Journal for Numerical Methods in Engineering* 85.8 (2011), pp. 987–1016. DOI: 10.1002/nme.3001.

[52] J. Gerbeau and M. Vidrascu. "A Quasi-Newton Algorithm Based on a Reduced Model for Fluid-Structure Interaction Problems in Blood Flows". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 37.4 (2003), pp. 631–647. DOI: 10.1051/m2an:2003049.

[53] A. Glaser and V. Rokhlin. "A New Class of Highly Accurate Solvers for Ordinary Differential Equations". In: *Journal of Scientific Computing* 38.3 (2009), pp. 368–399. DOI: 10.1007/s10915-008-9245-1.

[54] P. M. Gresho and R. L. Sani. "On pressure boundary conditions for the incompressible Navier-Stokes equations". In: *International Journal for Numerical Methods in Fluids* 7.10 (1987), pp. 1111–1145. DOI: 10.1002/fld.1650071008.

[55] R. Haelterman, B. Lauwens, F. Van Utterbeeck, H. Bruyninckx, and J. Vierendeels. "On the similarities between the quasi-Newton least squares method and GMRes". In: *Journal of Computational and Applied Mathematics* 273 (2015), pp. 25–28. DOI: 10.1016/j.cam.2014.05.020.

[56] R. Haelterman, J. Degroote, D. Van Heule, and J. Vierendeels. "The Quasi-Newton Least Squares Method: A New and Fast Secant Method Analyzed for Linear Systems". In: *SIAM Journal on Numerical Analysis* 47.3 (2009), pp. 2347–2368. DOI: 10.1137/070710469.

[57] E. Hairer, M. Roche, and C. Lubich. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Vol. 1409. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1989. DOI: 10.1007/BFb0093947.

[58] E. Hairer and G. Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*. Vol. 14. Springer-Verlag Berlin Heidelberg, 1991, pp. xvi+601. DOI: 10.1007/978-3-642-05221-7.

[59] A. C. Hansen and J. Strain. "On the order of deferred correction". In: *Applied Numerical Mathematics* 61.8 (2011), pp. 961–973. DOI: 10.1016/j.apnum.2011.04.001.

[60] J. Hron and S. Turek. "A monolithic FEM/multigrid solver for an ALE formulation of fluid-structure interaction with applications in biomechanics". In: *Lecture Notes in Computational Science and Engineering* 53 (2006), pp. 146–170. DOI: 10.1007/3-540-34596-5_7.

[61] M.-C. Hsu and Y. Bazilevs. "Fluid-structure interaction modeling of wind turbines: simulating the full machine". In: *Computational Mechanics* 50 (2012), pp. 821–833. DOI: 10.1007/s00466-012-0772-0.

[62] J. Huang, J. Jia, and M. Minion. "Accelerating the convergence of spectral deferred correction methods". In: *Journal of Computational Physics* 214.2 (2006), pp. 633–656. DOI: 10.1016/j.jcp.2005.10.004.

[63] R. I. Issa. "Solution of the implicitly discretised fluid flow equations by operator-splitting". In: *Journal of Computational Physics* 62.1 (1986), pp. 40–65. DOI: 10.1016/0021-9991(86)90099-9.

[64] V. John. "Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder". In: *International Journal for Numerical Methods in Fluids* 44.7 (2004), pp. 777–788. DOI: 10.1002/fld.679.

[65] V. John and J. Rang. "Adaptive time step control for the incompressible Navier-Stokes equations". In: *Computer Methods in Applied Mechanics and Engineering* 199.9-12 (2010), pp. 514–524. DOI: 10.1016/j.cma.2009.10.005.

[66] V. Kazemi-Kamyab, A. H. van Zuijlen, and H. Bijl. "Analysis and application of high order implicit Runge-Kutta schemes to collocated finite volume discretization of the incompressible Navier-Stokes equations". In: *Computers & Fluids* 108 (2015), pp. 107–115. DOI: 10.1016/j.compfluid.2014.11.025.

[67] C. A. Kennedy and M. H. Carpenter. "Additive Runge-Kutta schemes for convection-diffusion-reaction equations". In: *Applied Numerical Mathematics* 44.1-2 (2003), pp. 139–181. DOI: 10.1016/S0168-9274(02)00138-1.

[68] S. Koziel, J. W. Bandler, and K. Madsen. "Towards a rigorous formulation of the space mapping technique for engineering design". In: *Proceedings - IEEE International Symposium on Circuits and Systems*. 2005, pp. 5605–5608. DOI: 10.1109/ISCAS.2005.1465908.

[69] D. Kushnir and V. Rokhlin. "A Highly Accurate Solver for Stiff Ordinary Differential Equations". In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1296–A1315. DOI: 10.1137/100810216.

[70] U. Kuttler and W. A. Wall. "Fixed-point fluid-structure interaction solvers with dynamic relaxation". In: *Computational Mechanics* 43.1 (2008), pp. 61–72. DOI: 10.1007/s00466-008-0255-5.

[71] U. Kuttler and W. A. Wall. "Vector Extrapolation for Strong Coupling Fluid-Structure Interaction Solvers". In: *Journal of Applied Mechanics* 76.March 2009 (2009), p. 021205. DOI: 10.1115/1.3057468.

[72] A. T. Layton and M. L. Minion. "Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics". In: *Journal of Computational Physics* 194.2 (2004), pp. 697–715. DOI: 10.1016/j.jcp.2003.09.010.

[73] A. T. Layton and M. L. Minion. "Implications of the Choice of Quadrature Nodes for Picard Integral Deferred Corrections Methods for Ordinary Differential Equations". In: *BIT Numerical Mathematics* 45.2 (2005), pp. 341–373. DOI: 10.1007/s10543-005-0016-1.

[74] F. Lien and M. Leschziner. "A general non-orthogonal collocated finite volume algorithm for turbulent flow at all speeds incorporating second-moment closure". In: *Computer Methods in Applied Mechanics and Engineering* 114 (1994), pp. 123–148. DOI: 10.1016/0045-7825(94)90166-X.

[75] P. O. Marklund and L. Nilsson. "Simulation of airbag inflation processes using a coupled fluid structure approach". In: *Computational Mechanics* 29.4-5 (2002), pp. 289–297. DOI: 10.1007/s00466-002-0341-z.

[76] M. Mehl, B. Uekermann, H. Bijl, D. S. Blom, B. Gatzhammer, and A. H. van Zuijlen. "Parallel coupling numerics for partitioned fluid-structure interaction simulations". In: *Computers & Mathematics with Applications* 71.4 (2016), pp. 869–891. DOI: 10.1016/j.camwa.2015.12.025.

[77] C. Michler, E. H. van Brummelen, and R. de Borst. "An interface Newton-Krylov solver for fluid-structure interaction". In: *International Journal for Numerical Methods in Fluids* 47.10-11 (2005), pp. 1189–1195. DOI: 10.1002/fld.850.

[78] C. Michler, E. H. van Brummelen, and R. De Borst. "An investigation of Interface-GMRES(R) for fluid-structure interaction problems with flutter and divergence". In: *Computational Mechanics* 47.1 (2011), pp. 17–29. DOI: 10.1007/s00466-010-0519-8.

[79] M. Minion. "A hybrid parareal spectral deferred corrections method". In: *Communications in Applied Mathematics and Computational Science* 5.2 (2010), pp. 265–301. DOI: 10.2140/camcos.2010.5.265.

[80] M. L. Minion. "Semi-implicit projection methods for incompressible flow based on spectral deferred corrections". In: *Applied Numerical Mathematics*. Vol. 48. 2004, pp. 369–387. DOI: 10.1016/j.apnum.2003.11.005.

[81] A. Montlaur, S. Fernandez-Mendez, and A. Huerta. "High-order implicit time integration for unsteady incompressible flows". In: *International Journal for Numerical Methods in Fluids* 70.5 (2012), pp. 603–626. DOI: 10.1002/fld.2703.

[82] S. Piperno. "Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations". In: *International Journal for Numerical Methods in Fluids* 25.10 (1997), pp. 1207–1226. DOI: 10.1002/(SICI)1097-0363(19971130)25:10<1207::AID-FLD616>3.0.CO;2-R.

[83] J. Rang. "An analysis of the Prothero-Robinson example for constructing new DIRK and ROW methods". In: *Journal of Computational and Applied Mathematics* 262 (2014), pp. 105–114. DOI: 10.1016/j.cam.2013.09.062.

[84] J. Rang. "The Prothero and Robinson example: Convergence studies for Runge-Kutta and Rosenbrock-Wanner methods". In: *Applied Numerical Mathematics* 108 (2016), pp. 37–56. DOI: 10.1016/j.apnum.2016.04.012.

[85] C. Rhie and W. Chow. "A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation". In: *3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference*. Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, 1982, pp. 1–11. DOI: 10.2514/6.1982-998.

[86] T. Richter. "Goal-oriented error estimation for fluid-structure interaction problems". In: *Computer Methods in Applied Mechanics and Engineering* 223-224 (2012), pp. 28–42. DOI: 10.1016/j.cma.2012.02.014.

[87] T. Richter. "A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation". In: *International Journal for Numerical Methods in Engineering* 104.5 (2015), pp. 372–390. DOI: 10.1002/nme.4943.

[88] M. R. Ross, M. A. Sprague, C. A. Felippa, and K. C. Park. "Treatment of acoustic fluid-structure interaction by localized Lagrange multipliers and comparison to alternative interface-coupling methods". In: *Computer Methods in Applied Mechanics and Engineering* 198.9-12 (2009), pp. 986–1005. DOI: 10.1016/j.cma.2008.11.006.

[89] M. Schafer, S. Turek, F. Durst, E. Krause, and R. Rannacher. "Benchmark Computations of Laminar Flow Around a Cylinder". In: *Flow Simulation with High-Performance Computers II, Volume 48 of Notes on Numerical Fluid Mechanics (NNFM)*. Vol. 48. Vieweg + Teubner Verlag, 1996, pp. 547–566. DOI: 10.1007/978-3-322-89849-4_39.

[90] T. P. Scholcz, A. H. van Zuijlen, and H. Bijl. "Space-mapping in fluid-structure interaction problems". In: *Computer Methods in Applied Mechanics and Engineering* 281 (2014), pp. 162–183. DOI: 10.1016/j.cma.2014.07.028.

[91] W. Z. Shen, J. A. Michelsen, and J. N. Sorensen. "Improved Rhie-Chow interpolation for unsteady flow computations". In: *AIAA Journal* 39.12 (2001), pp. 2406–2409. DOI: 10.2514/3.15042.

[92] R. Speck, D. Ruprecht, M. Emmett, M. Minion, M. Bolten, and R. Krause. "A multi-level spectral deferred correction method". In: *BIT Numerical Mathematics* 55.3 (2015), pp. 843–867. DOI: 10.1007/s10543-014-0517-x.

[93] R. Speck, D. Ruprecht, M. Minion, M. Emmett, and R. Krause. "Inexact Spectral Deferred Corrections". In: *Lecture Notes in Computational Science and Engineering*. Springer International Publishing, 2016, pp. 389–396. DOI: 10.1007/978-3-319-18827-0_39.

[94] K. Stein, R. Benney, V. Kalro, T. E. Tezduyar, J. Leonard, and M. Accorsi. "Parachute fluid-structure interactions: 3-D computation". In: *Computer Methods in Applied Mechanics and Engineering* 190.3-4 (2000), pp. 373–386. DOI: 10.1016/S0045-7825(00)00208-5.

[95] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. "Space-time finite element techniques for computation of fluid-structure interactions". In: *Computer Methods in Applied Mechanics and Engineering* 195.17-18 (2006), pp. 2002–2027. DOI: 10.1016/j.cma.2004.09.014.

[96] S. Turek and J. Hron. "Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow". In: *Fluid-Structure Interaction*. Vol. 53. Springer Berlin Heidelberg, 2006, pp. 371–385. DOI: 10.1007/3-540-34596-5_15.

[97] H. F. Walker and P. Ni. "Anderson Acceleration for Fixed-Point Iterations". In: *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735. DOI: 10.1137/10078356X.

[98] M. Weiser. "Faster SDC convergence on non-equidistant grids by DIRK sweeps". In: *BIT Numerical Mathematics* 55.4 (2015), pp. 1219–1241. DOI: 10.1007/s10543-014-0540-y.

[99] B. Yu, Y. Kawaguchi, W. Tao, and H. Ozoe. "Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method". In: *Numerical Heat Transfer, Part B: Fundamentals* 41 (2002), pp. 85–94. DOI: 10.1080/104077902753385027.

[100] B. Yu, W. Tao, J. Wei, Y. Kawaguchi, T. Tagawa, and H. Ozoe. "Discussion on momentum interpolation method for collocated grids of incompressible flow". In: *Numerical Heat Transfer, Part B: Fundamentals* 42 (2002), pp. 141–166. DOI: 10.1080/10407790190053879.

[101] A. H. van Zuijlen and H. Bijl. "Multi-level accelerated sub-iterations for fluid-structure interaction". In: *Lecture Notes in Computational Science and Engineering*. Vol. 73 LNCSE. 2010, pp. 1–25. DOI: 10.1007/978-3-642-14206-2_1.

Acknowledgements

> It isn't what we say or think that defines us, but what we do.
>
> *(Jane Austen)*

I would like to thank my promotor Hester Bijl and my copromotor Alexander (Sander) van Zuijlen. I had the privilege to do both my master's thesis and now my Ph.D. thesis under their supervision. Numerous discussions with Sander have led to new insights, new ideas and better understandings of the research.

I would also like to thank Miriam Mehl, Sabine Roller, and Dörte Sternel for being great project leaders of the Exascale Simulation of Fluid-Structure-Acoustics Interactions (ExaFSA) project. Many times we could meet as a project somewhere in Germany and discuss how to go forward.

Of course, I should mention Thijs. Together we have built the FOAM-FSI library, looked at the consistent time integration schemes and mesh deformation techniques in OpenFOAM, which was a lot of fun. Many discussions with Benjamin, Florian, Klaudius, Verena and Oliver were of a great help running large simulations with preCICE on the SuperMUC system in Munich. Wouter, Liesbeth, Rogier, and Martin, thanks for the many conversations during lunch on whatever topic.

And I would like to thank my family. Alexander, thanks for the talks we had on a wide variety of subjects, serious and non-serious. Michiel, I particularly liked the trip to Leuven where we could talk, and of course had to drive through the parking lot of Zaventem. Mom and dad, many thanks for always supporting me in whatever I choose or decide to do. Thanks for being there, for listening and talking about everything.

Finally, I would like to thank my beautiful wife for her support. I must say, you are not just my wife, you are my life and my inspiration. All I can say is thank you.

List of publications

Journal articles

[1] D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction". In: *Computer Methods in Applied Mechanics and Engineering* 296 (2015), pp. 211–231. DOI: 10.1016/j.cma.2015.08.004.

[2] D. S. Blom, P. Birken, H. Bijl, F. Kessels, A. Meister, and A. H. van Zuijlen. "A comparison of Rosenbrock and ESDIRK methods combined with iterative solvers for unsteady compressible flows". In: *Advances in Computational Mathematics* 42.6 (2016), pp. 1401–1426. DOI: 10.1007/s10444-016-9468-x.

[3] D. S. Blom, V. Kazemi-Kamyab, T. Gillebaart, A. H. van Zuijlen, and H. Bijl. "On the application of spectral deferred corrections to incompressible flow on unstructured grids". In: *Journal of Computational Physics* under review (2016).

[4] D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Arbitrarily high order time integration for partitioned fluid-structure interaction simulations using integral deferred corrections". In: *Journal of Computational Physics* under review (2016).

[5] L. Florentie, D. S. Blom, T. P. Scholcz, A. H. van Zuijlen, and H. Bijl. "Analysis of space mapping algorithms for application to partitioned fluid-structure interaction problems". In: *International Journal for Numerical Methods in Engineering* 105.2 (2016), pp. 138–160. DOI: 10.1002/nme.4979.

[6] T. Gillebaart, D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Time consistent fluid structure interaction on collocated grids for incompressible flow". In: *Computer Methods in Applied Mechanics and Engineering* 298.298 (2016), pp. 159–182. DOI: 10.1016/j.cma.2015.09.025.

[7] T. Gillebaart, D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Adaptive radial basis function mesh deformation using data reduction". In: *Journal of Computational Physics* 321 (2016), pp. 997–1025. DOI: 10.1016/j.jcp.2016.05.036.

[8] M. Mehl, B. Uekermann, H. Bijl, D. S. Blom, B. Gatzhammer, and A. H. van Zuijlen. "Parallel coupling numerics for partitioned fluid-structure interaction simulations". In: *Computers & Mathematics with Applications* 71.4 (2016), pp. 869–891. DOI: 10.1016/j.camwa.2015.12.025.

Book chapters

[1] D. S. Blom, T. Ertl, O. Fernandes, S. Frey, H. Klimach, V. Krupp, M. Mehl, S. Roller, D. C. Sternel, B. Uekermann, T. Winter, and A. van Zuijlen. "Partitioned Fluid-Structure-Acoustics Interaction on Distributed Data: Numerical Results and Visualization". In: *Lecture Notes in Computational Science and Engineering*. 2016, pp. 267–291. DOI: 10.1007/978-3-319-40528-5_12.

[2] D. S. Blom, B. Uekermann, M. Mehl, A. H. van Zuijlen, and H. Bijl. "Multi-Level Acceleration of Parallel Coupled Partitioned Fluid-Structure Interaction with Manifold Mapping". In: *Lecture Notes in Computational Science and Engineering*. Vol. 105. Springer International Publishing, 2015, pp. 135–150. DOI: 10.1007/978-3-319-22997-3_8.

[3] D. S. Blom, F. Lindner, M. Mehl, K. Scheufele, B. Uekermann, and A. van Zuijlen. "A Review on Fast Quasi-Newton and Accelerated Fixed-Point Iterations for Partitioned Fluid-Structure Interaction Simulation". In: *Advances in Computational Fluid-Structure Interaction and Flow Simulation*. Ed. by Y. Bazilevs and K. Takizawa.

Birkhauser Basel, 2016, pp. 257–269. DOI: 10.1007/978-3-319-40827-9_20.

Conference proceedings

[1] D. S. Blom, H. Bijl, P. Birken, A. Meister, and A. H. van Zuijlen. "Rosenbrock time integration for unsteady flow simulations". In: *5th International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2013*. 2013, pp. 937–948.

[2] D. S. Blom, A. H. van Zuijlen, and H. Bijl. "Acceleration of strongly coupled fluid-structure interaction with manifold mapping". In: *Proceedings of WCCM XI, Proceedings of ECCM V, Proceedings of ECFD VI* WCCM XI (2014), pp. 1–12.

[3] D. S. Blom, V. Krupp, A. H. van Zuijlen, H. Klimach, S. Roller, and H. Bijl. "On parallel scalability aspects of strongly coupled partitioned fluid-structure-acoustics interaction". In: *6th International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2015; Venice; Italy.* 2015, pp. 556–565.

[4] O. Fernandes, D. S. Blom, S. Frey, A. H. van Zuijlen, H. Bijl, and T. Ertl. "On in-situ visualization for strongly coupled partitioned fluid-structure interaction". In: *6th International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2015; Venice; Italy.* 2015, pp. 544–555.

Curriculum Vitæ

David Blom

01 - 03 - 1989	Born in Bulawayo, Zimbabwe.

Education

2000 - 2006	Gymnasium Apeldoorn
2007 - 2010	B.Sc. in Aerospace Engineering Delft University of Technology
2010 - 2013	M.Sc. in Aerodynamics & Wind Energy Faculty of Aerospace Engineering Delft University of Technology
2013 - 2017	Ph.D. in Aerodynamics Faculty of Aerospace Engineering Delft University of Technology