Multi-level quasi-Newton coupling algorithms for the partitioned simulation of fluid-structure interaction

Joris Degroote[a], Jan Vierendeels[a]

[a] *Department of Flow, Heat and Combustion Mechanics, Ghent University,*
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium

Abstract

In partitioned fluid-structure interaction simulations, the flow equations and the structural equations are solved separately. Coupling iterations between the flow calculation and the structural calculation can be used to enforce the equilibrium conditions on the fluid-structure interface. Low wave number Fourier modes of the difference between the correct interface displacement and the interface displacement during Gauss-Seidel coupling iterations are typically unstable for strongly coupled problems with a Dirichlet-Neumann decomposition. Using interface quasi-Newton iterations with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS), these unstable modes are automatically detected and stabilized. As the unstable modes have a low wave number and can hence be resolved on coarser grids, the new multi-level IQN-ILS (ML-IQN-ILS) technique first constructs the approximation for the inverse of the Jacobian on coarser grid levels and then uses it and improves it further on the original, finest grid level. This multi-level approach reduces the number of coupling iterations on the finest grid level and can also be applied to the interface block quasi-Newton (IBQN-LS) technique. One-dimensional and three-dimensional numerical results demonstrate that this new class of multi-level quasi-Newton coupling techniques can reduce the duration of a partitioned fluid-structure interaction simulation.

Keywords: fluid-structure interaction, partitioned, coupling algorithm, quasi-Newton, tube

---

Email addresses: `Joris.Degroote@UGent.be` (Joris Degroote), `Jan.Vierendeels@UGent.be` (Jan Vierendeels)

URL: `http://www.fsi.ugent.be/` (Joris Degroote)

Preprint submitted to Computer Methods in Applied Mechanics and Engineering October 27, 2017

1. Introduction

The mutual interaction between a fluid flow and a deforming structure is referred to as fluid-structure interaction (FSI). Life-saving examples are the opening of a parachute [1] or an air bag [2]. Undesired occurrences of fluid-structure interaction are collapses of bridges [3] and cooling towers due to wind and also flutter of aircraft wings [4] and turbine blades [5]. In the biomedical field, the interaction between an elastic artery [6, 7] or a heart chamber [8] and the blood that flows through them is of interest. Also for the design of artificial heart valves [9, 10], fluid-structure interaction needs to be taken into account.

For the simulation of fluid-structure interaction, there are two approaches. Monolithic simulation techniques solve the governing equations of the fluid flow and the structure simultaneously [11–14]. Conversely, partitioned techniques solve the flow equations and the structural equations separately. An advantage of the partitioned approach to simulate this coupled problem is that the flow equations and the structural equations can be solved with a different solution technique. Moreover, the partitioned approach allows to couple existing flow solvers and structural solvers.

In this article, the focus lies on partitioned simulation techniques which couple the flow solver and the structural solver as 'black boxes', which means that the discretization and solution techniques of the solvers do not have to be known. These techniques can be categorized as explicit or implicit. Several explicit (also known as loosely or weakly coupled) partitioned techniques exist [4, 15–18], which are suitable for aeroelastic simulations [19]. These techniques solve the flow equations and the structural equations separately and only once (or a fixed number of times) in each time step. Therefore, these techniques do not impose the equilibrium of the traction and velocity (or displacement) on the fluid-structure interface, which results in restrictions on the time step for stability reasons [19–21].

Implicit (or strongly coupled) partitioned techniques enforce the equilibrium of the traction and velocity (or displacement) on the fluid-structure interface in each time step. This can be achieved by, for example, Gauss-Seidel iterations or Newton-Raphson iterations. During Gauss-Seidel iterations in a time step, the flow equations and the structural equations are solved successively until some convergence tolerance is reached. After the solution of the flow equations, the boundary condition on the solid side of the interface is updated and vice versa. Once these coupling iterations within the time step have converged, the solution is the same as would have been found with a monolithic solver. Several strongly coupled partitioned techniques are able to couple 'black-box' solvers, for example

Gauss-Seidel iterations with Aitken relaxation [22–24], the Interface Generalized Minimal Residual method (Interface-GMRES) [25], the Interface Block Quasi-Newton technique with an approximation for the Jacobians from Least-Squares models (IBQN-LS) [26] and the Interface Quasi-Newton technique with an approximation for the Inverse of the Jacobian from a Least-Squares model (IQN-ILS) [27]. The convergence and stability properties of various of these methods are compared in [28].

Vector extrapolation [29, 30] can also be used. However, Küttler and Wall [31] conclude that Aitken relaxation is much simpler to implement and yields a faster simulation in many cases. Vector extrapolation methods perform a number of Gauss-Seidel iterations (with relaxation) before they extrapolate. By contrast, quasi-Newton methods like IQN-ILS perform only one Gauss-Seidel iteration at the beginning of the time step, followed by quasi-Newton steps. This is a major difference between the vector extrapolation methods and the quasi-Newton methods. Nevertheless, there is a similarity between the extrapolation itself and the rank-one update of the Jacobian's inverse by the quasi-Newton method. For a more detailed comparison, the reader is referred to [32].

Also the coupling iterations of the implicit partitioned techniques can suffer from stability issues. Consider, for example, Gauss-Seidel coupling iterations for a given fluid-structure interaction problem with a Dirichlet-Neumann decomposition. Once the coupling iterations have converged, the final solution of the time step takes into account the interaction between the fluid and the structure (up to the convergence tolerance of the coupling iterations). However, during the solution of the flow equations, all structural degrees of freedom are fixed and vice versa. Hence, this coupling technique treats the interaction implicitly in each time step but explicitly in each coupling iteration within a time step. The explicit treatment of the interaction during the coupling iterations can cause divergence of the Gauss-Seidel iterations with a Dirichlet-Neumann decomposition. Implicit treatment of the interaction during the coupling iterations can be achieved with (quasi-)Newton techniques.

Several stability analyses have been performed for the incompressible, inviscid flow in a straight, elastic tube. The parameters in these analyses are the length, radius and thickness of the tube, the density of the fluid and the structure, and the elastic properties of the structure. Causin et al. [20] determined the maximal relaxation factor that leads to convergence of Gauss-Seidel iterations for a Dirichlet-Neumann and a Neumann-Dirichlet decomposition of the fluid-structure interaction problem. Badia et al. [33] did the same for Robin-Dirichlet and Robin-Neumann decomposition. Degroote et al. performed a Fourier decomposition of the difference between the correct interface displacement and the interface displacement during a Gauss-Seidel coupling iteration with a Dirichlet-Neumann decomposition, first using an independent rings model without inertia in the structure [34] and later using an interacting segments model with structural inertia [35]. These Fourier analyses showed that typically only a fraction of all Fourier modes is unstable, notably the modes with a low wave number. Moreover, van Brummelen [28] come to the same conclusion using a semi-infinite open fluid domain bounded by a string or a beam, which demonstrates that this conclusion is not model dependent.

The above mentioned difference between the correct interface displacement and the interface displacement during a coupling iteration is further referred to as the error. Figure 1 depicts the amplification of the different wave numbers in this error during Gauss-Seidel coupling iterations for the calculation of the flow in a 1D flexible tube with a Dirichlet-Neumann decomposition (see Section 5.1 for the model description) [34]. In this figure, the time step is small and the structural stiffness is low. In every Gauss-Seidel coupling iteration, the amplitude of each Fourier mode in the error is multiplied with the amplification factor for the corresponding wave number. In this case, the modes with a wave number between 0 and $\pi/4$ are unstable because they have an amplification factor larger than one and so they will grow in each Gauss-Seidel iteration. One unstable mode is sufficient to cause divergence of the Gauss-Seidel iterations.

From these stability analyses and Figure 1, two lessons can be learnt, as will be explained below. While the standard, one-level IQN-ILS technique only takes advantage of the first one, the multi-level IQN-ILS (ML-IQN-ILS) technique, which is introduced in this article, takes advantage of both of them.

From Figure 1, it can be observed that only a fraction (approximately a quarter in this case) of the modes is unstable during Gauss-Seidel coupling iterations with a Dirichlet-Neumann decomposition. As mentioned above, Gauss-Seidel iterations with a Dirichlet-Neumann decomposition treat the interaction between the fluid and the structure explicitly during a coupling iteration because they solve the flow equations with all structural degrees of freedom fixed and vice versa. Thus only a fraction of the Fourier modes is unstable in the case of explicit treatment of the interaction during the coupling iterations. Only these unstable modes need to be treated implicitly during the coupling iterations. For the modes with an error amplification smaller than one in Figure 1, Gauss-Seidel iterations are an acceptable solutions strategy. From the Fourier analyses, it can thus be derived that no full-rank exact Jacobian is required to obtain fast convergence, which explains the performance of quasi-Newton methods like standard IQN-ILS and IBQN-LS.

The first lesson is that if a quasi-Newton technique like IQN-ILS is used for the implicit treatment of the interaction, then only a low-rank approximation for the exact Jacobian is required, as long as it represents the behaviour of the small fraction of unstable and slowly converging modes. For the modes which are not included in the Jacobian of this quasi-Newton technique, the inverse of this Jacobian corresponds with minus the identity matrix so the quasi-Newton iterations are equivalent to Gauss-Seidel iterations (see Eq. (30)) [32].

It can also be noticed in Figure 1 that the error modes with a low wave number have the highest amplification and are consequently most unstable. Based on this insight, the second lesson is that, due to their low wave number, the behaviour of the unstable modes can be determined on a relatively coarse grid. The new ML-IQN-ILS technique uses more than one grid level, each with a different number of grid points. It first calculates the coupled solution on the coarsest grid level and constructs the low-rank approximation for the inverse of the Jacobian as present in standard IQN-ILS while doing so. Subsequently, coupling iterations are performed on the second, finer grid level, during which the approximation for the inverse of the Jacobian obtained on the coarsest grid level is further improved by performing coupling iterations. This procedure is then repeated until the solution on the finest grid level has been found. This procedure will be explained in detail further in this work.

The goal of the multi-level IQN-ILS technique is thus to obtain the low-rank approximation for the inverse of the Jacobian required for the convergence of the coupling iterations on the finest grid level at a lower cost, by constructing it partly on coarser grid levels. As this reduces the number of coupling iterations on the finest grid level and as the cost of the coupling iterations on the coarser grid levels is low, the total time required to solve the coupled problem decreases. This new multi-level approach is depicted in Figure 2 for two grid levels. As only data on the fluid-structure interface is exchanged, this partitioned multi-level coupling technique can couple black-box solvers.

The name multi-grid is not used because it already refers to a common solution technique [36], which has been used for fluid-structure interaction simulations in for example [14, 37, 38] and which is different from the ML-IQN-ILS coupling technique presented here. A major difference is that in the standard multi-grid technique, the coarse grids provide a correction for the smooth error components on the fine grids. By contrast, the presented ML-IQN-ILS technique uses the coarse grids to generate an approximation for the inverse of the Jacobian on the fine grids. Accordingly, the ML-IQN-ILS technique begins each time step on the coarsest grid and ends on the finest grid. Hence, at variance with the standard multi-grid technique, the presented ML-IQN-ILS technique does not revisit the coarse grids during the solution process for a time step. While the different grid levels in multi-grid techniques are hierarchical, they are completely independent in the ML-IQN-ILS technique. As can be seen in Figure 2, the presented ML-IQN-ILS technique uses no communication between the different grid levels during the coupling iterations in a time step. Only data on the fluid-structure interface is exchanged between the different grid levels of the fluid and structure subdomains and the so-called 'coupling grid' which has a fine resolution.

Although not required, the calculation in the fluid and/or structure subdomains on each grid level could be performed with a multi-grid solver. In that case, however, the coarse grids used in the multi-grid solver would be independent of and totally unrelated to the coarse grids used on the different levels by the ML-IQN-ILS coupling technique presented in this article. Also no coupling iterations would be performed between the coarse grid problems of the multi-grid solvers. In fact, the solution techniques used in the flow solver and structural solver are not interfering with the ML-IQN-ILS coupling technique, which allows both solvers to be black-box solvers.

From the wide range of applications, the standard IQN-ILS coupling algorithm can be considered suitable for the partitioned solution of fluid-structure interaction problems in general, especially if they require implicit coupling and if the user wants to treat the flow solver and the structural solver as black boxes. Conversely, the presented multi-level algorithms are particularly developed for problems with fine grids where the phenomena driving the fluid-structure interaction can be described on a coarser grid. This will be demonstrated by the numerical examples.

The remainder of this article is organized as follows. Section 2 gives a brief overview of the governing equations and the notations. The interpolation between the different grid levels is described in Section 3, before the detailed explanation of ML-IQN-ILS in Section 4. As multi-level IBQN-LS (ML-IBQN-LS) is obtained from IBQN-LS in a similar way as ML-IQN-ILS from IQN-ILS, this derivation is provided in Appendix A. Numerical results in Section 5 for the flow in a one-dimensional (1D) tube and the propagation of a pressure wave in a three-dimensional (3D) tube illustrate the performance of ML-IQN-ILS and ML-IBQN-LS compared to the standard, one-level IQN-ILS and IBQN-LS. Finally, Section 6 offers the conclusions. Appendix B compares IQN-ILS with Interface-GMRES.

2. Governing equations

Figure 3 depicts an abstract fluid-structure interaction problem. The subdomains are indicated as Ω_f and Ω_s and their boundaries as Γ_f and Γ_s, with the subscripts f and s respectively denoting fluid and structure. The fluid-structure interface $\Gamma_{fs} = \Gamma_f \cap \Gamma_s$ is the common boundary of these subdomains. $\vec{e}_x$, $\vec{e}_y$ and $\vec{e}_z$ are the unit vectors in the horizontal, vertical and out-of-plane direction, respectively.

2.1. Flow equations

This article only considers incompressible fluids as they prove to be most challenging for the partitioned fluid-structure interaction techniques. However, it would be no problem to consider compressible fluids instead. The unsteady flow of an incompressible fluid is governed by the conservation of mass and the Navier-Stokes equations, given by

$$\nabla \cdot \vec{v} = 0 \tag{1a}$$

$$\rho_f \frac{\partial \vec{v}}{\partial t} + \rho_f \nabla \cdot (\vec{v}\vec{v}) - \nabla \cdot \bar{\sigma}_f = \vec{f}_f \tag{1b}$$

for $\vec{x} \in \Omega_f$. In these equations, ρ_f is the fluid density, $\vec{v}$ the fluid velocity and t the time. $\vec{f}_f$ represents the body forces per unit of volume on the fluid. For Newtonian fluids with dynamic viscosity μ_f, the stress tensor $\bar{\sigma}_f$ is defined as

$$\bar{\sigma}_f = -p\bar{I} + 2\mu_f \bar{\epsilon}_f \tag{2a}$$

with p the pressure and $\bar{I}$ the unit tensor. The rate of strain tensor $\bar{\epsilon}_f$ is given by

$$\bar{\epsilon}_f = \frac{1}{2}\left[\nabla\vec{v} + (\nabla\vec{v})^{\mathrm{T}}\right]. \tag{2b}$$

2.2. Structural equations

The deformation $\vec{u}$ of the structure is determined by the conservation of momentum

$$\rho_s \frac{\mathrm{d}^2\vec{u}}{\mathrm{d}t^2} - \nabla \cdot \bar{\sigma}_s = \vec{f}_s \tag{3}$$

for $\vec{x} \in \Omega_s$ with ρ_s the structural density and $\vec{f}_s$ the body forces per unit volume on the structure. The Cauchy stress tensor $\bar{\sigma}_s$ relates forces in the deformed configuration to areas in the deformed configuration, whereas the second Piola-Kirchhoff stress tensor $\bar{S}$ combines forces in the reference configuration with areas in the reference configuration. The relation between these tensors is given by

$$\bar{S} = J\bar{F}^{-1}\bar{\sigma}_s\bar{F}^{-\mathrm{T}} \tag{4}$$

with $\bar{F}$ the deformation gradient tensor and $J = \det(\bar{F})$. In large displacement calculations, the relation between the second Piola-Kirchhoff stress tensor $\bar{S}_s$ and the Green-Lagrange strain tensor $\bar{E}_s$ is imposed by the constitutive equation of the material. The Green-Lagrange strain tensor is given by

$$\bar{\epsilon}_s = \frac{1}{2}\left[\nabla\vec{u} + (\nabla\vec{u})^{\mathrm{T}} + (\nabla\vec{u})^{\mathrm{T}}\,\nabla\vec{u}\right] \tag{5}$$

for large displacements. All displacements are relative to the initial geometry.

2.3. Equilibrium conditions

The equilibrium conditions on the fluid-structure interface ($\vec{x} \in \Gamma_{fs}$) are the kinematic condition

$$\vec{v} = \frac{\mathrm{d}\vec{u}}{\mathrm{d}t} \tag{6}$$

and the dynamic condition

$$\bar{\sigma}_f \cdot \vec{n}_f = -\bar{\sigma}_s \cdot \vec{n}_s, \tag{7}$$

which stipulate that the velocity and the traction have to be the same on both sides of the interface. The vector $\vec{n}_f$ ($\vec{n}_s$) is the unit normal that points outwards from the subdomain Ω_f (Ω_s). Appropriate boundary conditions are imposed on $\Gamma_f \setminus \Gamma_{fs}$ and on $\Gamma_s \setminus \Gamma_{fs}$, depending on the problem at hand. A Dirichlet-Neumann decomposition of the fluid-structure interaction problem is applied, so the flow equations are solved with a given velocity (or displacement) of the fluid-structure interface and the structural equations are solved with a given traction on the interface.

2.4. Discrete equations

The flow equations and the structural equations are discretized in space and time with a method of choice, which results in a system of coupled discrete equations with the flow variables and the structural variables as unknowns. The discrete flow equations are represented by $\mathbb{F}$, the discrete structural equations by $\mathbb{S}$. The vector $\boldsymbol{v}$ groups all flow variables (velocity, pressure, etc.) in Ω_f; the vector $\boldsymbol{u}$ groups all structural variables (displacement, stress, etc.) in Ω_s. The displacement of the interface Γ_{fs} with respect to the initial geometry is represented by the vector $\boldsymbol{d}$ and the traction on the interface by the vector $\boldsymbol{s}$. In the case of a Dirichlet-Neumann decomposition, the displacement of the interface is considered as a function of the structural degrees of freedom ($\boldsymbol{d} = \boldsymbol{d}(\boldsymbol{u})$) and the traction on the interface as a function of the flow degrees of freedom ($\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{v})$). Hence, the coupled problem can be written as

$$\begin{cases} \mathbb{F}(\boldsymbol{v}, \boldsymbol{d}(\boldsymbol{u})) = \mathbf{0} \\ \mathbb{S}(\boldsymbol{u}, \boldsymbol{s}(\boldsymbol{v})) = \mathbf{0} \end{cases} \tag{8}$$

in which all variables are at the new time level t^{n+1}; the dependence of the solution on the variables at $t^n, t^{n-1}, \ldots$ is hidden.

The flow solver calculates the flow variables $\boldsymbol{v}$ that satisfy $\mathbb{F}(\boldsymbol{v}, \boldsymbol{d}(\boldsymbol{u})) = \mathbf{0}$ for a given interface displacement $\boldsymbol{d}$. From the flow field $\boldsymbol{v}$, the traction on the interface $\boldsymbol{s}$ is extracted. Therefore, the flow solver is represented by the function

$$\boldsymbol{s} = \boldsymbol{\mathcal{F}}(\boldsymbol{d}). \tag{9}$$

Similarly, the structural solver calculates the structural variables $\boldsymbol{u}$ that satisfy $\mathbb{S}(\boldsymbol{u}, \boldsymbol{s}(\boldsymbol{v})) = \mathbf{0}$ for a given traction on the interface $\boldsymbol{s}$. The displacement of the interface $\boldsymbol{d}$ is subsequently extracted from $\boldsymbol{u}$, so the structural solver is represented by

$$\boldsymbol{d} = \boldsymbol{\mathcal{S}}(\boldsymbol{s}). \tag{10}$$

The flow solver and the structural solver defined above are considered as black-box functions, which can be evaluated for a given value of their argument but whose Jacobian matrices are inaccessible. The input and output of these functions are limited to the fluid-structure interface.

3. Interpolation on the interface

As the multi-level coupling techniques use several grid levels for the flow equations and the structural equations, data has to be interpolated between different discretizations of the fluid-structure interface. However, even though the discretization of the interface inside the flow solver and the structural solver depends on the grid level, all operations of the coupling algorithm are performed on a unique grid, the so-called ‘coupling grid’ (see Figure 2). For the examples shown in this work, this coupling grid is identical to the interface discretization of the finest fluid grid.

Four different interpolations on the fluid-structure interface are required per coupling iteration. In each coupling iteration, the displacement is interpolated from the coupling grid to the interface of the current fluid grid, the flow equations are solved and the resulting distribution of the traction vector is interpolated from the interface of the fluid grid to the coupling grid. Subsequently, a distribution of the traction vector is interpolated from the coupling grid to the interface of the structural grid and the structural equations are solved, followed by interpolation of the displacement from the interface of the structural grid to the coupling grid.

In line with the definition of the flow solver and structural solver as black-box functions, the interpolation on the interface should not require access to the discretization in the solvers. Therefore, interpolation with radial basis functions is applied [39–41]. However, if a single interpolant would be created using all points on the interface, then a linear system with dimension proportional to the number of points on the interface would have to be solved to calculate the interpolation coefficients. The sparsity and condition number of this system would depend on the radius of the radial basis functions. To avoid the solution of this relatively large linear system, a local interpolant is constructed in the neighbourhood of each point on the interface. The coefficients of the local interpolant at a given point are calculated by solving a small linear system, based on the points in the immediate vicinity, as explained below. Many other interpolation techniques could be used instead but the interpolation is not the focus of this work.

The interpolation of a general variable z from grid a to grid b is described below. The variable z represents a component of the interface displacement $\vec{u}$ or traction $\bar{\sigma} \cdot \vec{n}$ in direction $\vec{e}_x$, $\vec{e}_y$ or $\vec{e}_z$. As values are interpolated between the coupling grid and the fluid or structural side of the interface, either grid a or grid b is equal to the coupling grid and the other one is the fluid-structure interface of a grid level in either the fluid or structure subdomain. For each point $\vec{x}_{b,k}$ ($k = 1, \ldots, m_b$) on grid b, three different steps are performed, namely first selection of the m points on grid a nearest to $\vec{x}_{b,k}$, then construction of the interpolant and finally the interpolation itself. Apart from the selection of the nearest points, the computing time of this interpolation procedure scales linearly with the number of points on the interface.

The selection of the m nearest points in the first step is based on the Euclidean distance between the point $\vec{x}_{b,k}$ and the points $\vec{x}_{a,j}$ ($j = 1, \ldots, m_a$) on grid a. Without countermeasures, incorrect points could be selected if the structure contains surfaces which are close to each other or sharp corners (see Figure 4). To avoid such problems, the fluid-structure interface has to be divided into a number of surfaces manually. Of course, the fluid grid, the structural grid and the coupling grid have to be divided in the same way so that there is a one-to-one relation between the surfaces on the different grids. Corners should only be present at the common boundary of two surfaces and not within a surface. For point $\vec{x}_{b,k}$ on a given surface of grid b, only the points $\vec{x}_{a,j}$ on the corresponding surface of grid a are considered when the nearest points are searched.

Through the values $z_{a,j}$ at the points $\vec{x}_{a,j}$ ($j = 1, \ldots, m$) around $\vec{x}_{b,k}$, a radial basis function interpolant $z(\vec{x})$ is constructed. This interpolant has the form

$$z(\vec{x}) = \sum_{j=1}^{m} \alpha_j \phi_r(||\vec{x} - \vec{x}_{a,j}||) + p(\vec{x}). \tag{11}$$

The scalars α_j are the coefficients of the scaled basis function $\phi_r(|| \cdot ||)$ which is radial with respect to the Euclidean distance

$$||\vec{x}|| = \sqrt{x_1^2 + \ldots + x_d^2}, \tag{12}$$

with d the dimension. In this work, a function introduced by Wendland [40] is used as basis function, namely

$$\phi(||\vec{x}||) = (1 - ||\vec{x}||)_+^4 (4||\vec{x}|| + 1). \tag{13}$$

The plus-sign behind the first term denotes that this term is zero if $1 - ||\vec{x}|| < 0$ such that ϕ has a compact support, which means that it is zero if $||\vec{x}|| > 1$. The scaled basis function $\phi_r(|| \cdot ||)$ with radius r is then defined as

$$\phi_r(||\vec{x}||) = \phi(||\vec{x}||/r). \tag{14}$$

The radius is chosen as the maximal distance between the point $\vec{x}_{b,k}$ and any of the points $\vec{x}_{a,j}$

$$r = \max_{j=1 \ldots m} ||\vec{x}_{a,j} - \vec{x}_{b,k}||. \tag{15}$$

The term p in Eq. (11) is a low-degree polynomial. Combined with the radial basis function in Eq. (13), a linear polynomial

$$p(\vec{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d \tag{16}$$

yields a unique interpolant [40]. Moreover, linear polynomials are captured exactly by the interpolant, so rigid body translations are exactly recovered when z represents a component of the displacement.

The coefficients α_j ($j = 1, \dots, m$) and $\beta_{j'}$ ($j' = 0, \dots, d$) are determined by the interpolation conditions

$$z(\vec{x}_{a,j}) = z_{a,j} \tag{17}$$

for $j = 1, \dots, m$ and the additional conditions

$$\sum_{j=1}^{m} \alpha_j q(\vec{x}_{a,j}) = 0 \tag{18}$$

for all polynomials q with a degree lower than the degree of p. All these conditions can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{z}_a \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{P} \\ \boldsymbol{P}^{\mathrm{T}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}, \tag{19}$$

with the vector $\boldsymbol{z}_a \in \mathbb{R}^{m \times 1}$ containing the $z_{a,j}$, $\boldsymbol{\alpha} \in \mathbb{R}^{m \times 1}$ the α_j and $\boldsymbol{\beta} \in \mathbb{R}^{(d+1) \times 1}$ the $\beta_{j'}$. The symmetric matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times m}$ has as elements the evaluations of the basis function

$$\Phi_{k,j} = \phi_r(||\vec{x}_{a,k} - \vec{x}_{a,j}||) \tag{20}$$

and the j^{th} row of matrix $\boldsymbol{P} \in \mathbb{R}^{m \times (d+1)}$ is given by

$$\begin{bmatrix} 1 & x_{a,j,1} & \dots & x_{a,j,d} \end{bmatrix}. \tag{21}$$

After the calculation of the coefficient vector from Eq. (19), the value of z at $\vec{x}_{b,i}$ is calculated as

$$z(\vec{x}_{b,k}) = \begin{bmatrix} \boldsymbol{\Phi}' & \boldsymbol{P}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}' & \boldsymbol{P}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{P} \\ \boldsymbol{P}^{\mathrm{T}} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{z}_a \\ \boldsymbol{0} \end{bmatrix}, \tag{22}$$

with the row matrix $\boldsymbol{\Phi}' \in \mathbb{R}^{1 \times m}$ containing the evaluations of the basis function

$$\Phi'_{1,j} = \phi_r(||\vec{x}_{b,k} - \vec{x}_{a,j}||) \tag{23}$$

and the row matrix $\boldsymbol{P}' \in \mathbb{R}^{1 \times (d+1)}$ given by

$$\boldsymbol{P}' = \begin{bmatrix} 1 & x_{b,k,1} & \dots & x_{b,k,d} \end{bmatrix}. \tag{24}$$

The row matrix in Eq. (22) is truncated to its first m columns

$$\boldsymbol{H} = \left[\begin{bmatrix} \boldsymbol{\Phi}' & \boldsymbol{P}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{P} \\ \boldsymbol{P}^{\mathrm{T}} & \boldsymbol{0} \end{bmatrix}^{-1} \right]_{1 \times m} \tag{25}$$

as the last $d + 1$ columns are always multiplied by zero. Hence, the interpolation can be written as

$$z(\vec{x}_{b,k}) = \boldsymbol{H}\boldsymbol{z}_a. \tag{26}$$

As $\boldsymbol{H}$ does not depend on z, each component of the displacement or traction at a given point of grid b is interpolated with the same $\boldsymbol{H}$. In this article, the matrices $\boldsymbol{H}$ are calculated once using the initial position of the grids, but it is of course possible to recalculate these matrices in each coupling iteration.

4. ML-IQN-ILS

In the explanation of this coupling algorithm, a prime denotes the Jacobian matrix of a function and a hat refers to an approximation. The output of a solver is indicated with a tilde as it is often not the same as the input of the other solver. For example, the displacement calculated by the structural solver is indicated with a tilde because it is different from the displacement used by the flow solver in the same coupling iteration. The grid level is indicated with a subscript i and the coupling iteration within time step $n + 1$ with a superscript k. The superscript $n + 1$ is omitted wherever possible. The standard algorithm with a single grid level is described first, followed by the multi-level algorithm with g grid levels. The first grid level is the coarsest grid level and the g^{th} grid level is the finest one.

The FSI problem reformulated as a set of nonlinear equations in the interface's displacement

$$\boldsymbol{\mathcal{R}}(\boldsymbol{d}) = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}) - \boldsymbol{d} = \boldsymbol{0} \tag{27}$$

can be solved by means of Newton-Raphson iterations

$$\text{solve} \quad \boldsymbol{\mathcal{R}}'^{k} \Delta \boldsymbol{d}^k = -\boldsymbol{r}^k \tag{28a}$$

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \Delta \boldsymbol{d}^k \tag{28b}$$

with the residual calculated as

$$\boldsymbol{r}^k = \boldsymbol{\mathcal{R}}(\boldsymbol{d}^k) = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^k) - \boldsymbol{d}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k. \tag{29}$$

$\boldsymbol{\mathcal{R}}'^{k}$ denotes the Jacobian of $\boldsymbol{\mathcal{R}}$, evaluated at $\boldsymbol{d}^k$. The Newton-Raphson iterations in the time step have converged when $||\boldsymbol{r}^k||_2 \leq \varepsilon$ with ε the convergence tolerance. However, the exact Jacobian of $\boldsymbol{\mathcal{R}}$ is unknown as the Jacobians of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}$ are unavailable. Moreover, the linear system in Eq. (28a) with as dimension the number of degrees of freedom in the displacement of the fluid-structure interface has to be solved in each Newton-Raphson iteration. If the Jacobian $\boldsymbol{\mathcal{R}}'$ is approximated and quasi-Newton iterations are performed, black-box solvers can be used. However, the linear system in Eq. (28a) still needs to be solved. As will be explained below, it is more advantageous to approximate the *inverse* of the Jacobian by applying the least-squares technique introduced by Vierendeels et al. [26] on a particular set of vectors, which is done by the standard IQN-ILS algorithm. The quasi-Newton iterations with the approximation for the inverse of the Jacobian can be written as

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \widehat{\left(\boldsymbol{\mathcal{R}}'^k\right)^{-1}} \left(-\boldsymbol{r}^k\right). \tag{30}$$

The quasi-Newton iterations start from the initial guess ($k = 0$)

$$\boldsymbol{d}^k = \frac{5}{2}\boldsymbol{d}^n - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2} \tag{31}$$

which is an extrapolation based on the previous time steps. Lower order extrapolations are used for the first two time steps.

It can be seen from Eq. (30) that the approximation for the inverse of the Jacobian does not have to be created explicitly; a procedure to calculate the product of this matrix with the vector $-\boldsymbol{r}^k$ is sufficient. The vector $-\boldsymbol{r}^k$ is the difference between the desired residual, i.e. $\boldsymbol{0}$, and the current residual $\boldsymbol{r}^k$ and it is further denoted as $\Delta\boldsymbol{r} = \boldsymbol{0} - \boldsymbol{r}^k = -\boldsymbol{r}^k$. The matrix-vector product in Eq. (30) is calculated from information obtained during the previous quasi-Newton iterations. Eq. (29) shows that the flow equations and structural equations are solved in quasi-Newton iteration k, resulting in $\tilde{\boldsymbol{d}}^k = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^k)$ and the corresponding residual $\boldsymbol{r}^k$. So, at the beginning of quasi-Newton iteration $k+1$, a set of known residual vectors

$$\boldsymbol{r}^k, \quad \boldsymbol{r}^{k-1}, \quad \ldots, \quad \boldsymbol{r}^1, \quad \boldsymbol{r}^0 \tag{32a}$$

and the corresponding set of vectors $\tilde{\boldsymbol{d}}$

$$\tilde{\boldsymbol{d}}^k, \quad \tilde{\boldsymbol{d}}^{k-1}, \quad \ldots, \quad \tilde{\boldsymbol{d}}^1, \quad \tilde{\boldsymbol{d}}^0 \tag{32b}$$

are available. After each coupling iteration, the difference between the vectors from the current coupling iteration and the vectors from the previous coupling iteration is calculated

$$\Delta\boldsymbol{r}^{k-1} = \boldsymbol{r}^k - \boldsymbol{r}^{k-1} \tag{33a}$$

$$\Delta\tilde{\boldsymbol{d}}^{k-1} = \tilde{\boldsymbol{d}}^k - \tilde{\boldsymbol{d}}^{k-1}. \tag{33b}$$

This yields a set of differences $\Delta\boldsymbol{r}^j$ and the corresponding set of differences $\Delta\tilde{\boldsymbol{d}}^j$ which both grow in each coupling iteration ($j = 0, \ldots, k-1$). These vectors are stored as the columns of the matrices

$$\boldsymbol{V}^k = \begin{bmatrix}\Delta\boldsymbol{r}^{k-1} & \Delta\boldsymbol{r}^{k-2} & \ldots & \Delta\boldsymbol{r}^1 & \Delta\boldsymbol{r}^0\end{bmatrix} \tag{34a}$$

and

$$\boldsymbol{W}^k = \begin{bmatrix}\Delta\tilde{\boldsymbol{d}}^{k-1} & \Delta\tilde{\boldsymbol{d}}^{k-2} & \ldots & \Delta\tilde{\boldsymbol{d}}^1 & \Delta\tilde{\boldsymbol{d}}^0\end{bmatrix}. \tag{34b}$$

The number of columns in $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ is indicated with v which is not always equal to k as will be explained further and which is generally much smaller than the number of rows u. Nevertheless, in simulations with a low number of degrees of freedom on the interface, it is possible that the number of columns has to be limited to u by discarding the rightmost columns.

The vector $\Delta\boldsymbol{r} = \boldsymbol{0} - \boldsymbol{r}^k$ is approximated as a linear combination of the known $\Delta\boldsymbol{r}^j$

$$\Delta\boldsymbol{r} \approx \boldsymbol{V}^k\boldsymbol{c}^k \tag{35}$$

with $\boldsymbol{c}^k \in \mathbb{R}^{v\times 1}$ the coefficients of the decomposition. Because $v \leq u$, Eq. (35) is an overdetermined set of equations for the elements of $\boldsymbol{c}^k$ and hence the least-squares solution to this linear system is calculated. For that reason, the so-called economy-size QR-decomposition of $\boldsymbol{V}^k$ is calculated using Householder transformations [42]

$$\boldsymbol{V}^k = \boldsymbol{Q}^k\boldsymbol{R}^k, \tag{36}$$

with $\boldsymbol{Q}^k \in \mathbb{R}^{u\times v}$ an orthogonal matrix and $\boldsymbol{R}^k \in \mathbb{R}^{v\times v}$ an upper triangular matrix. The coefficient vector $\boldsymbol{c}^k$ is then determined by solving the triangular system

$$\boldsymbol{R}^k\boldsymbol{c}^k = \boldsymbol{Q}^{k\mathrm{T}}\Delta\boldsymbol{r} \tag{37}$$

using back substitution. If a $\Delta\boldsymbol{r}^{j'}$ vector is (almost) a linear combination of other $\Delta\boldsymbol{r}^j$ vectors, one of the diagonal elements of $\boldsymbol{R}^k$ will (almost) be zero. Therefore, the equation corresponding to that row of $\boldsymbol{R}^k$ cannot be solved during the back substitution. If a small diagonal element is detected, the corresponding columns in $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ are removed. Subsequently, the QR-decomposition (Eq. (36)) is performed again. This procedure is repeated until none of the diagonal elements is too small. The tolerance ε_s for the detection of small diagonal elements depends on how accurately the flow equations and structural equations are solved. An appropriate value for ε_s can be determined by analyzing the change of the vector $\tilde{\boldsymbol{d}}$ due to a small perturbation of the vector $\boldsymbol{d}$. If the perturbation is too small, the resulting change will be numerical noise. The value of ε_s should be chosen so that the change of $\tilde{\boldsymbol{d}}$ has a physical meaning if the perturbation of $\boldsymbol{d}$ has an L^2-norm larger than ε_s. If the solution of the flow equations and the structural equations is calculated with more significant digits, for example by using stricter convergence criteria inside the solvers, then a smaller value of ε_s can be used.

The $\Delta\tilde{\boldsymbol{d}}$ that corresponds to $\Delta\boldsymbol{r}$ is subsequently calculated as a linear combination of the previous $\Delta\tilde{\boldsymbol{d}}^j$, analogous to Eq. (35), giving

$$\Delta\tilde{\boldsymbol{d}} = \boldsymbol{W}^k\boldsymbol{c}^k. \tag{38}$$

From Eq. (29), it follows that

$$\Delta\boldsymbol{r} = \Delta\tilde{\boldsymbol{d}} - \Delta\boldsymbol{d} \tag{39}$$

and substitution of Eq. (38) in Eq. (39) results in

$$\Delta\boldsymbol{d} = \boldsymbol{W}^k\boldsymbol{c}^k - \Delta\boldsymbol{r}. \tag{40}$$

Because the coefficients $\boldsymbol{c}^k$ are a function of $\Delta\boldsymbol{r}$, Eq. (40) shows how $\Delta\boldsymbol{d}$ can be approximated for a given $\Delta\boldsymbol{r}$. Hence, Eq. (40) can be seen as a procedure to calculate the product of the approximation for the inverse of the Jacobian and a vector $\Delta\boldsymbol{r} = -\boldsymbol{r}^k$

$$\Delta\boldsymbol{d} = \widehat{\left(\boldsymbol{\mathcal{R}}'^k\right)}^{-1}\Delta\boldsymbol{r} = \boldsymbol{W}^k\boldsymbol{c}^k + \boldsymbol{r}^k. \tag{41}$$

The standard IQN-ILS algorithm with one grid level as described above has a lot in common with the Interface-GMRES algorithm [25], but there are also significant differences, which are discussed in Appendix B.

Algorithm 1 shows the Multi-Level IQN-ILS (ML-IQN-ILS) algorithm in detail. Lines 8 to 18 are the standard IQN-ILS algorithm as described above. Around the standard algorithm, an additional loop over the grid levels is added (line 5). First, the coupled solution is calculated on the coarsest grid level. Then, starting from that solution, coupling iterations on the following, finer grid level are performed. These steps are subsequently repeated for all grid levels until the solution on the finest grid has been found. The variable ℓ ensures that at least one coupling iteration is performed on each grid level. A simplified flowchart of ML-IQN-ILS is depicted in Figure 5.

The displacement and the residual are not changed when the grid level i changes, as both are defined on the coupling grid. As explained above, the coupling algorithm itself works with a unique coupling grid, which determines the dimension

Algorithm 1 The multi-level IQN-ILS (ML-IQN-ILS) algorithm.

```
$k = \ell = 0$
$\boldsymbol{d}^k = \frac{5}{2}\boldsymbol{d}^n - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$
$\tilde{\boldsymbol{d}}^k = \boldsymbol{\mathcal{S}}_1 \circ \boldsymbol{\mathcal{F}}_1(\boldsymbol{d}^k)$
$\boldsymbol{r}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k$
for $i = 1$ to $g$ do
    while $||\boldsymbol{r}^k||_2 > \varepsilon_i$ or $\ell = 0$ do
        $\ell = 1$
        if $k = 0$ then
            $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \omega \boldsymbol{r}^k$
        else
            construct $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$
            calculate QR-decomposition $\boldsymbol{V}^k = \boldsymbol{Q}^k \boldsymbol{R}^k$
            solve $\boldsymbol{R}^k \boldsymbol{c}^k = -\boldsymbol{Q}^{k\mathrm{T}} \boldsymbol{r}^k$
            $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \boldsymbol{W}^k \boldsymbol{c}^k + \boldsymbol{r}^k$
        end if
        $\tilde{\boldsymbol{d}}^{k+1} = \boldsymbol{\mathcal{S}}_i \circ \boldsymbol{\mathcal{F}}_i(\boldsymbol{d}^{k+1})$
        $\boldsymbol{r}^{k+1} = \tilde{\boldsymbol{d}}^{k+1} - \boldsymbol{d}^{k+1}$
        $k = k + 1$
    end while
    $\ell = 0$
end for
for $i = 1$ to $g - 1$ do
    synchronize $\boldsymbol{\mathcal{F}}_i$ and $\boldsymbol{\mathcal{S}}_i$ with $\boldsymbol{\mathcal{F}}_g$ and $\boldsymbol{\mathcal{S}}_g$
end for
```

of the approximation for the inverse of the Jacobian. The different grid levels that are used for the multi-level technique are only present inside the flow solver and structural solver. The solvers have to interpolate the data from the boundary of their grid to the coupling grid of the coupling code. The interpolation on the fluid-structure interface is thus the responsibility of the solvers (or a layer around the actual solvers). In this way, the acceleration of the coupling iterations and the interpolation of the data on the fluid-structure interface are completely separated, which facilitates the implementation.

Because the coupling algorithm operates on the coupling grid, the difference between $\boldsymbol{r}$ and $\tilde{\boldsymbol{d}}$ in consecutive coupling iterations is always interpolated to a fixed number of grid points, regardless of the current grid level. As a result, the modes that have been generated on a coarse grid level can be used to accelerate the coupling iterations on the finer grid levels. The same least-squares model is used for all grid levels so the number of columns in the matrices $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ increases on each grid level.

Michler et al. [43] demonstrated that Krylov vectors from previous iterations and/or time steps can be reused to accelerate the convergence of the coupling iterations in a time step for Interface-GMRES. The reuse of vectors from previous time steps has also successfully been applied to IQN-ILS and IBQN-LS [44]. In ML-IQN-ILS, a similar approach is followed but now data is reused from previous (=coarser) grid levels instead of from previous iteration and/or time levels.

Because the matrices $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ have to contain at least one column to perform a quasi-Newton step, a Gauss-Seidel step using relaxation with factor ω (line 9) is performed in each time step when $k = 0$. The relaxation is added because for most strongly coupled problems, Gauss-Seidel iterations without relaxation are unstable and the displacement of the interface could change significantly in that case. Often, the flow solver and/or structural solver would then not be able to calculate the solution without this relaxation because their respective inputs would be too far from the previous coupling iteration. Without this relaxation, excessive grid distortion could occur, possibly leading to divergence of the iterations inside the solvers. The counter k is only set to zero at the beginning of the time step (line 1) and not when the coupling iterations on a following grid level start. As a result, this single relaxation step is only performed on the coarsest grid level.

The numerical experiments in Section 5 indicate that vectors $\Delta\boldsymbol{r}^j$ and $\Delta\tilde{\boldsymbol{d}}^j$ from a coarse grid level can accelerate the coupling iterations on a fine grid level. However, it should be noted that the difference between $\boldsymbol{r}$ and $\tilde{\boldsymbol{d}}$ in the last coupling iteration on a certain grid level i and the first coupling iteration on the following grid level $i+1$,

$$\Delta \boldsymbol{r}^{j-1} = \boldsymbol{\mathcal{R}}_{i+1}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{R}}_{i}(\boldsymbol{d}^{j-1}) \tag{42a}$$

$$\Delta \tilde{\boldsymbol{d}}^{j-1} = \boldsymbol{\mathcal{S}}_{i+1} \circ \boldsymbol{\mathcal{F}}_{i+1}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{S}}_{i} \circ \boldsymbol{\mathcal{F}}_{i}(\boldsymbol{d}^{j-1}), \tag{42b}$$

should not be added to $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$. Otherwise, the approximation for the inverse of $\boldsymbol{\mathcal{R}}'$ would not only relate a change of the residual to a change of the interface's displacement, but would also represent the additional features that become visible due to a change of the grid level. If these differences are added to $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ nonetheless, the convergence of the coupling iterations on grid level $i+1$ is hampered in the numerical experiments. Although it would be useful to know the change $\boldsymbol{\mathcal{S}}_{i} \circ \boldsymbol{\mathcal{F}}_{i}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{S}}_{i} \circ \boldsymbol{\mathcal{F}}_{i}(\boldsymbol{d}^{j-1})$ due to $\boldsymbol{\mathcal{R}}_{i}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{R}}_{i}(\boldsymbol{d}^{j-1})$ or the change $\boldsymbol{\mathcal{S}}_{i+1} \circ \boldsymbol{\mathcal{F}}_{i+1}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{S}}_{i+1} \circ \boldsymbol{\mathcal{F}}_{i+1}(\boldsymbol{d}^{j-1})$ due to $\boldsymbol{\mathcal{R}}_{i+1}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{R}}_{i+1}(\boldsymbol{d}^{j-1})$, the differences in Eqs. (42) are biased because the terms have been calculated on two different grid levels. The differences in Eqs. (42) do not only contain information on the change from coupling iteration k to $k+1$, but also on the change from grid level i to $i+1$. As grid level $i+1$ is finer than grid level i, additional features can be resolved on grid level $i+1$. When the differences in Eqs. (42) are not used, the number of columns in $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ at the end of the time step is equal to the number of coupling iterations minus the number of grid levels.

On line 6, it can be seen that the convergence criterion depends on the grid level. This makes it possible to use a less stringent criterion for the coupling iterations on the coarse grid levels as no accurate solution of the coupled problem is desired there. An alternative is to limit the number of coupling iterations on the coarse grid levels.

Lines 22 to 24 show that synchronization is necessary at the end of the time step. Once the solution has been found on the finest grid level, all degrees of freedom on the coarser grid levels have to be corrected. A possible approach to the synchronization is to interpolate the data in the entire fluid and solid domain from the finest grid level to all other grid levels. If no such mechanism is available because the solvers are black boxes, the interface displacement and traction calculated during the coupling iterations on the finest grid level can be applied to the interface of the coarser grid levels and the flow equations and structural equations can be solved once more on all but the finest grid level, with this displacement and traction as boundary condition. However, this approach can result in a difference between the solution in the fluid and solid domain after a (large) number of time steps. For example, some vortices can be resolved on a fine grid but not on a coarse grid, which can lead to a different flow field in time. The synchronization of all coarse grid levels can be performed in parallel.

As ML-IBQN-LS is obtained from IBQN-LS in a similar way as ML-IQN-ILS from IQN-ILS, this derivation is provided in Appendix A.

5. Numerical results

All numerical results have been obtained using a dedicated cluster node with two quad-core Intel Xeon X5355 processors.

5.1. Unsteady flow in a 1D tube

The first example is the unsteady, incompressible flow in a straight, flexible tube with a circular cross-section and length L, depicted in Figure 6. This example is straightforward to implement and yet it is a representative test for a coupling technique [34, 35]. The numerical model is one-dimensional and gravity and viscosity are not taken into account. The flow is governed by the continuity and momentum equation which can be written in conservative form as

$$\frac{\partial a}{\partial t} + \frac{\partial av}{\partial z} = 0 \tag{43a}$$

$$\frac{\partial av}{\partial t} + \frac{\partial av^2}{\partial z} + \frac{1}{\rho_f}\left(\frac{\partial ap}{\partial z} - p\frac{\partial a}{\partial z}\right) = 0 \tag{43b}$$

with z the coordinate along the axis of the tube, $a = \pi r^2$ the cross-sectional area of the tube and r the inner radius. t is the time, v the velocity along the axis of the tube, p the pressure and ρ_f the density of the fluid.

The behaviour of the elastic tube wall is described with a Hookean constitutive relation. The structure contains no mass, as the inertia of the tube wall is neglected with regard to that of the fluid. An axisymmetric model is used in the coordinate system (r,φ,z), with φ the angle in the cross-sectional plane as indicated in Figure 6. The stress in the tube wall in circumferential direction $\sigma_{\varphi\varphi}$ is approximated as

$$\sigma_{\varphi\varphi} = E\frac{r - r_o}{r_o} + \sigma_{\varphi\varphi o} \tag{44}$$

with E the Young's modulus and r_o the radius for which $\sigma_{\varphi\varphi} = \sigma_{\varphi\varphi o}$. As other stress components are neglected, this model only allows for radial motion of the tube wall. The force balance on the fluid-structure interface is

$$rp = \sigma_{\varphi\varphi}h \tag{45}$$

with h the thickness of the tube wall. By substituting the constitutive equation (Eq. (44)), $r_o p_o = \sigma_{\varphi\varphi o} h$ and $a = \pi r^2$ in Eq. (45), the following relation holds

$$a = a_o \left(\frac{\frac{p_o}{2\rho_f} - c_{MK}^2}{\frac{p}{2\rho_f} - c_{MK}^2} \right)^2 \tag{46}$$

with the Moens-Korteweg wave speed given by

$$c_{MK} = \sqrt{\frac{Eh}{2\rho_f r_o}}. \tag{47}$$

The resulting wave speed c is

$$c^2 = \frac{a}{\rho_f \frac{\mathrm{d}a}{\mathrm{d}p}} = c_{MK}^2 - \frac{p}{2\rho_f}. \tag{48}$$

which is used to impose a non-reflecting boundary

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{1}{c\rho_f} \frac{\mathrm{d}p}{\mathrm{d}t} \tag{49}$$

at the outlet.

The tube is discretized using a one-dimensional grid with N cells of length Δz, as indicated in Figure 6. The fluid velocity and pressure are stored in the cell centres. Central discretization is used for all terms in the continuity and momentum equation, except for the convective term in the momentum equation which is discretized with a first-order upwind scheme. The time discretization scheme is backward Euler and the time step is indicated with Δt. The conservation of mass and momentum in a control volume around cell centre j is expressed by the following system of equations

$$\frac{\Delta z}{\Delta t} \left(a_j - a_j^n \right) + v_{j+1/2} a_{j+1/2} - v_{j-1/2} a_{j-1/2} - \frac{\alpha}{\rho_f} \left(p_{j+1} - 2p_j + p_{j-1} \right) = 0 \tag{50a}$$

$$\begin{aligned} \frac{\Delta z}{\Delta t} \left(v_j a_j - v_j^n a_j^n \right) + v_j v_{j+1/2} a_{j+1/2} - v_{j-1} v_{j-1/2} a_{j-1/2} \\ + \frac{1}{2\rho_f} \left(a_{j+1/2} \left(p_{j+1} - p_j \right) + a_{j-1/2} \left(p_j - p_{j-1} \right) \right) = 0 \end{aligned} \tag{50b}$$

for $v_j \geq 0$. The subscripts j, $j+1$ and $j-1$ indicate the cell centres ($j = 1, \ldots, N$) and the subscript $j \pm 1/2$ signifies the values calculated at the cell interfaces, $v_{j-1/2} = (v_{j-1} + v_j)/2$ and $v_{j+1/2} = (v_j + v_{j+1})/2$. A pressure stabilization term [45] with coefficient $\alpha = a_o/\left(v_o + \Delta z/\Delta t\right)$ has been added in the continuity equation to prohibit pressure wiggles due to central discretization of the pressure in the momentum equation, with v_o the reference fluid velocity.

The equation for the structure (Eq. (46)) does not require further discretization as it can directly be used to calculate the cross-sectional area of a segment for a given pressure in that segment. The pressure-outlet condition in Eq. (49) is discretized as

$$p_{out} = 2\rho_f \left[c_{MK}^2 - \left(\sqrt{c_{MK}^2 - \frac{p_{out}^n}{2\rho_f}} - \frac{v_{out} - v_{out}^n}{4} \right)^2 \right], \tag{51}$$

which takes into account the dependence of c on p (Eq. (48)) in the integration from time level n to $n+1$. At the inlet, the velocity is imposed as

$$v_{in} = v_o + \frac{v_o}{10}\sin^2(\pi n \tau). \tag{52}$$

The pressure at the inlet and the velocity at the outlet are linearly extrapolated

$$p_{in} = 2p_1 - p_2 \tag{53a}$$

$$v_{out} = 2v_N - v_{N-1}. \tag{53b}$$

The parameters of this case are grouped in two dimensionless parameters, namely

$$\kappa = \frac{c_o}{v_o} = \frac{\sqrt{\frac{Eh}{2\rho_f r_o} - \frac{p_o}{2\rho_f}}}{v_o} \quad \text{and} \quad \tau = \frac{v_o \Delta t}{L}. \tag{54}$$

In all simulations presented in this section, the dimensionless stiffness is $\kappa = 10$ and the dimensionless time step is $\tau = 0.01$. One period of the inlet boundary condition, i.e. 100 time steps, is simulated. The initial conditions are a dimensionless velocity of $v/c_o = 0.1$, a dimensionless cross-sectional area of $a/a_o = 1$ and a dimensionless pressure of $p/(\rho_f c_o^2) = 0$. The interpolation on the fluid-structure interface is performed as described in Section 3 with $m = 5$. The convergence tolerance is $\varepsilon_i = 10^{-5}||\boldsymbol{r}^0||_2$ for all grid levels.

Table 1 demonstrates that if a coarse grid level with 10^3 points and a fine grid level with 10^4 points are used, both ML-IQN-ILS and ML-IBQN-LS lead to a reduction in the overall simulation time by approximately 35 % compared to IQN-ILS and IBQN-LS. The average number of coupling iterations per time step on the fine grid level is approximately half of that in a simulation with a fine grid only. If a coarse grid level with 4×10^3 points and a fine grid level with 10^4 points are used, the reduction of the simulation time obtained by using the multi-level techniques is smaller. Therefore, it can be concluded that the ratio of the number of degrees of freedom on the fine grid level to the number of degrees of freedom on the coarse grid level has to be sufficiently large to obtain a significant reduction of the simulation time. In this one-dimensional simulation, the grid is refined with a factor 10. For two- and three-dimensional simulations, a smaller factor (e.g. 4) will yield the required difference in terms of number of degrees of freedom.

By adding an intermediate grid level with 4×10^3 grid points between the coarse grid level with 10^3 points and the fine grid level with 10^4 points, the average number of coupling iterations on the finest grid level decreases slightly compared to the simulation with two grid levels. However, the additional coupling iterations on the intermediate grid level more or less outweigh the small reduction of the number of coupling iterations on the fine grid so that the duration of the simulation with three grid levels is similar to the duration of the simulation with two grid levels. Therefore, only two grid levels will be used in the remainder of this work. Nevertheless, more than two grid levels might reduce the duration of the simulation significantly compared to two grid levels in other cases, for example if there is a larger difference in number of degrees of freedom between the coarsest and finest grid level.

The normalized time 1.0 in Table 1 refers to an absolute time of 3m45s for the entire simulation with 100 time steps and 10000 grid points on the fine level using a single core of an Intel Xeon X5355 processor. However, more than 99 % of this time is spent in the flow solver but not in the coupling algorithm itself. The absolute time thus mainly measures the efficiency of the implementation of the flow solver for this 1D example. Absolute time information can only be used as a means of comparison between different partitioned techniques if exactly the same solvers, the same compiler settings and the same hardware are used. By contrast, the average number of coupling iterations per time step (as listed in Table 1) is more or less independent of these factors. Nevertheless, the absolute time can be useful for a comparison with monolithic solution techniques, but it is also dependent on hardware and implementation.

Figure 7 shows the convergence of the coupling iterations during the first time step when using ML-IQN-ILS and ML-IBQN-LS with two and three grid levels. When the coupling algorithm proceeds from the first grid level to the second, finer grid level, the residual increases but the initial residual on the second grid level is approximately a factor 10^2 smaller than the initial residual on the first grid level. Moreover, the residual decreases faster in the first few coupling iterations on the second (and third) grid level than in the first few coupling iterations on the first grid level. The difference between the initial residual on the second and third grid level is smaller than between the first and second grid level.

As mentioned above, different convergence tolerances or a limited number of coupling iterations can be used on the different grid levels to avoid that the coupled solution is calculated too accurately on the coarser grid levels. Table 2 lists the number of coupling iterations per time step as well as the relative duration of simulations with different limits for the number of coupling iterations on the coarse grid. In the first test with each algorithm, the number of coarse grid coupling iterations is not limited. Then, the number of coarse grid coupling iterations is successively limited to 8, 7 and 6. As the number of coarse grid iterations is reduced, the number of fine grid iterations increases slowly, which offsets the gain on the coarse grid. As a result, this limit on the number of coarse grid iterations causes an increase of the duration of the simulation.

5.2. Propagation of a pressure wave in a 3D tube

The second case is a simulation of a straight flexible tube with radius 0.005 m and length 0.05 m, as described by Fernandez and Moubachir [46], Formaggia et al. [47], Gerbeau and Vidrascu [48]. This tube is a simplified model for a large artery. The finite volume flow solver uses second-order discretization for the pressure and first-order upwind for the momentum. It solves the Navier-Stokes equations in arbitrary Lagrangian-Eulerian (ALE) formulation with PISO pressure-velocity coupling and the first-order backward Euler time integration scheme. A spring model is used for the ALE grid movement in the flow solver. In this model, the grid edges are replaced by linear springs and the resulting equations are solved by performing 20 fixed-point iterations. The finite element structural solver uses implicit Hilber-Hughes-Taylor time integration of shell elements with 8 nodes and takes into account the geometric nonlinearities due to the large deformation of the structure.

The tube's wall is a linear elastic material with density 1200 kg/m^3, Young's modulus 3×10^5 N/m^2, Poisson's ratio 0.3 and thickness 0.001 m. The structure is clamped in all directions at the inlet and outlet. The fluid is incompressible and has a density of 1000 kg/m^3 and a viscosity of 0.003 Pas. Both the fluid and the structure are initially at rest. During the first 3×10^{-3} s, an overpressure of 1333.2 N/m^2

is applied at the inlet. The wave propagates through the tube during 10^{-2} s, simulated with time steps of 10^{-4} s. Pressure contours on the fluid-structure interface are shown in Figure 8 and they correspond well with those in [46–48].

Table 3 lists the number of coupling iterations per time step and per grid level, averaged over the entire simulation, and the relative duration of the simulations. Only two grid levels with an unlimited number of coupling iterations on the coarse grid have been used, as the 1D simulations with three grid levels and a limited number of coarse grid iterations did not result in significant reductions of the simulation's duration. The coarse grid level contains 34944+1824 degrees of freedom for the flow and the structure, respectively. For the fine grid level, each direction is refined with a factor 4, giving 2247168+28032 degrees of freedom. Because the structure is not a volume but a surface with one layer of shell elements, the number of degrees of freedom for the solid increases with a factor 16 and not 64. The interpolation on the fluid-structure interface is performed as described in Section 3 with $m = 81$, which provides a sufficiently smooth interpolation.

In the simulation with two grid levels, the number of coupling iterations on the fine grid is reduced by approximately 50 % compared to a simulation with a fine grid only. ML-IQN-ILS and ML-IBQN-LS are equally fast for this case. As the cost of the coupling iterations on the coarse grid level is relatively small, the duration of the simulation also decreases by approximately 50 %. In this simulation, six cores are used for the fluid and two cores for the solid. Moreover, the same number of cores is used on both grid levels. The convergence tolerance is $\varepsilon_i = 10^{-3}||\boldsymbol{r}^0||_2$ for all grid levels.

The normalized time 1.0 in Table 3 refers to an absolute time of 20h33m for the entire simulation with 100 time steps. However, more than 99 % of this time is spent in the flow solver and structural solver. The absolute time thus mainly measures the efficiency of these solvers. While absolute time can be useful for a comparison with monolithic solution techniques, the average number of coupling iterations per time step should be used for a comparison with other partitioned techniques.

The optimal number of grid levels and the ratio between the number of degrees of freedom in the different levels is obviously problem dependent. Nevertheless, based on the numerical experiments, the authors suggest using two grid levels. The ratio between the number of degrees of freedom in the fine and coarse levels is recommended to be 10 or more.

Figure 9 depicts the convergence of the coupling iterations in the first time step for ML-IQN-ILS and ML-IBQN-LS. When the coupling algorithm proceeds from the coarse to the fine grid level, the residual increases but the initial residual on the fine grid level is still at least 2 orders of magnitude smaller than the initial residual on the coarse grid level. Moreover, the convergence during the first few coupling iterations on the fine grid level is faster than during the first few coupling iterations on the coarse grid level.

From the numerical results presented above, the performance improvement of the multi-level IQN-ILS and IBQN-LS over the standard, one-level IQN-ILS and IBQN-LS is clear. The comparison with other partitioned techniques can be derived by combining these results with [44], which indicates that standard IQN-ILS and IBQN-LS are faster than Interface-GMRES and Gauss-Seidel iterations with Aitken relaxation.

6. Conclusions

A new class of multi-level coupling techniques for partitioned simulation of fluid-structure interaction has been developed. These techniques are based on the fundamental insight from stability analyses on Gauss-Seidel coupling iterations with a Dirichlet-Neumann decomposition that in the difference between the current and the correct interface displacement, the Fourier modes with a low wave number are most unstable. The ML-IQN-ILS technique has been described in detail, starting from the standard algorithm with one grid level. This multi-level technique first calculates the coupled solution on the coarsest grid level and subsequently uses that solution as the starting point for the coupling iterations on the following, finer grid level. Moreover, the approximation for the inverse of the Jacobian constructed on the coarser grid levels accelerates the convergence of the coupling iterations on the finer grid levels. The ML-IBQN-LS technique can be derived similarly. The numerical results show that these multi-level algorithms can reduce the duration of a partitioned fluid-structure interaction simulation, if the difference in number of degrees of freedom between the grid levels is sufficient.

Acknowledgments

J. Degroote gratefully acknowledges a Ph.D. fellowship of the Research Foundation - Flanders (FWO).

Appendix A ML-IBQN-LS

In this appendix, it is explained how the ML-IBQN-LS algorithm can be derived from the standard IBQN-LS technique. The standard IBQN-LS coupling technique solves the FSI problem written as

$$\begin{cases} \boldsymbol{\mathcal{F}}(\boldsymbol{d}) - \boldsymbol{s} = \boldsymbol{0} \\ \boldsymbol{\mathcal{S}}(\boldsymbol{s}) - \boldsymbol{d} = \boldsymbol{0} \end{cases} \tag{55}$$

with block Newton-Raphson iterations of the Gauss-Seidel type. The linear system

$$\begin{bmatrix} \widehat{\boldsymbol{\mathcal{F}}'} & -\boldsymbol{I} \\ -\boldsymbol{I} & \widehat{\boldsymbol{\mathcal{S}}'} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{d} \\ \Delta\boldsymbol{s} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\mathcal{F}}(\boldsymbol{d}) - \boldsymbol{s} \\ \boldsymbol{\mathcal{S}}(\boldsymbol{s}) - \boldsymbol{d} \end{bmatrix} \tag{56}$$

is thus first solved for $\Delta\boldsymbol{d}$, followed by an update of $\boldsymbol{d}$ and the right-hand side. Subsequently, the modified system is solved for $\Delta\boldsymbol{s}$ and afterwards $\boldsymbol{s}$ is updated. The input and output of the flow solver are denoted as $\boldsymbol{d}^{k+1}$ and $\tilde{\boldsymbol{s}}^{k+1}$ and the input and output of the structural solver as $\boldsymbol{s}^{k+1}$ and $\tilde{\boldsymbol{d}}^{k+1}$.

Starting from the displacement $\boldsymbol{d}^k$ that was given as input to the flow solver in the previous coupling iteration, the displacement $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \Delta\boldsymbol{d}^k$ is calculated by solving the system

$$\left(\boldsymbol{I} - \widehat{\boldsymbol{\mathcal{S}}'^k}\widehat{\boldsymbol{\mathcal{F}}'^k}\right)\Delta\boldsymbol{d}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k + \widehat{\boldsymbol{\mathcal{S}}'^k}(\tilde{\boldsymbol{s}}^k - \boldsymbol{s}^k) \tag{57}$$

for $\Delta\boldsymbol{d}^k$. As the Jacobians of the black-box functions are unknown, they are approximated by a least-squares model. The linear system in Eq. (57) is solved in a matrix-free way with an iterative Krylov solver like the Generalized Conjugate Residual (GCR) method [49] or the mathematically equivalent Generalized Minimal Residual (GMRES) method [50]. The matrix on the left-hand side of Eq. (57) and thus the approximate Jacobians $\widehat{\boldsymbol{\mathcal{F}}'^k}$ and $\widehat{\boldsymbol{\mathcal{S}}'^k}$ do not have to be calculated explicitly; a procedure to calculate the product of these matrices with a vector is sufficient.

The procedure to calculate the product of the approximate Jacobian $\widehat{\boldsymbol{\mathcal{F}}'^k}$ with a vector uses the previous inputs

$$\boldsymbol{d}^k, \quad \ldots, \quad \boldsymbol{d}^0 \tag{58a}$$

and the corresponding outputs

$$\tilde{\boldsymbol{s}}^k = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^k), \quad \ldots, \quad \tilde{\boldsymbol{s}}^0 = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^0). \tag{58b}$$

of the flow solver. After each coupling iteration, the difference between the vectors from the current coupling iteration and the vectors from the previous coupling iteration is calculated.

$$\Delta \boldsymbol{d}^{k-1} = \boldsymbol{d}^{k} - \boldsymbol{d}^{k-1} \tag{59a}$$

$$\Delta \tilde{\boldsymbol{s}}^{k-1} = \tilde{\boldsymbol{s}}^{k} - \tilde{\boldsymbol{s}}^{k-1} \tag{59b}$$

All $\Delta \boldsymbol{d}^{j}$ and $\Delta \tilde{\boldsymbol{s}}^{j}$ ($j = 0, \ldots, k-1$) from the current time step are stored as columns of the matrices $\boldsymbol{V}_{f}^{k}$ and $\boldsymbol{W}_{f}^{k}$, with the subscript f referring to the flow solver. Subsequently, the economy-size QR-decomposition of $\boldsymbol{V}_{f}^{k}$ is calculated. To determine the product of $\widehat{\boldsymbol{\mathcal{F}}'^{k}}$ with a vector $\Delta \boldsymbol{d}$, the triangular system

$$\boldsymbol{R}_{f}^{k} \boldsymbol{c}_{f}^{k} = \boldsymbol{Q}_{f}^{k\,\mathrm{T}} \Delta \boldsymbol{d} \tag{60}$$

is solved for $\boldsymbol{c}_{f}^{k}$, after which the matrix-vector product is calculated as

$$\widehat{\boldsymbol{\mathcal{F}}'^{k}} \Delta \boldsymbol{d} = \boldsymbol{W}_{f}^{k} \boldsymbol{c}_{f}^{k}. \tag{61}$$

The product of $\widehat{\boldsymbol{\mathcal{S}}'^{k}}$ with a vector is calculated analogously, based on the inputs and outputs of the structural solver.

Once $\boldsymbol{d}^{k+1}$ has been obtained, the corresponding traction distribution $\tilde{\boldsymbol{s}}^{k+1} = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{k+1})$ is calculated and the matrices $\boldsymbol{V}_{f}^{k+1}$, $\boldsymbol{W}_{f}^{k+1}$, $\boldsymbol{Q}_{f}^{k+1}$ and $\boldsymbol{R}_{f}^{k+1}$ are constructed. To calculate the traction distribution $\boldsymbol{s}^{k+1} = \boldsymbol{s}^{k} + \Delta \boldsymbol{s}^{k}$ that has to be applied on the structure, the system

$$\left(\boldsymbol{I} - \widehat{\boldsymbol{\mathcal{F}}'^{k+1}} \widehat{\boldsymbol{\mathcal{S}}'^{k}} \right) \Delta \boldsymbol{s}^{k} = \tilde{\boldsymbol{s}}^{k+1} - \boldsymbol{s}^{k} + \widehat{\boldsymbol{\mathcal{F}}'^{k+1}} (\tilde{\boldsymbol{d}}^{k} - \boldsymbol{d}^{k+1}) \tag{62}$$

is solved, again with the matrix-free iterative solver. Each time the solution to either the flow problem or the structural problem has been calculated, the procedure for the product of the corresponding solver's approximate Jacobian with a vector is improved by means of that solver's latest input and output.

The matrices $\boldsymbol{V}_{f}^{k}$, $\boldsymbol{W}_{f}^{k}$, $\boldsymbol{V}_{s}^{k}$ and $\boldsymbol{W}_{s}^{k}$ have to contain at least one column to calculate the quasi-Newton update; otherwise a relaxation with factor ω is used for the interface's displacement (line 11 in Algorithm 2) and the traction distribution is passed on without modification (line 20).

Algorithm 2 shows the Multi-Level IBQN-LS (ML-IBQN-LS) algorithm. Lines 10 to 29 are the standard IBQN-LS algorithm, which is wrapped in a loop over the grid levels (line 7). Coupling iterations are first performed on the coarsest grid level. The solution of these coupling iterations is then used as initial value for

Algorithm 2 One time step with the multi-level IBQN-LS (ML-IBQN-LS) algorithm.

```
$k = \ell = 0$
$\boldsymbol{d}^k = \frac{5}{2}\boldsymbol{d}^n - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$
$\tilde{\boldsymbol{s}}^k = \boldsymbol{\mathcal{F}}_1(\boldsymbol{d}^k)$
$\boldsymbol{s}^k = \tilde{\boldsymbol{s}}^k$
$\tilde{\boldsymbol{d}}^k = \boldsymbol{\mathcal{S}}_1(\boldsymbol{s}^k)$
$\boldsymbol{r}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k$
for $i = 1$ to $g$ do
    while $||\boldsymbol{r}^k||_2 > \varepsilon_i$ or $\ell = 0$ do
        $\ell = 1$
        if $k = 0$ then
            $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \omega\boldsymbol{r}^k$
        else
            construct $\boldsymbol{V}_s^k$ and $\boldsymbol{W}_s^k$
            calculate QR-decomposition $\boldsymbol{V}_s^k = \boldsymbol{Q}_s^k\boldsymbol{R}_s^k$
            solve Eq. (57) for $\Delta\boldsymbol{d}^k$
            $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \Delta\boldsymbol{d}^k$
        end if
        $\tilde{\boldsymbol{s}}^{k+1} = \boldsymbol{\mathcal{F}}_i(\boldsymbol{d}^{k+1})$
        if $k = 0$ then
            $\boldsymbol{s}^{k+1} = \tilde{\boldsymbol{s}}^{k+1}$
        else
            construct $\boldsymbol{V}_f^{k+1}$ and $\boldsymbol{W}_f^{k+1}$
            calculate QR-decomposition $\boldsymbol{V}_f^{k+1} = \boldsymbol{Q}_f^{k+1}\boldsymbol{R}_f^{k+1}$
            solve Eq. (62) for $\Delta\boldsymbol{s}^k$
            $\boldsymbol{s}^{k+1} = \boldsymbol{s}^k + \Delta\boldsymbol{s}^k$
        end if
        $\tilde{\boldsymbol{d}}^{k+1} = \boldsymbol{\mathcal{S}}_i(\boldsymbol{s}^{k+1})$
        $\boldsymbol{r}^{k+1} = \tilde{\boldsymbol{d}}^{k+1} - \boldsymbol{d}^{k+1}$
        $k = k + 1$
    end while
    $\ell = 0$
end for
for $i = 1$ to $g - 1$ do
    synchronize $\boldsymbol{\mathcal{F}}_i$ and $\boldsymbol{\mathcal{S}}_i$ with $\boldsymbol{\mathcal{F}}_g$ and $\boldsymbol{\mathcal{S}}_g$
end for
```

the coupling iterations on the following, finer grid level. This procedure is subsequently repeated until the solution on the finest grid has been found. The variable ℓ ensures that at least one coupling iteration is performed on each grid level.

The coupling algorithm works with the coupling grid for all calculations, so all interpolations between the different grid levels and the coupling grid occur inside the flow solver and the structural solver. As a result, the same least-squares models for the flow solver and the structural solver can be used for all grid levels so that the coarse grid data accelerate the coupling iterations on the finer grid levels. The counter k is also set to zero at the beginning of the time step (line 1) only. Hence, the relaxation of the displacement on line 11 and the copy of the tractions on line 20 are only performed on the coarsest grid level. By means of a convergence tolerance depending on the grid level (line 8) or a limitation of the number of coupling iterations on the coarser grid levels, it can be avoided that the solution of the coupled problem is calculated too accurately on the coarse grid levels.

Because of the influence of the grid level, the difference between $\boldsymbol{d}$ and $\tilde{\boldsymbol{s}}$ in the last coupling iteration on a certain grid level i and the first coupling iteration on the following grid level $i+1$,

$$\Delta \boldsymbol{d}^{j-1} = \boldsymbol{d}^{j} - \boldsymbol{d}^{j-1} \tag{63a}$$

$$\Delta \tilde{\boldsymbol{s}}^{j-1} = \boldsymbol{\mathcal{F}}_{i+1}(\boldsymbol{d}^{j}) - \boldsymbol{\mathcal{F}}_{i}(\boldsymbol{d}^{j-1}), \tag{63b}$$

should not be added to $\boldsymbol{V}_f^k$ and $\boldsymbol{W}_f^k$. Similarly, the difference between $\boldsymbol{s}$ and $\tilde{\boldsymbol{d}}$ in the last coupling iteration on a certain grid level i and the first coupling iteration on the following grid level $i+1$,

$$\Delta \boldsymbol{s}^{j-1} = \boldsymbol{s}^{j} - \boldsymbol{s}^{j-1} \tag{64a}$$

$$\Delta \tilde{\boldsymbol{d}}^{j-1} = \boldsymbol{\mathcal{S}}_{i+1}(\boldsymbol{s}^{j}) - \boldsymbol{\mathcal{S}}_{i}(\boldsymbol{s}^{j-1}), \tag{64b}$$

should not be added to $\boldsymbol{V}_s^k$ and $\boldsymbol{W}_s^k$. All these differences are biased because the terms have been calculated on two different grid levels.

Synchronization between the different grid levels is also necessary for the ML-IBQN-LS algorithm (lines 33 to 35). Again, the interface displacement and traction resulting from the coupling iterations on the finest grid level have to be applied to the fluid-structure interface of all other grid levels. The fluid and structure subdomains of the other grid levels have to be updated accordingly, either by solving the equations once more or by interpolating the data from the finest grid level.

Appendix B Comparison between IQN-ILS and Interface-GMRES

The standard IQN-ILS algorithm with one grid level has a lot in common with the Interface-GMRES algorithm [25], as both algorithms use changes of the interface data from one coupling iteration to another and as both solve least-squares problems. However, there are also some significant differences. The major difference, as will be explained below, is that IQN-ILS constructs an approximation for the inverse of the Jacobian with $\Delta \boldsymbol{r}^j$ and $\Delta \tilde{\boldsymbol{d}}^j$ ($j = 0, \ldots, k-1$), while Interface-GMRES uses $\Delta \boldsymbol{r}^j$ and $\Delta \boldsymbol{d}^j$.

To perform a quasi-Newton step, IQN-ILS decomposes the residual of the last quasi-Newton step $\Delta \boldsymbol{r} = -\boldsymbol{r}^k$ on the set of $\Delta \boldsymbol{r}^j = \boldsymbol{r}^{j+1} - \boldsymbol{r}^j$ ($j = 0, \ldots, k-1$) by solving the least-squares problem

$$\boldsymbol{c}^k = \arg\min_{\boldsymbol{c}^k} ||\boldsymbol{r}^k + \sum_{j=0}^{k-1} c_j^k (\boldsymbol{r}^{j+1} - \boldsymbol{r}^j)||_2. \tag{65}$$

Using Eq. (41), the new interface displacement is then calculated as

$$\begin{aligned}
\boldsymbol{d}^{k+1} &= \boldsymbol{d}^k + \Delta \boldsymbol{d} && \text{(66a)}\\
&= \boldsymbol{d}^k + \sum_{j=0}^{k-1} c_j^k (\tilde{\boldsymbol{d}}^{j+1} - \tilde{\boldsymbol{d}}^j) + \boldsymbol{r}^k && \text{(66b)}\\
&= \tilde{\boldsymbol{d}}^k + \sum_{j=0}^{k-1} c_j^k (\tilde{\boldsymbol{d}}^{j+1} - \tilde{\boldsymbol{d}}^j) && \text{(66c)}\\
&= \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^k) + \sum_{j=0}^{k-1} c_j^k (\boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{j+1}) - \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^j)). && \text{(66d)}
\end{aligned}$$

Interface-GMRES, on the other hand, decomposes the residual of the previous quasi-Newton step $\Delta \boldsymbol{r} = -\boldsymbol{r}^*$ on a set of $\Delta \boldsymbol{r}^j = \boldsymbol{r}^{j+1} - \boldsymbol{r}^j$ ($j = 0, \ldots, k-1$), coming from a sequence of Gauss-Seidel iterations, by solving the least-squares problem

$$\boldsymbol{c}^k = \arg\min_{\boldsymbol{c}^k} ||\boldsymbol{r}^* + \sum_{j=0}^{k-1} c_j^k (\boldsymbol{r}^{j+1} - \boldsymbol{r}^*)||_2. \tag{67}$$

The interface displacement $\boldsymbol{d}^{**}$ in the new quasi-Newton step is then calculated as

$$
\begin{aligned}
\boldsymbol{d}^{**} &= \boldsymbol{d}^{*} + \Delta\boldsymbol{d} &\text{(68a)}\\
&= \boldsymbol{d}^{*} + \sum_{j=0}^{k-1} c_j^k (\boldsymbol{d}^{j+1} - \boldsymbol{d}^{*}), &\text{(68b)}
\end{aligned}
$$

with $\boldsymbol{d}^{*}$ the interface displacement from the previous quasi-Newton step. Hence, the interface displacement resulting from a quasi-Newton step is a linear combination of previous interface displacements in the case of Interface-GMRES, but not in the case of IQN-ILS.

As $\boldsymbol{d}^{**}$ is a linear combination of previous interface displacements, the dimension of the basis spanned by the $\Delta\boldsymbol{d}^{j}$ is not increased in the quasi-Newton steps of Interface-GMRES. The least-squares model thus improves in the quasi-Newton steps of IQN-ILS but not in those of Interface-GMRES. Therefore, Interface-GMRES needs a number of Gauss-Seidel iterations (optionally with relaxation and/or orthogonalization) between two quasi-Newton steps in order to improve the least-squares model. From Eq. (65), it can be seen that if $\boldsymbol{r}^{k}$ is orthogonal to the basis spanned by the $\Delta\boldsymbol{r}^{j}$ ($j = 0, \ldots, k-1$), then all decomposition coefficients c_j^k are zero. In that case, the quasi-Newton step of IQN-ILS is actually a Gauss-Seidel step ($\boldsymbol{d}^{k+1} = \boldsymbol{d}^{k} + \boldsymbol{r}^{k} = \tilde{\boldsymbol{d}}^{k}$) while Interface-GMRES does not change $\boldsymbol{d}$. In a comparison between the performance of both techniques, IQN-ILS appeared to be faster than Interface-GMRES [44].

References

[1] T. Tezduyar, S. Sathe, K. Stein, Solution techniques for the fully discretized equations in computation of fluid-structure interactions with the space-time formulations, Computer Methods in Applied Mechanics and Engineering 195 (41–43) (2006) 5743–5753.

[2] M. Khan, M. Moatamedi, M. Souli, T. Zeguer, Multiphysics out of position airbag simulation, International Journal of Crashworthiness 13 (2) (2008) 159–166.

[3] K. Billah, R. Scanlan, Resonance, Tacoma Narrows Bridge Failure, and Undergraduate Physics Textbooks, American Journal of Physics 59 (2) (1991) 118–124.

[4] C. Farhat, K. van der Zee, P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, Computer Methods in Applied Mechanics and Engineering 195 (17–18) (2006) 1973–2001.

[5] K. Willcox, J. Paduano, J. Peraire, Low Order Aerodynamic Models For Aeroelastic Control Of Turbomachines, in: 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, St Louis, MO, USA, 1–11, 1999.

[6] J.-F. Gerbeau, M. Vidrascu, P. Frey, Fluid-structure interaction in blood flows on geometries based on medical imaging, Computers & Structures 83 (2–3) (2005) 155–165.

[7] W. Wall, T. Rabczuk, Fluid-structure interaction in lower airways of CT-based lung geometries, International Journal for Numerical Methods in Fluids 57 (5) (2008) 653–675.

[8] C. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220–252.

[9] C. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[10] K. Dumont, J. Vierendeels, R. Kaminsky, G. Van Nooten, P. Verdonck, D. Bluestein, Comparison of the hemodynamic and thrombogenic performance of two bileaflet mechanical heart valves using a CFD/FSI model, Journal of Biomechanical Engineering - Transactions of the ASME 129 (4) (2007) 558–565.

[11] K.-J. Bathe, H. Zhang, M. Wang, Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions, Computers & Structures 56 (2–3) (1995) 193–213.

[12] K.-J. Bathe, H. Zhang, Finite element developments for general fluid flows with structural interactions, International Journal for Numerical Methods in Engineering 60 (1) (2004) 213–232.

[13] M. Heil, An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems, Computer Methods in Applied Mechanics and Engineering 193 (1–2) (2004) 1–23.

[14] J. Hron, S. Turek, A monolithic FEM/Multigrid Solver for ALE formulation of fluid structure interaction with application in biomechanics, in: H.-J. Bungartz, M. Schäfer (Eds.), Fluid-Structure Interaction – Modelling, Simulation, Optimisation, no. 53 in Lecture Notes in Computational Science and Engineering, Springer, Berlin, 146–170, 2006.

[15] S. Piperno, C. Farhat, B. Larrouturou, Partitioned procedures for the transient solution of coupled aeroelastic problems. Part I: model problem, theory and two-dimensional application, Computer Methods in Applied Mechanics and Engineering 124 (1–2) (1995) 79–112.

[16] M. Lesoinne, C. Farhat, A higher-order subiteration free staggered algorithm for non-linear transient aeroelastic problems, AIAA Journal 36 (9) (1998) 1754–1756.

[17] S. Piperno, C. Farhat, Partitioned procedures for the transient solution of coupled aeroelastic problems. Part II: energy transfer analysis and three-dimensional applications, Computer Methods in Applied Mechanics and Engineering 190 (24–25) (2001) 3147–3170.

[18] A. van Zuijlen, A. de Boer, H. Bijl, Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations, Journal of Computational Physics 224 (1) (2007) 414–430.

[19] E. van Brummelen, Added mass effects of compressible and incompressible flows in fluid-structure interaction, Journal of Applied Mechanics 76 (2) (2009) 021206–1–7.

[20] P. Causin, J.-F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid-structure problems, Computer Methods in Applied Mechanics and Engineering 194 (42–44) (2005) 4506–4527.

[21] C. Förster, W. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Computer Methods in Applied Mechanics and Engineering 196 (7) (2007) 1278–1293.

[22] D. Mok, W. Wall, E. Ramm, Accelerated iterative substructuring schemes for instationary fluid-structure interaction, in: K.-J. Bathe (Ed.), Computational Fluid and Solid Mechanics, Elsevier, 1325–1328, 2001.

[23] R. Wüchner, A. Kupzok, K.-U. Bletzinger, A framework for stabilized partitioned analysis of thin membrane-wind interaction, International Journal for Numerical Methods in Fluids 54 (6–8) (2007) 945–963.

[24] U. Küttler, W. Wall, Fixed-point fluid-structure interaction solvers with dynamic relaxation, Computational Mechanics 43 (1) (2008) 61–72.

[25] C. Michler, E. van Brummelen, R. de Borst, An interface Newton-Krylov solver for fluid-structure interaction, International Journal for Numerical Methods in Fluids 47 (10-11) (2005) 1189–1195.

[26] J. Vierendeels, L. Lanoye, J. Degroote, P. Verdonck, Implicit coupling of partitioned fluid-structure interaction problems with reduced order models, Computers & Structures 85 (11–14) (2007) 970–976.

[27] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction, Computers & Structures 87 (11–12) (2009) 793–801.

[28] E. van Brummelen, Partitioned iterative solution methods for fluid-structure interaction, International Journal for Numerical Methods in Fluids 65 (1–3) (2011) 3–27.

[29] A. Sidi, Efficient implementation of minimal polynomial and reduced rank extrapolation methods, Journal of Computational and Applied Mathematics 36 (3) (1991) 305–337.

[30] M. Mešina, Convergence acceleration for the iterative solution of the equations X = AX + f, Computer Methods in Applied Mechanics and Engineering 10 (2) (1977) 165–173.

[31] U. Küttler, W. Wall, Vector extrapolation for strong coupling fluid-structure interaction solvers, Journal of Applied Mechanics 76 (2) (2009) 021205–1–7.

[32] J. Degroote, Development of Algorithms for the Partitioned Simulation of Strongly Coupled Fluid-Structure Interaction Problems, Ph. D. thesis, Ghent University, Department of Flow, Heat and Combustion Mechanics, 2010.

[33] S. Badia, F. Nobile, C. Vergara, Fluid-structure partitioned procedures based on Robin transmission conditions, Journal of Computational Physics 227 (14) (2008) 7027–7051.

[34] J. Degroote, P. Bruggeman, R. Haelterman, J. Vierendeels, Stability of a coupling technique for partitioned solvers in FSI applications, Computers & Structures 86 (23–24) (2008) 2224–2234.

[35] J. Degroote, S. Annerel, J. Vierendeels, Stability analysis of Gauss-Seidel iterations in a partitioned simulation of fluid-structure interaction, Computers & Structures 88 (5–6) (2010) 263–271.

[36] A. Brandt, Multilevel adaptive solutions to boundary-value problems, Mathematics of Computation 31 (138) (1977) 333–390.

[37] E. van Brummelen, K. van der Zee, R. de Borst, Space/time multigrid for a fluid-structure-interaction problem, Applied Numerical Mathematics 58 (12) (2008) 1951–1971.

[38] M. Gee, U. Küttler, W. Wall, Truly monolithic algebraic multigrid for fluid-structure interaction, International Journal for Numerical Methods in Engineering 85 (8) (2011) 987–1016.

[39] A. de Boer, A. van Zuijlen, H. Bijl, Review of coupling methods for non-matching meshes, Computer Methods in Applied Mechanics and Engineering 196 (8) (2007) 1515–1525.

[40] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in Computational Mathematics 4 (1) (1995) 389–396.

[41] A. Beckert, H. Wendland, Multivariate interpolation for fluid-structure-interaction problems using radial basis functions, Aerospace Science and Technology 5 (2) (2001) 125–134.

[42] G. Golub, C. Van Loan, Matrix computations, Johns Hopkins University Press, Baltimore, MD, USA, 3$^{\text{rd}}$ edn., 1996.

[43] C. Michler, E. van Brummelen, R. de Borst, Error-amplification analysis of subiteration-preconditioned GMRES for fluid-structure interaction, Computer Methods in Applied Mechanics and Engineering 195 (17–18) (2006) 2124–2148.

[44] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, J. Vierendeels, Performance of partitioned procedures in fluid-structure interaction, Computers & Structures 88 (7–8) (2010) 446–457.

[45] J. Vierendeels, K. Riemslagh, E. Dick, A multigrid semi-implicit line-method for viscous incompressible and low-Mach-number flows on high aspect ratio grids, Journal of Computational Physics 154 (2) (1999) 310–341.

[46] M. Fernandez, M. Moubachir, A Newton method using exact Jacobians for solving fluid-structure coupling, Computers & Structures 83 (2–3) (2005) 127–142.

[47] L. Formaggia, J.-F. Gerbeau, F. Nobile, A. Quarteroni, On the coupling of 3D and 1D Navier-Stokes equations for flow problems in compliant vessels, Computer Methods in Applied Mechanics and Engineering 191 (6–7) (2001) 561–582.

[48] J.-F. Gerbeau, M. Vidrascu, A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows, ESAIM: Mathematical Modelling and Numerical Analysis 37 (4) (2003) 631–648.

[49] S. Eisenstat, H. Elman, M. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM Journal on Numerical Analysis 20 (2) (1983) 345–357.

[50] Y. Saad, M. Schultz, GMRES: A Generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific & Statistical Computing 7 (3) (1986) 856–869.

Figures

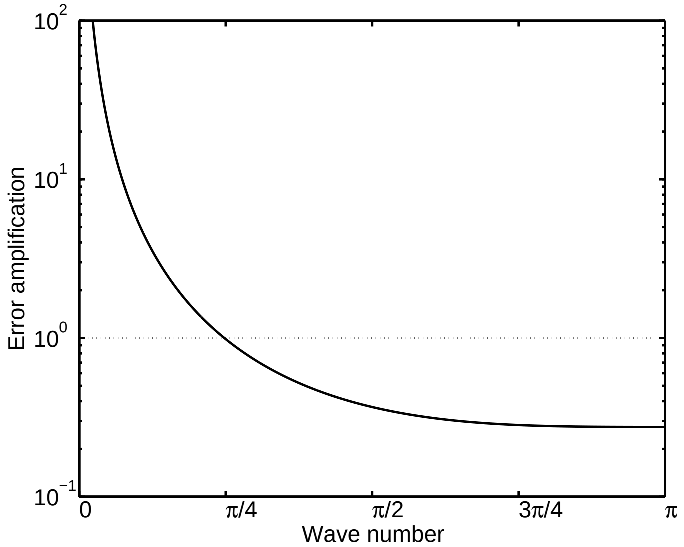

Figure 1: The amplification of the different wave numbers in the error during Gauss-Seidel coupling iterations for the calculation of the flow in a 1D flexible tube with a Dirichlet-Neumann decomposition (see Section 5.1 for the model description). This error is defined as the difference between the correct interface displacement and the interface displacement during the coupling iteration. Only a fraction of the wave numbers is unstable (i.e. amplification larger than one) and the low wave numbers are most unstable [34].

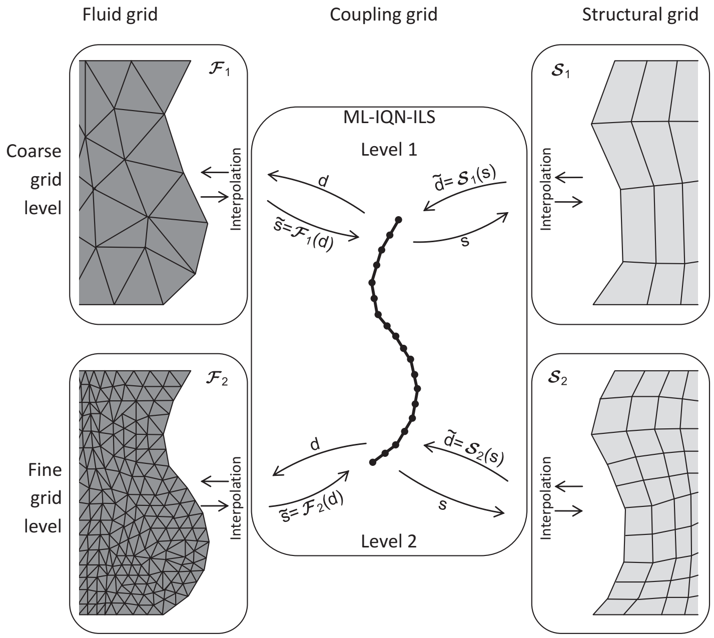

Figure 2: The coarse and fine fluid grid (left) and the coarse and fine structural grid (right), together with the unique coupling grid (centre) in a multi-level simulation with two grid levels. $\boldsymbol{d}$ represents the displacement of the interface while $\boldsymbol{s}$ represents the traction distribution on the interface. $\boldsymbol{\mathcal{F}}$ denotes the flow solver and $\boldsymbol{\mathcal{S}}$ the structural solver. The output of a solver is indicated with a tilde as this value is not always directly given as input to the other solver. In the multi-level IQN-ILS algorithm, coupling iterations are first performed on the coarse grid level (level 1) to construct the approximation for the inverse of the Jacobian as present in IQN-ILS at a lower cost. Subsequently, this approximation for the inverse of the Jacobian is used and improved further during the coupling iterations on the fine grid level (level 2), resulting in fewer coupling iterations on the fine grid level. Combined with the low cost of the coupling iterations on the coarse grid, this yields a reduction of the total time required to solve the coupled problem. All interface data on the different grid levels are interpolated to and from the unique coupling grid, which determines the dimension of the approximation for the inverse of the Jacobian.

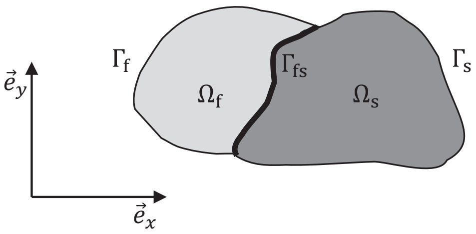

Figure 3: The abstract representation of the fluid subdomain Ω_f, the structure subdomain Ω_s, their boundaries Γ_f and Γ_s and the fluid-structure interface Γ_{fs}.

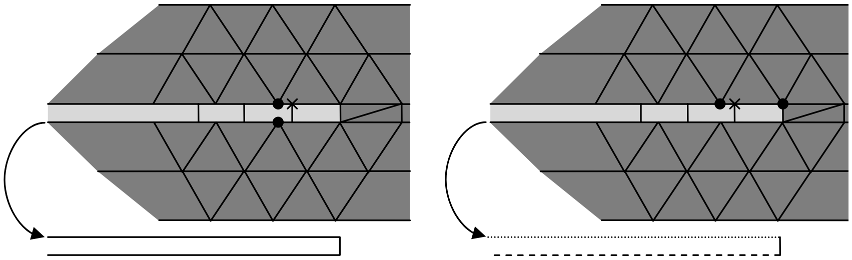

Figure 4: The selection of the two points of the dark grey grid (●) nearest to a point of the light grey grid (×), without (left) and with (right) division of the interface into surfaces. The three different surfaces are indicated with different line styles at the bottom of the figure.

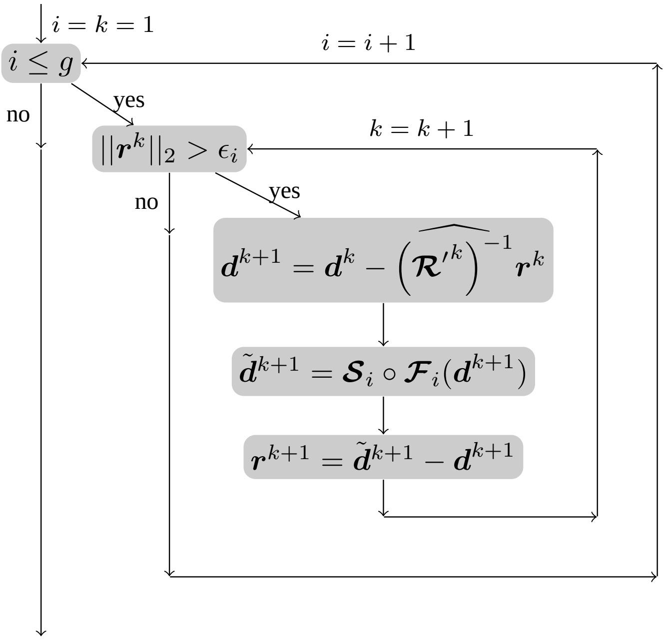

Figure 5: The simplified flowchart of one time step using ML-IQN-ILS, without extrapolation or relaxed Gauss-Seidel iteration.

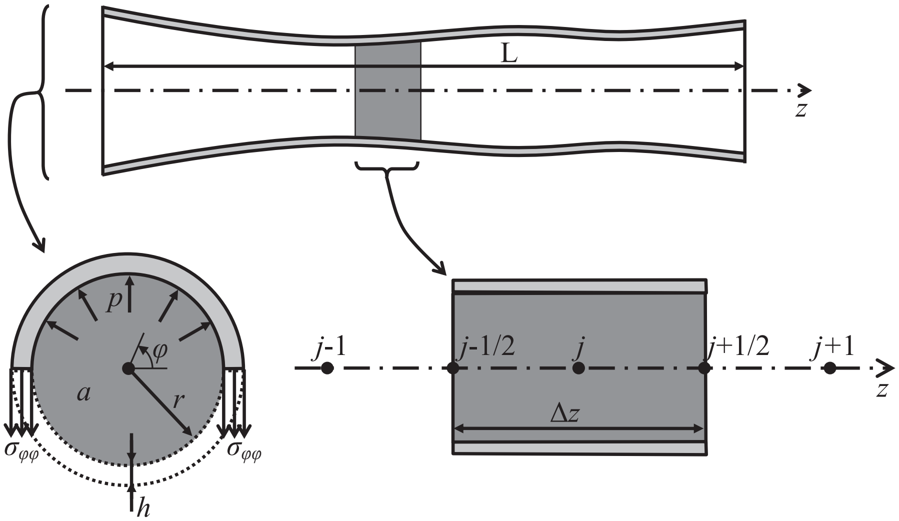

Figure 6: The 1D model for the unsteady flow in a flexible tube, with details of the cross-section and a control volume used in the discretization of the governing equations.

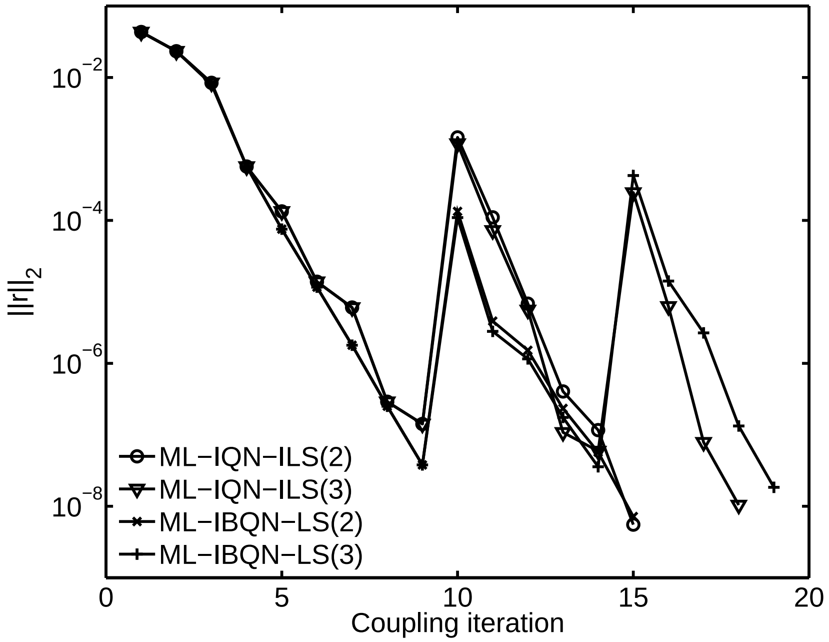

Figure 7: The convergence of the coupling iterations in the first time step for the unsteady flow in a 1D tube. When the coupling algorithm proceeds to the following, finer grid level, the residual increases but it is still smaller than the initial residual on the previous grid level. Moreover, the convergence during the first few coupling iterations on the second (and third) grid level is faster than during the first few iterations on the first grid level.

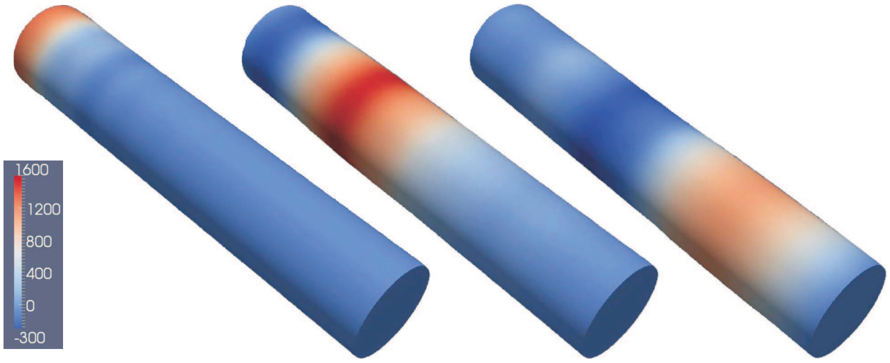

Figure 8: The pressure contours (in Pa) on the fluid-structure interface of the finest grid level for the propagation of a pressure wave in a 3D tube after 10^{-3} s (left), $5{\times}10^{-3}$ s (centre) and $9{\times}10^{-3}$ s (right).

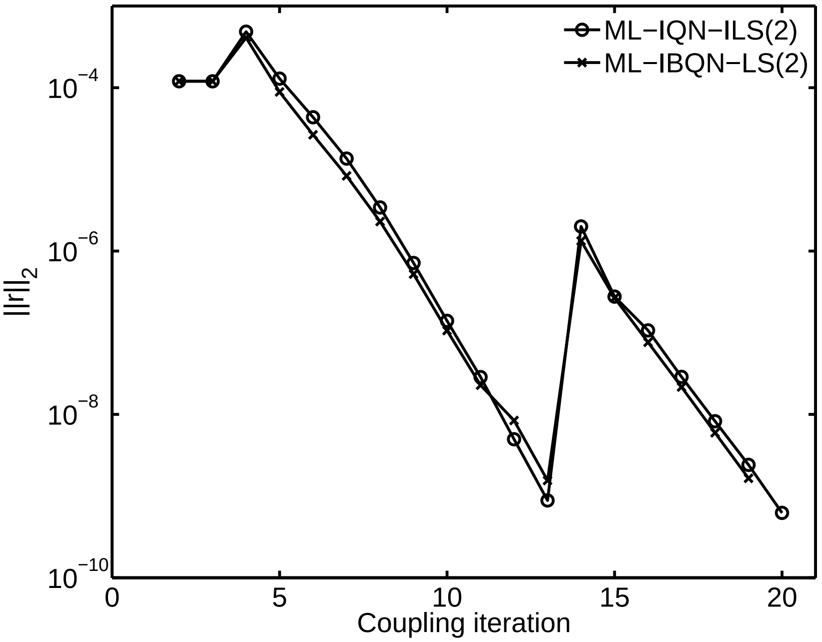

Figure 9: The convergence of the coupling iterations in the first time step for the propagation of a pressure wave in a 3D tube. When the coupling algorithm proceeds from the coarse to the fine grid level, the residual increases but it is still smaller than the initial residual. Moreover, the convergence during the first few coupling iterations on the fine grid level is faster than during the first few coupling iterations on the coarse grid level.

Tables

Algorithm	Grid points			Iterations			Duration
IQN-ILS			10^4			9.2	1.6
ML-IQN-ILS(2)	10^3		10^4	9.2		5.2	1.0
ML-IQN-ILS(2)		4×10^3	10^4		9.2	4.2	1.2
ML-IQN-ILS(3)	10^3	4×10^3	10^4	9.2	5.1	3.9	1.0
IBQN-LS			10^4			8.6	1.6
ML-IBQN-LS(2)	10^3		10^4	8.6		4.5	1.2
ML-IBQN-LS(2)		4×10^3	10^4		8.6	3.6	1.3
ML-IBQN-LS(3)	10^3	4×10^3	10^4	8.6	4.3	3.8	1.1

Table 1: The influence of the number of grid levels for the flow in a 1D flexible tube. The g in the notations ML-IQN-ILS(g) and ML-IBQN-LS(g) denotes the number of grid levels. The convergence tolerance is $\varepsilon_i = 10^{-5}||\boldsymbol{r}^0||_2$ for all grid levels. The number of coupling iterations per time step has been averaged over the entire simulation.

Algorithm	Iterations		Duration
ML-IQN-ILS(2)	9.2	5.2	1.0
ML-IQN-ILS(2)	8	5.4	1.0
ML-IQN-ILS(2)	7	5.7	1.0
ML-IQN-ILS(2)	6	6.4	1.1
ML-IBQN-LS(2)	8.6	4.5	1.2
ML-IBQN-LS(2)	8	4.5	1.1
ML-IBQN-LS(2)	7	4.8	1.1
ML-IBQN-LS(2)	6	4.9	1.1

Table 2: The influence of the number of coarse grid iterations for the flow in a 1D flexible tube. The g in the notations ML-IQN-ILS(g) and ML-IBQN-LS(g) denotes the number of grid levels. The number of grid points is 10^3 on the first grid level and 10^4 on the second grid level. The number of coarse grid iterations is limited to respectively 8, 7 and 6 in the second, third and fourth simulation with each algorithm. The number of coupling iterations per time step has been averaged over the entire simulation.

Algorithm	Iterations		Duration
IQN-ILS		13.2	1.9
ML-IQN-ILS(2)	12.1	7.0	1.0
IBQN-LS		13.3	2.0
ML-IBQN-LS(2)	12.5	6.6	1.0

Table 3: The comparison between one and two grid levels for the propagation of a pressure wave in a 3D tube. The g in the notations ML-IQN-ILS(g) and ML-IBQN-LS(g) denotes the number of grid levels. The number of coupling iterations per time step has been averaged over the entire simulation.