

Contents

1	Numerical Experiments	3
1.1	Comparison of mesh moving models	3
1.2	Investigation of temporal stability	5
1.2.1	Results	6
1.3	Optimization of the Newtonsolver	9
1.3.1	Consistent methods	10
1.3.2	Non-consisten methods	10
1.3.3	Comparison of speedup methods	10

Numerical Experiments

1.1 Comparison of mesh moving models

Mesh moving models are essential for numerical stability of fluid-structure interaction solvers. If the fluid mesh doesn't conform with the solid deformation, the risk of mesh entanglement increases with the possibility of instabilities or breakdown of the solver. In general, mesh models have shown to be either robust concerning mesh entanglements at the cost of computational time, or computationally efficient with less robustness [24]. However, computational efficiency has proven not only to be dependent of the complexity of model, but also the regularity of the fluid mesh, reducing Newton iterations needed per time step [48].

In this section we compare the mesh moving models from section 3.1.4. for the FSI-3 benchmark. The linear elastic model was found not applicable in section 4.2.3. Therefore, only the Laplace and biharmonic model will be considered. We will compare vertical displacement of the elastic flag, regularity of the fluid mesh, and number of Newton iterations per time step. To evaluate the regularity of the fluid mesh, the minimum value of the jacobian of the deformation gradient have been considered in [48].

$$J_f = \det(\hat{\mathbf{F}}_f) = \det(I + \hat{\nabla} \hat{\mathbf{u}}_f)$$

where I is the identity matrix and $\hat{\mathbf{u}}_f$ is the fluid mesh deformation. The jacobian serves as a measure of mesh entanglement, meaning if $J_f \geq 0$, there are no crossing cells in the fluid mesh. A serial naive Newton solver is used, avoiding any effects of speed-up techniques which may effect Newton iterations (see section 6.3.3).

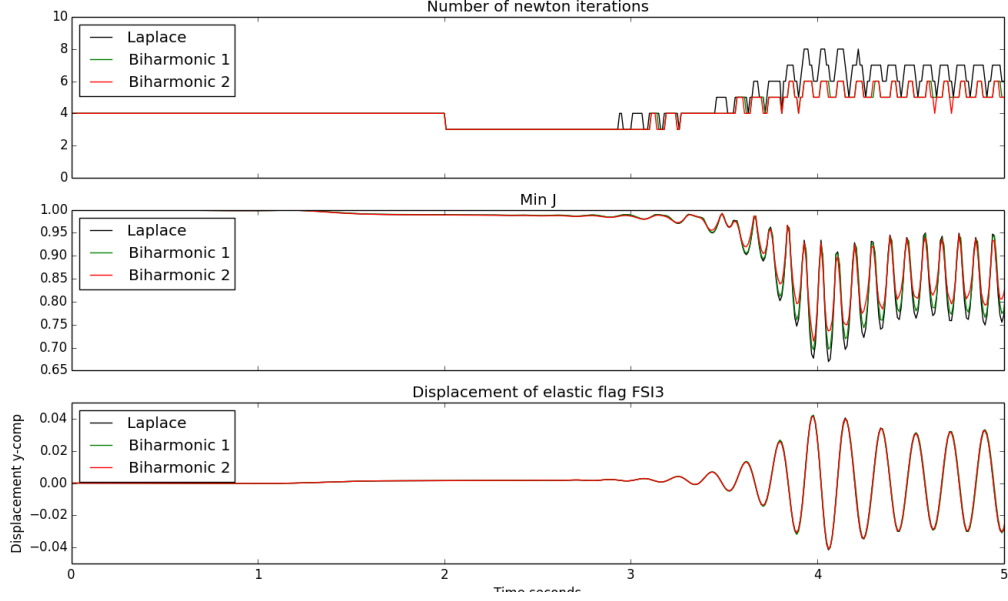


Figure 1.1: Investigation of mesh moving models for the FSI3 benchbark in the time interval $t \in [0, 5]$, comparing number of Newton iterations, mesh regularity, and vertical displacement of elastic flag.

Results

Figure 1.1 compares the mesh moving models in the time interval $t = [0, 5]$, when a stable periodic solution is obtained. All models shows a minima of mesh regularity at $3.8s < t < 4.2$, which is expected due to the largest deformation of the elastic flag. Both models shows a larger number of Newton-iterations are needed at each time step, when the elastic flag starts oscillating for $t > 3s$. The biharmonic models is superior in terms of number of iterations need per time step, and mesh regularity in comparison with the Laplace model. Further, the biharmonic 2 model shows better mesh regularity than biharmonic 1, but shows equal behavior in terms of Newton-iterations. For all models, no distinct difference in deformation of the y-component is found.

Discussion

The numerical results confirms biharmonic models produce a better regularity of the fluid mesh cells, which in turn reduces the number of Newton-iterations needed per time step. However, better evolution of mesh cells is by no means necessary for solving the FSI-3 problem. Therefore, the Laplace model remains a good choice, and its simplicity is preferable in terms of computational time (a topic to be discussed in section 1.2).

1.2 Investigation of temporal stability

One of the main challenges for constructing time-stepping schemes for ALE-methods, is the additional non-linearity introduced by the domain-velocity term in the fluid problem [10],

$$\hat{\mathbf{J}}_f(\hat{F}_f^{-1}(\hat{\mathbf{v}}_f - \frac{\partial \hat{\mathbf{T}}_f}{\partial t}) \cdot \hat{\nabla})\hat{\mathbf{v}}_f \quad (1.1)$$

Closer inspection of the convection term **reviles** spatial and temporal differential operators depending non-linearly on one another. These differential operators often appear separated, making discretization of a general time-stepping scheme not directly intuitive. The **domain-velocity** $\frac{\partial \hat{\mathbf{T}}_f}{\partial t}$ have proven to effect the stability of first and second-order time stepping schemes on fixed grids, but to what extent remains unclear [10, 9]. The second order Crank-Nicolson used in this thesis, have also shown to suffer from temporal stability for long-term simulations of fluid problems, on fixed-grids [47]. The unconditionally stable Crank-Nicolson scheme is restricted by the condition [47],

$$k \leq ch^{\frac{2}{3}} \quad (1.2)$$

Where c is a costant, while k and h is the time-step and a mesh-size parameter

while for the stability of the time derivative of the ALE-mapping, no accurate restriction is obtained (but thorough explored in [10]). As a result, time step restriction is necessary to ensure that numerical stability [10].

The temporal stability for the implicit Crank-Nicolson scheme, for the validation benchmark chosen in this thesis, was studied in [30]. The criteria for the numerical experiments was to obtain a stable solution in the time interval of 10 seconds, by temporal and spatial refinement studies. Following the ideas of [30], a second order scheme based on the Crank-Nicolson yields two **possibilities**.

Discretization 1.1. *Crank-Nicolson secant method*

$$\left[\frac{\hat{\mathbf{J}}(\hat{\mathbf{u}}^n)\hat{\nabla}\hat{\mathbf{v}}^n\hat{\mathbf{F}}_W^{-1}}{2} + \frac{\hat{\mathbf{J}}(\hat{\mathbf{u}}^{n-1})\hat{\nabla}\hat{\mathbf{v}}^{n-1}\hat{\mathbf{F}}_W^{-1}}{2} \right] \frac{\hat{\mathbf{u}}^n - \hat{\mathbf{u}}^{n-1}}{k}$$

Discretization 1.2. *Crank-Nicolson midpoint-tangent method*

$$\left[\frac{\hat{\mathbf{J}}(\hat{\mathbf{u}}_{cn})\hat{\nabla}\hat{\mathbf{v}}_{cn}\hat{\mathbf{F}}_W^{-1}}{2} \right] \frac{\hat{\mathbf{u}}^n - \hat{\mathbf{u}}^{n-1}}{k} \quad \hat{\mathbf{u}}_{cn} = \frac{\hat{\mathbf{u}}^n + \hat{\mathbf{u}}^{n-1}}{2} \quad \hat{\mathbf{v}}_{cn} = \frac{\hat{\mathbf{v}}^n + \hat{\mathbf{v}}^{n-1}}{2}$$

The numerical experiments showed very similar performance for Discretization 1.1 and 1.2, and significant differences of temporal accuracy was not found [30]. However, spatial and temporal refinement showed the implicit Crank-Nicolson scheme gave stability problems for certain time-steps k . Choosing $k = [0.005, 0.003]$, the FSI-3 problem (Section ??) suffered from numerical instabilities. Interestingly, the instabilities occurred earlier in simulation time for increasing mesh refinement. A similar experiment in [49], showed reducing the time step $k = 0.001$ yield stable long-time simulation for both Discretization 1.1 and 1.2

To cope with the numerical unstabilities two approaches have been suggested in the literature, the *shifted Crank-Nicolson* and the *frac-step method* [30, 49, 47]. In this thesis the shifted Crank-Nicolson scheme was considered, introducing stability to the overall system by shifting the θ parameter slightly to the implicit side. If the shift is dependent of the time-step k such that $\frac{1}{2} \leq \theta \leq \frac{1}{2} + k$, the scheme will be of second order [30].

1.2.1 Results

A numerical investigation of temporal stability is shown in Figure 1.2, 1.3, where the shifted Crank-Nicolson scheme $\theta = 0.5 + \Delta t$, is compared the original Crank-Nicolson $\theta = 0.5$. The shifted version clearly show stability properties surpassing the original Crank-Nicolson scheme, for all numerical experiments. Choosing $\Delta t = 0.01$, the shifted Crank-Nicolson scheme retain long-time temporal stability, while capturing the physics of the benchmark. While for the ordinary Crank-Nicolson scheme, numerical experiments showed choosing $\Delta t = 0.001$ was necessary to ensure stability, confirming the results found in [49]. Thus, reducing the time steps needed by a factor of ten.

Figure 1.2 shows choosing $\Delta t \in [0.2, 0.1]$ results in a steady-state solution. I believe this observation can be explained by influence the solid problem (Section). A centered Crank-Nicolson scheme $\theta = \frac{1}{2}$ is energy conservative, meaning little or no energy is dissipated. While a backward-Euler scheme $\theta = 1$ has little or no conservation of energy, meaning energy is easily dissipated from the system. The shifted Crank-Nicolson scheme dissipate more energy from the structure, if the choice of time step is sufficiently high, such as $\Delta t \in [0.2, 0.1] \rightarrow \theta = [0.7, 0.6]$. Therefore, no periodic oscillation of the elastic flag is obtained. The validation of the solid solver shows this property, for the sub-problems CSM-1 and CSM-3. Given the same solid parameters, a steady-state solution is obtained for CSM-1 ($\theta = 1.0$), while CSM-3 yields a periodic solution CSM-3 ($\theta = 0.5$), shown in Figure 1.4 .

For $\Delta t \in [0.05, 0.02]$, the numerical scheme is close to centered, and conservation of energy is nearly preserved. However, spatial refinement by keeping a constant mesh resolution, initiates breakdown of the Newton-solver at an earlier time step. The breakdown is not due to mesh entanglement of the ALE-mapping, but divergence of Newton method [30]. It is assumed that the divergence is linked to the influence of

the domain velocity, by the research found in [10], but no clear time step restriction is obvious. Indeed, several works indicates choosing time step for a shifted Crank-Nicolson scheme is based on trial and error [49, 47].

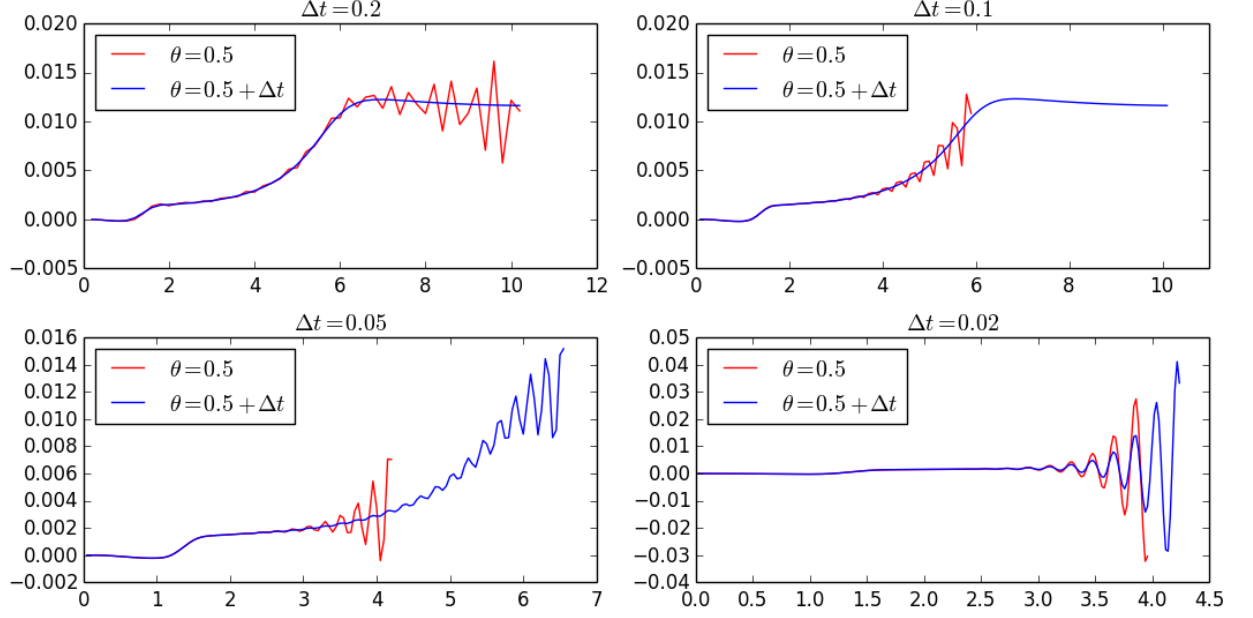


Figure 1.2: Investigation of temporal stability for the FSI3 benchbark in the time interval $t \in [0, 10]$, comparing the shifted and centered Crank-Nicolson scheme.

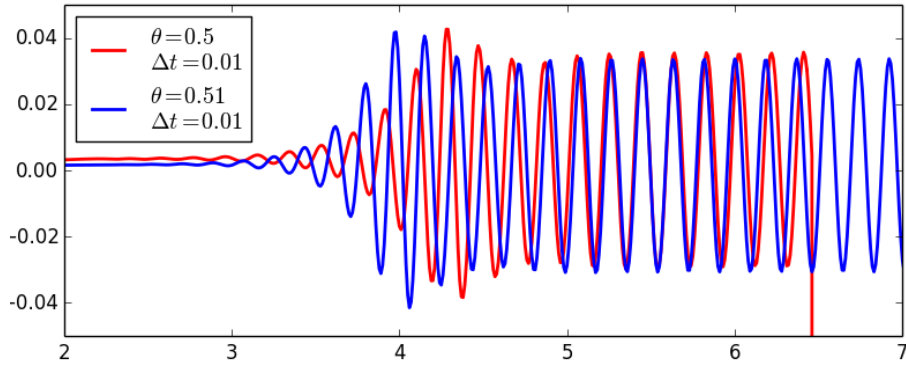


Figure 1.3: Investigation of temporal stability for the FSI3 benchbark in the time interval $t \in [0, 10]$, comparing the shifted and centered Crank-Nicolson scheme.

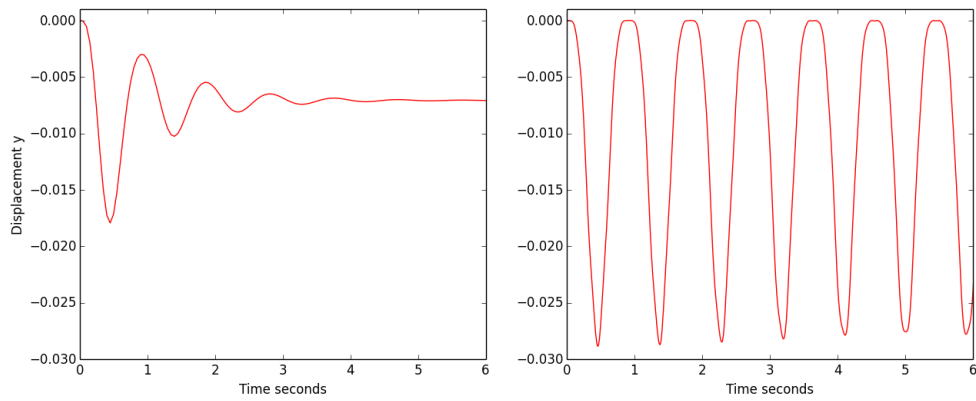


Figure 1.4: A comparison of a centered Crank-Nicolson scheme and backward Euler scheme

1.3 Optimization of the Newtonsolver

A *bottleneck* **express** a phenomena where the total performance of a complete implementation is limited to small code fragments, accounting for the primary consumption of computer resources.

As for many other applications, within **computational science** one can often assume the **consummation** of resources follows the *The Pareto principle*. Meaning that for different types of events, roughly 80% of the effects come from 20% of the causes. An analogy to computational sciences it that 80% of the computational demanding operations comes from 20% of the code. In our case, the bottleneck is the **newtonsolver**. The two main reasons for this **is**

- **Jacobian assembly**

The construction of the Jacobian matrix for the total **residue** of the **system**, is the most time demanding **operations** within the whole computation.

- **Solver.**

As **iterative solvers are limited** for the solving of fluid-structure interaction problems, direct solvers was implemented for this thesis. As such, the operation of solving a linear problem at each iteration is computational demanding, leading to less computational efficient operations. **Mention order of iterations?**

Facing these problems, several attempts **was** made to **speed-up the implementation**. The FEniCS project consist of several nonlinear solver backends, **were** fully user-customization option are available. However one main problem which **we** met was the fact that **FEniCS assembles** the matrix of the different variables over the whole mesh, even though the variable is only defined in **one to the sub-domains** of the **system**. In **our** case the pressure is only defined within the fluid domain, and therefore the matrix for the total residual consisted of several zero columns within the structure region. **FEniCS provides a solution for such problems**, but therefore **we were forced** to construct our own solver and not make use of the built-in nonlinear solvers.

Newtons method can be written as,

$$\mathbf{F}(\mathbf{x}_n) + \nabla \mathbf{F}(\mathbf{x}_n)(\mathbf{x}_n - \mathbf{x}_0) = 0$$

where \mathbf{F} is the **residue**, \mathbf{x}_n , \mathbf{x}_0 is vector, and $\nabla \mathbf{F}$ is the Jacobian of the **residue**

Assembly of the Jacobian at each iteration is CPU-intensive, hence the main speed-up effort were **centered** around Jacobian assembly. Of the **speed-ups methods** explored in this thesis, some some are *consistent* methods while others are *nonconsistent*. Consistent methods are methods that always will **work**, involving smarter assembly methods of the linear system **??**. The **non-consistent** method presented involves simplifications of the linear system **??**, which **in turn** affects the convergence of **Newtons** method. Hence, by **cheaper** Jacobi assembly, additional **Newton iterations** are often necessary for convergence. Therefore one also **risk** breakdown of the solver as the **Newton iterations may not converge**.

1.3.1 Consistent methods

Jacobi buffering

By inspection of the Jacobi matrix, linear terms remain constant within each time step. By assembling linear terms only in the first Newton iteration, reduces the assembly time at each time step. Jacobi buffering is not a simplification of the original problem, meaning convergence of Newton method is unaffected.

1.3.2 Non-consistent methods

Reuse of Jacobian

If one approach of reusing the Jacobian for the linear system was proposed. In other words, the LU-factorization of the system is reused until the Jacobi is re-assembled. This method greatly reduced the computational time for each time step.

Quadrature reduce

The assemble time of the Jacobian greatly depends on the degree of polynomials used in the discretisation of the total residual. Within FEniCS the order of polynomials representing the Jacobian can be adjusted. The use of lower order polynomials reduces assemble time of the Jacobian at each newton-iteration, however it leads to an inexact Jacobian which may results to additional iterations.

1.3.3 Comparison of speedup methods

In Figure 1.5, all speed-up techniques are compared on the time interval $t = [0, 5]$.

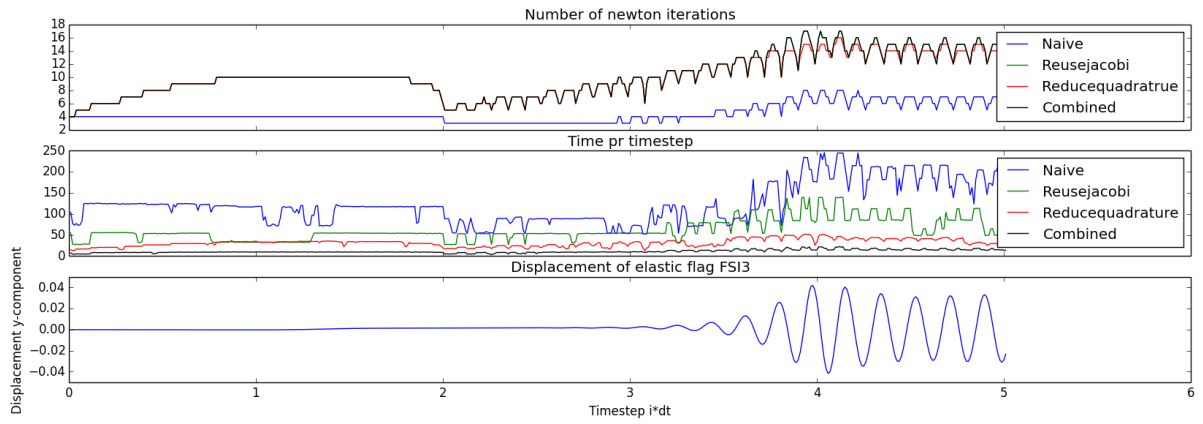


Figure 1.5: Comparison of speed-up techniques for the laplace mesh model

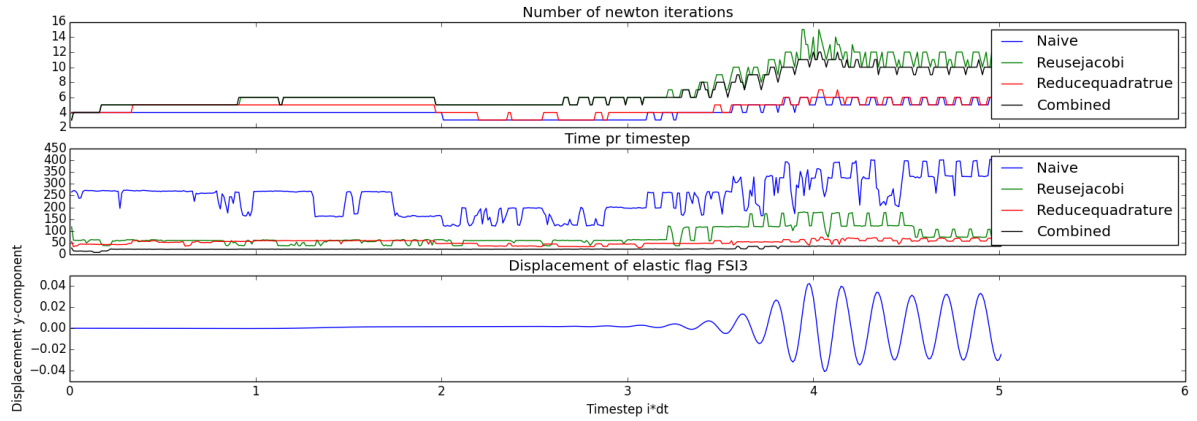


Figure 1.6: Comparison of speed-up techniques for the biharmonic type 1 mesh model

Table 1.1: Comparison of speedup techniques

Laplace					
Implementation	Naive	Buffering	Reducequad.	Reusejacobi	Combined
Mean time/-timestep	123.1		31.4	61.3	11.1
Speedup %	1.0		74.46%	50.19%	90.97 %
Mean iteration	4.49		10.1	10.2	10.2
Biharmonic Type 1					
Implementation	Naive	Buffering	Reducequad.	Reusejacobi	Combined
Mean time/-timestep	243.3	307.6	51.6	76.7	24.8
Speedup %	1.0	-26%	78.7%	68.4 %	89.7%
Mean iteration	4.1	6.2	4.6	7.1	6.8
Biharmonic Type 2					
Implementation	Naive	Buffering	Reducequad.	Reusejacobi	Combined
Mean time/-timestep			60.5	95.3	20.7
Speedup %	1.0	%	%	%	%
Mean iteration	4.1		6.29	6.9	6.9

Bibliography

- [1] Martin Alnæs, Anders Logg, Kristian Ølgaard, Marie Rognes, and Garth Wells. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(212):4207–4211, 2015.
- [2] Robert T Biedron and Elizabeth M Lee-Rausch. Rotor Airloads Prediction Using Unstructured Meshes and Loose CFD/CSD Coupling.
- [3] P I Crumpton. Implicit time accurate solutions on unstructured dynamic grids. (95):1–23, 1995.
- [4] J Donea, A Huerta, J-Ph Ponthot, and A Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. (1969):1–38, 2004.
- [5] Richard P Dwight. Robust Mesh Deformation using the Linear Elasticity Equations.
- [6] Stéphane Étienne, D Tremblay, and Dominique Pelletier. Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems. *36th AIAA Fluid Dynamics Conference and Exhibit*, (June):1–11, 2006.
- [7] Miguel A Fernández and Jean-Frédéric Gerbeau. Algorithms for fluid-structure interaction problems. 2009.
- [8] Miguel A. Fernández, Jean Frederic Gerbeau, and Ceremade Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.
- [9] L Formaggia and F Nobile. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-west journal of numerical mathematics*, Vol. 7(No. 2):105–131, 1991.
- [10] Luca Formaggia and Fabio Nobile. Stability analysis of second-order time accurate schemes for ALE-FEM. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41 SPEC. ISS.):4097–4116, 2004.
- [11] Christiane Förster, Wolfgang A. Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007.

- [12] Bernhard Gatzhammer. Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions. page 261, 2014.
- [13] Philippe Geuzaine. Numerical Simulations of Fluid-Structure Interaction Problems using MpCCI. (1):1–5.
- [14] Brian T. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 2003.
- [15] Gerhard A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. 2000.
- [16] Jaroslav Hron and Stefan Turek. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. *Fluid-Structure Interaction*, 53:371–385, 2006.
- [17] Ming-Chen Hsu and Yuri Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50(6):821–833, dec 2012.
- [18] Su-Yuen Hsu, Chau-Lyan Chang, and Jamshid Samareh. A Simplified Mesh Deformation Method Using Commercial Structural Analysis Software.
- [19] Jay D. Humphrey. *Cardiovascular Solid Mechanics*. Springer New York, New York, NY, 2002.
- [20] Hrvoje Jasak and Željko Tuković. Automatic mesh motion for the unstructured Finite Volume Method. *Transactions of Famena*, 30(2):1–20, 2006.
- [21] Jae-Hyun Kim and Hyung-Cheol Shin. Application of the ALE technique for underwater explosion analysis of a submarine liquefied oxygen tank. *Ocean Engineering*, 35(8-9):812–822, jun 2008.
- [22] V V Meleshko. Bending of an Elastic Rectangular Clamped Plate: Exact Versus 'Engineering' Solutions. *Journal of Elasticity*, 48(1):1–50, 1997.
- [23] José Merodio and Giuseppe Saccomandi. Continuum Mechanics - Volume I. In *Volume 1*, chapter 3, pages 82–84. EOLSS, 2011.
- [24] Selim MM and Koomullil RP. Mesh Deformation Approaches – A Survey. *Journal of Physical Mathematics*, 7(2), 2016.
- [25] J Newman. *Marine Hydrodynamics*. 1977.
- [26] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, Cambridge, 2010.
- [27] M Razzaq, Stefan Turek, Jaroslav Hron, J F Acker, F Weichert, I Grunwald, C Roth, M Wagner, and B Romeike. Numerical simulation and benchmarking of fluid-structure interaction with application to Hemodynamics. *Fundamental Trends in Fluid-Structure Interaction*, 1:171–199, 2010.

BIBLIOGRAPHY

- [28] T. Richter and T. Wick. Finite elements for fluid-structure interaction in ALE and fully Eulerian coordinates. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2633–2642, 2010.
- [29] Thomas Richter. Fluid Structure Interactions. 2016.
- [30] Thomas Richter and Thomas Wick. On Time Discretizations of Fluid-Structure Interactions. pages 377–400. 2015.
- [31] Patrick J. Roache. Code Verification by the Method of Manufactured Solutions. *Journal of Fluids Engineering*, 124(1):4, 2002.
- [32] P.J. Roache. *Verification and Validation in Computational Science and Engineering*. Computing in Science Engineering, Hermosa Publishers, 1998, 8-9, 1998.
- [33] Edward J. Rykiel. Testing ecological models: The meaning of validation. *Ecological Modelling*, 90(3):229–244, 1996.
- [34] Kambiz Salari and Patrick Knupp. Code Verification by the Method of Manufactured Solution. Technical report, Sandia National Laboratories, 2000.
- [35] LE Schwer. Guide for verification and validation in computational solid mechanics. *American Society of Mechanical Engineers*, PTC 60(V&V 10):1–15, 2006.
- [36] Jason P Sheldon, Scott T Miller, and Jonathan S Pitt. Methodology for Comparing Coupling Algorithms for Fluid-Structure Interaction Problems. *World Journal of Mechanics*, 4(February):54–70, 2014.
- [37] J.C. Simo and F. Armero. Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier-Stokes and Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 111(1-2):111–154, jan 1994.
- [38] Ian Sommerville. Verification and Validation. Technical Report February, 2006.
- [39] K Stein, T Tezduyar, and R Benney. Mesh Moving Techniques for Fluid-Structure Interactions With Large Displacements.
- [40] Keith Stein, Richard Benney, Tayfun Tezduyar, and Jean Potvin. Fluid–structure interactions of a cross parachute: numerical simulation. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):673–687, dec 2001.
- [41] Ryo Torii, Marie Oshima, Toshio Kobayashi, Kiyoshi Takagi, and Tayfun E. Tezduyar. Fluid–structure interaction modeling of a patient-specific cerebral aneurysm: influence of structural modeling. *Computational Mechanics*, 43(1):151–159, dec 2008.

- [42] Stefan Turek, Jaroslav Hron, Mudassar Razzaq, and Hilmar Wobker. Numerical Benchmarking of Fluid-Structure Interaction : A comparison of different discretization and solution approaches.
- [43] A V, T Passerini, A Quaini, U Villa, A Veneziani, and S Canic. Numerical Analysis and Scientific Computing Preprint Series Validation of an open source framework for the simulation of blood flow in rigid and deformable vessels Preprint # 17 Department of Mathematics University of Houston. 2013.
- [44] Jan Vierendeels. Comparison of the Hemodynamic and Thrombogenic Performance of Two Bileaflet Mechanical Heart Valves Using a CFD/FSI Model. *Journal of Biomechanical Engineering*, 129(4):558, jan 2007.
- [45] Wolfgang A. Wall, Axel , Gerstenberger, Peter , Gamnitzer, Christiane , Förster, and Ekkehard , Ramm. Large Deformation Fluid-Structure Interaction – Advances in ALE Methods and New Fixed Grid Approaches. In *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, pages 195—232. Springer Berlin Heidelberg, 2006.
- [46] Frank White. *Viscous fluid flow*. McGraw-Hill, third edit edition.
- [47] T. Wick. Stability Estimates and Numerical Comparison of Second Order Time-Stepping Schemes for Fluid-Structure Interactions. In *Numerical Mathematics and Advanced Applications 2011*, pages 625–632. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [48] T Wick and Thomas Wick. Variational-monolithic ALE fluid-structure interaction: Comparison of computational cost and mesh regularity using different mesh motion techniques.
- [49] Thomas Wick. *Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve*. PhD thesis, Heidelberg.
- [50] Thomas Wick. Solving Monolithic Fluid-Structure Interaction Problems in Arbitrary Lagrangian Eulerian Coordinates with the deal.II Library.
- [51] Thomas Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89(13-14):1456–1467, 2011.
- [52] Thomas Wick. Fully Eulerian fluid-structure interaction for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 255:14–26, 2013.
- [53] Klaus Wolf, Schloss Birlinghoven, Code Coupling Interface, Open Programming Interface, and Distributed Simulation. Mpcci – the General Code Coupling Interface. 6. *LS-DYNA Anwenderforum, Frankenthal 2007 IT*, pages 1–8, 2007.
- [54] P. Wriggers. *Computational contact mechanics, second ed.*, Springer. 2006.

BIBLIOGRAPHY

- [55] Hou Zhang, Xiaoli Zhang, Shanhong Ji, Yanhu Guo, Gustavo Ledezma, Nagi Elabbasi, and Hugues DeCougny. Recent development of fluid-structure interaction capabilities in the ADINA system. *Computers and Structures*, 81(8-11):1071–1085, 2003.