Performance of partitioned procedures in fluid-structure interaction

Joris Degroote[a,*], Robby Haelterman[b], Sebastiaan Annerel[a], Peter Bruggeman[c], Jan Vierendeels[a,*]

[a]*Department of Flow, Heat and Combustion Mechanics, Ghent University,*
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
[b]*Department of Mathematics (MWMW), Royal Military Academy,*
Renaissancelaan 30, B-1000 Brussels, Belgium
[c]*Department of Applied Physics, Ghent University,*
Jozef Plateaustraat 22, B-9000 Ghent, Belgium

Abstract

Partitioned simulations of fluid-structure interaction can be solved for the interface's position with Newton-Raphson iterations but obtaining the exact Jacobian is impossible if the solvers are "black boxes". It is demonstrated that only an approximate Jacobian is needed, as long as it describes the reaction to certain components of the error on the interface's position. Based on this insight, a quasi-Newton coupling algorithm with an approximation for the inverse of the Jacobian (IQN-ILS) has been developed and compared with a monolithic solver in previous work. Here, IQN-ILS is compared with other partitioned schemes such as IBQN-LS, Aitken relaxation and Interface-GMRES(R).

Key words: fluid-structure interaction, partitioned, interface, quasi-Newton, least-squares model, Aitken relaxation, Interface-GMRES(R)

1. Introduction

Recent research on fluid-structure interaction (FSI) has yielded both more complex applications and algorithmic improvements. Interesting applications of

[*] Corresponding author

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

URL: `http://www.FSI.UGent.be/` (Joris Degroote)

FSI can be found in aeronautics [1, 2] where FSI is used to assess flutter. Other challenging problems are the dynamic analysis of parachutes [3] due to the thin, highly flexible structure and the interaction between a structure and free-surface flow [4, 5]. Also of great interest are biomedical applications like blood flow in arteries [6, 7, 8] and in artificial heart valves [9, 10, 11].

An FSI problem consists of a fluid domain and a structural domain with in between the fluid-structure interface. This multi-physics problem with adjacent domains can be simulated in a monolithic or in a partitioned way. The former signifies that both the flow equations and structural equations are solved simultaneously [12, 13] while the latter means that they are solved separately. The monolithic approach requires a code developed for this particular combination of physical problems whereas the partitioned approach preserves software modularity. Moreover, the partitioned approach allows to solve the flow equations and the structural equations with different, possibly more efficient techniques which have been developed specifically for either flow equations or structural equations. On the other hand, a coupling algorithm is required to take into account the interaction between flow and motion of the structure in partitioned simulations.

Both Newton-Raphson methods and fixed-point methods can be used to solve FSI problems and other multi-physics problems with adjacent, non-overlapping domains like soil-structure interaction [14, 15], but also the particle finite element method [16], smooth particle hydrodynamics [17, 5], immersed boundary methods [18] and fictitious domain methods [19] have been used to solve FSI problems. The prefix *interface* is added to the classification if the method only operates on variables related to the fluid-structure interface and the prefix *block* before Newton-Raphson denotes that the Jacobian consists of distinct blocks. In the following paragraphs, an overview of some existing methods is given.

Block Newton-Raphson methods can be used in both the monolithic [12, 13, 20] and the partitioned [21, 22] approach. These methods solve the nonlinear flow equations and the structural equations for the variables in the entire fluid and solid domain with the Newton-Raphson method. As the flow problem is solved on a moving grid, the Jacobian of the flow equations with respect to the interface's position, the so-called shape derivative, is difficult to calculate. However, it is possible to calculate the complete Jacobian exactly [23]. The linear system within the Newton-Raphson iteration can also be solved without knowledge of the Jacobian with a matrix-free Krylov solver, using a finite-difference approximation of the Jacobian-vector product [20, 21].

Whereas block Newton-Raphson methods solve the flow and structural problem for the state in the entire fluid and solid domain, Fernandez and Moubachir

[24] rewrote the fluid-structure problem as a nonlinear problem in the state of the structure, with the flow state as internal variables of the problem. This system is subsequently solved with the Newton-Raphson method using the exact Jacobian.

It is also possible to reformulate an FSI problem as a system with only the degrees-of-freedom in the interface's position as unknowns and with all remaining variables in the fluid and solid domain as internal variables. This domain decomposition condenses the error of the FSI problem into a subspace related to the interface [25]. The FSI problem can hence be written as either a root-finding problem or a fixed-point problem, with the interface's position as unknowns.

Interface Newton-Raphson methods solve this root-finding problem with Newton-Raphson iterations, e.g. with an approximation of the Jacobian from a linear reduced-physics model [26, 27]. The linear system within the Newton-Raphson iteration can also be solved with a matrix-free Krylov solver, for example the generalized minimal residual method (GMRES), using an approximation of the Jacobian-vector product based on finite-differences or with a linear combination of the previous residual vectors in Interface-GMRES(R) [25, 28, 29].

The fixed-point problem can be solved with fixed-point iterations, also called (block) Gauss-Seidel iterations [22], which means that the flow problem and structural problem are solved successively until the change is smaller than the convergence criterion. However, the iterations converge slowly if at all, especially when the interaction between the fluid and the structure is strong due to a high fluid/structure density ratio or the incompressibility of the fluid [7]. The convergence of the fixed-point iterations can be stabilized and accelerated by Aitken relaxation and steepest descent relaxation, which adapt the relaxation factor in every iteration based on the previous iterations [30]. If the interaction between the fluid and the structure is weak, only one fixed-point iteration is required within each time step [1, 31, 32, 33, 34, 35]. These so-called staggered or loosely coupled methods do not enforce the equilibrium on the fluid-structure interface within one time step but they are suitable for aeroelastic simulations with a heavy and rather stiff structure.

Vierendeels et al. [8] rewrite the FSI problem as a system of equations with both the interface's position and the stress distribution on the interface as unknowns, and this system is solved with block quasi-Newton iterations of the Gauss-Seidel type. The Jacobians of the flow solver and structural solver are approximated by means of least-squares models, constructed with the position of the fluid-structure interface and the stress distribution on the interface in all previous quasi-Newton iterations within one time step [8, 36]. This method will be referred to as IBQN-LS, meaning interface block quasi-Newton with an approximation for the Jacobians from least-squares models.

In this work, the focus lies on partitioned methods and more specifically on methods that couple a black-box flow solver and structural solver. When the solvers are black boxes, it is difficult or even impossible to obtain the Jacobian matrices which are required in Newton-Raphson methods. However, a recent stability analysis [37, 38] on the unsteady, incompressible and inviscid flow in a straight elastic tube has demonstrated that only certain components of the error on the interface's position become unstable or are badly damped during Gauss-Seidel iterations between the flow solver and the structural solver. As will be explained in this work, this means that if quasi-Newton iterations are used, the approximate Jacobian only has to describe the reaction to those unstable or badly damped components; other components of the error will be damped anyhow during the coupling iterations. The complete Jacobian is thus not required for fast convergence of the coupled problem; an approximation of the Jacobian can be used instead.

Based on this insight, a quasi-Newton coupling algorithm with an approximation for the inverse of the Jacobian has been developed [39]. Because the *inverse* of the Jacobian is approximated, one avoids that a linear system with as dimension the number of degrees-of-freedom in the interface's position has to be solved in every quasi-Newton iteration. The approximation is constructed with the least-squares technique developed for nonlinear systems by Vierendeels et al. [8] and applied to linear systems by Haelterman et al. [40]. A matrix-free implementation of the least-squares technique is described in this work. The term "matrix-free" denotes that the approximate Jacobian is not constructed explicitly but that a procedure to calculate the product of the Jacobian with a vector is given such that less memory and CPU time is required. Moreover, information from previous time steps can be reused in the least-squares approximation and the results demonstrate that this extension greatly accelerates the convergence of the quasi-Newton iterations. The quasi-Newton algorithm presented in [39] is called IQN-ILS because it calculates the interface position using quasi-Newton iterations with an approximation for the inverse of the Jacobian from a least-squares model.

Other techniques that are able to couple black-box solvers are Aitken relaxation, Interface-GMRES and IBQN-LS, all of which have been introduced above. By comparing the algorithms of these techniques with the IQN-ILS algorithm, it is shown that the IQN-ILS method can easily be implemented in an FSI framework that currently uses (Aitken) relaxation or Interface-GMRES(R) and that IQN-ILS is more straightforward to implement than IBQN-LS.

In the results section, the performance of all these techniques is compared in two cases, both in terms of how often the flow problem and structural problem have to be solved within one time step and in terms of the total simulation time. The first case is the steady and unsteady two-dimensional (2D) simulation of a flexible beam behind a cylinder in a laminar flow, which is validated with a well-known benchmark [41]. The second case is the propagation of a pressure pulse in a three-dimensional (3D) flexible tube. These simulations indicate that the IQN-ILS is faster than Aitken relaxation and Interface-GMRES(R) and that the performance of IQN-ILS and IBQN-LS is similar.

In previous work [39], the performance of the IQN-ILS algorithm was compared with a monolithic Newton solver in five different cases considering various structural configurations and incompressible fluids. For each case when convergence was reached, the ratio of the time for the IQN-ILS simulation to the time for the monolithic Newton simulation was between 1/2 and 4, but there was a case for which the partitioned simulation did not converge. While problems of various characteristics were solved, still, only specific problems were considered and in general rather small problems in number of equations. Moreover, the solutions of the structural equations and the flow equations were calculated using a direct sparse solver with full Newton-Raphson iterations although different solver schemes, in particular much more efficient for the fluid equations when the number of elements becomes very large [42], are frequently used in the partitioned approach. The performance comparisons may consequently look different when different problems are solved and other solver schemes are used. However, the general observations given in [39] can be used to assess whether a monolithic or partitioned solution of a fluid-structure interaction problem might be more effective.

The remainder of this paper is organized as follows. After providing the necessary definitions in Section 2, Section 3 explains why an approximation for the Jacobian can be used. Section 4 describes the IQN-ILS technique and the matrix-free implementation of the least-squares approximation with reuse of information from previous time steps. In Sections 5, 6 and 7, IQN-ILS is compared with respectively IBQN-LS, Aitken relaxation and Interface-GMRES(R). The performance analysis of the coupling schemes is subsequently given in Section 8, followed by the conclusion in Section 9.

2. Definitions

This section gives a definition of the functions that represent the flow solver and structural solver, as they are used by interface Newton-Raphson and fixed-point methods. The governing equations in the fluid and solid domain, the discretizations and the solution techniques of the discrete equations are not described here because they are not important for the comparison between the coupling algorithms but they can be found in [39] or any book on computational fluid and solid mechanics.

Often, a velocity (or position after time integration) is imposed on the fluid side of the interface and a stress distribution is applied on the structural side, which is a Dirichlet-Neumann decomposition of the FSI problem [7]. The following abstract definitions emphasize that the solvers are treated as black boxes.

The function

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}) \tag{1}$$

is referred to as the flow solver and it concisely represents several operations. The discretized position $\boldsymbol{x} \in \mathbb{R}^u$ of the fluid-structure interface is given to the flow code, and the grid of the fluid domain adjacent to the interface is adapted accordingly. Subsequently, the grid velocity is calculated and the flow equations are solved for the fluid state in the entire fluid domain, which also results in a stress distribution $\boldsymbol{y} \in \mathbb{R}^w$ on the interface. In an unsteady simulation, both $\boldsymbol{x}$ and $\boldsymbol{y}$ are at the new time level and the boundary conditions and other settings of $\mathcal{F}$ are also adapted to the new time level.

The structural solver is represented by the function

$$\boldsymbol{x} = \mathcal{S}(\boldsymbol{y}). \tag{2}$$

This expression indicates that the fluid stress distribution on the interface is given to the structural code which then calculates the position of the entire structure and thus also the new position of the fluid-structure interface. With these definitions, the FSI problem is given by

$$\boldsymbol{x} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}) \quad \text{or} \quad \mathcal{R}(\boldsymbol{x}) = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}) - \boldsymbol{x} = 0, \tag{3}$$

in fixed-point or root-finding formulation, respectively, with $\mathcal{R}$ being the residual operator.

If the flow problem and the structural problem are not discretized in the same way on the fluid-structure interface, there has to be a mapping between the solvers. In this paper, it is assumed that this mapping is included in one of the solvers if necessary. An overview of mapping methods can be found in [43]. Radial basis functions can be used for both interpolation on the fluid-structure interface [44] and mesh motion in the fluid domain [45].

Although the solvers are used as black boxes, it is important to keep in mind that differences between the numerical methods of both codes can cause undesirable effects. When different time integration schemes are used by both solvers, it is possible that spurious oscillations in time are present in the acceleration of the fluid and the structure and thus also in the stresses on the interface. This has been demonstrated both analytically and numerically in [46].

In the remainder of this paper, all values and functions are at the new time level $n+1$, unless indicated otherwise with a left superscript. A right superscript indicates the coupling iteration while a subscript denotes the element in a vector. Capital letters denote matrices, bold lower case letters indicate vectors and lower case letters represent scalars. Approximations are indicated with a hat. The output of the solvers $\mathcal{F}$ and $\mathcal{S}$ is indicated with a tilde because this is only an intermediate value that is not passed on to the next coupling iteration. This tilde is dropped once the final value that will be used in the next iteration has been calculated. Depending on the context, the equality sign can denote either assignment or equality.

All coupling algorithms begin the next time step from an extrapolation of the interface's position

$$\boldsymbol{x}^0 = \frac{5}{2}({}^{n}\boldsymbol{x}) - 2({}^{n-1}\boldsymbol{x}) + \frac{1}{2}({}^{n-2}\boldsymbol{x}), \tag{4}$$

based on the previous time steps. Lower order extrapolations are used for the first two time steps.

3. Motivation for the use of quasi-Newton methods

Degroote et al. [37] have presented a stability analysis of the Gauss-Seidel iterations between the flow solver and the structural solver in a partitioned simulation of the one-dimensional (1D) unsteady flow in a straight, flexible tube. In this section, only the conclusion of this analysis is offered as a motivation for the use of quasi-Newton methods. To simplify the equations, the mass of the structure is neglected and the fluid is incompressible and inviscid. The governing equations are linearized such that Fourier error analysis can be conducted and the behaviour of the different wave numbers in the error on the interface's position can be analyzed. Although this simplified model does not include all physical details, it provides insight into the stability of coupling algorithms for partitioned FSI simulations.

The stability analysis investigates the difference between the correct position of the interface and its value during the Gauss-Seidel iterations (further denoted as the error) by decomposing it as a weighted sum of cosine functions with different wave numbers. The amplification μ of the error is written as a function of the wave number and three parameters, namely the dimensionless stiffness of the tube κ, the dimensionless time step τ and the number of degrees-of-freedom in the discretization of the interface u. For more details and the exact definition of these parameters, the reader is referred to [37].

Figure 1 depicts the amplification of the error as a function of the wave number. From this figure, it can be seen that the error amplification increases if κ or τ decreases and that especially the low wave numbers are unstable. However, even for a very flexible tube and a small time step, only some wave numbers are unstable. The physical meaning of these curves is shown in Figure 2. The position of the wall of a tube, which initially has a constant cross section and contains fluid at rest, is perturbed with two different wave numbers. At the inlet and outlet, a zero pressure is imposed. Because the fluid is incompressible, a displacement of the interface with a low wave number requires that the fluid is accelerated globally, which causes large pressure variations throughout the fluid and thus also a violent motion of the structure. On the other hand, a displacement with a high wave number only generates local fluid motion and hence smaller pressure gradients and structural motion. The pressure differences in the case of the high wave number can barely be seen since the same scale has been used for both wave numbers. The material properties, geometrical dimensions and time step for this simulation are identical to those in Section 8.2. Although the curves in Figure 1 depend on the geometry and discretization, this physical interpretation indicates that the underlying idea would remain valid in other situations.

Only the components of the error which are unstable ($\mu > 1$) or which disappear slowly ($\mu \approx 1$) in Gauss-Seidel iterations between the flow solver and the structural solver, or equivalently Richardson iterations for the FSI problem, have to be removed by means of a quasi-Newton technique. Thus, for the quasi-Newton iterations to converge quickly, the approximate Jacobian only has to describe the reaction to a limited number of components in the error on the interface's position, namely the components with low wave numbers. For the components which are not included in the approximate Jacobian, the quasi-Newton iterations correspond to Richardson iterations, as will be explained below.

4. IQN-ILS

In this section, the IQN-ILS method with a matrix-free implementation of the least-squares approximation is presented. IBQN-LS, Aitken relaxation and

Interface-GMRES(R) are reviewed in the following sections using the same notation.

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

$$\mathcal{R}(\boldsymbol{x}) = \boldsymbol{0} \tag{5}$$

can be solved by means of Newton-Raphson iterations

$$\text{solve} \quad \left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}^k} \Delta\boldsymbol{x}^k = -\boldsymbol{r}^k \tag{6a}$$

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k \tag{6b}$$

with the residual calculated as

$$\boldsymbol{r}^k = \mathcal{R}(\boldsymbol{x}^k) = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^k) - \boldsymbol{x}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k. \tag{7}$$

The Newton-Raphson iterations in one time step have converged when $||\boldsymbol{r}^k||_2 \leq \varepsilon_o$ with ε_o the convergence tolerance. However, the exact Jacobian of $\mathcal{R}$ is unknown as the Jacobians of $\mathcal{F}$ and $\mathcal{S}$ are unavailable. Moreover, a linear system (6a) with as dimension the number of degrees-of-freedom in the position of the fluid-structure interface has to be solved in every Newton-Raphson iteration. Although the number of degrees-of-freedom in the interface's position is generally smaller than the number of degrees-of-freedom in the entire fluid and structure domain, the Jacobian matrix $\mathrm{d}\mathcal{R}/\mathrm{d}\boldsymbol{x}$ is usually dense. As a result, the solution of the linear system (6a) corresponds to a significant computational cost in large simulations, especially if a direct solver is used.

If the Jacobian $\mathrm{d}\mathcal{R}/\mathrm{d}\boldsymbol{x}$ is approximated and quasi-Newtons iterations are performed, black-box solvers can be used but this approach does not circumvent that the linear system (6a) has to be solved. It is more advantageous to approximate the *inverse* of the Jacobian by applying the least-squares technique introduced by Vierendeels et al. [8] on a particular set of vectors as will be explained below. The quasi-Newton iterations with the approximation for the inverse of the Jacobian can be written as

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \left(\widehat{\left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}^k}}\right)^{-1} \left(-\boldsymbol{r}^k\right). \tag{8}$$

It can be seen from equation (8) 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. $\mathbf{0}$, and the current residual $\boldsymbol{r}^k$ and it is further denoted as $\Delta\boldsymbol{r} = \mathbf{0} - \boldsymbol{r}^k = -\boldsymbol{r}^k$. In this work, the matrix-vector product is calculated from information obtained during the previous quasi-Newton iterations. Equation (7) shows that the flow equations and structural equations are solved in quasi-Newton iteration k, resulting in $\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^k)$ and the corresponding residual $\boldsymbol{r}^k$. The vectors $\tilde{\boldsymbol{x}}$ and $\boldsymbol{r}$ from all previous coupling iterations are also available, giving a set of known residual vectors

$$\boldsymbol{r}^k, \quad \boldsymbol{r}^{k-1}, \quad \ldots, \quad \boldsymbol{r}^1, \quad \boldsymbol{r}^0 \tag{9a}$$

and the corresponding set of vectors $\tilde{x}$

$$\tilde{\boldsymbol{x}}^{k+1}, \quad \tilde{\boldsymbol{x}}^k, \quad \ldots, \quad \tilde{\boldsymbol{x}}^2, \quad \tilde{\boldsymbol{x}}^1. \tag{9b}$$

The differences between all vectors from previous iterations (superscript i) and the most recent vector (superscript k) are calculated

$$\Delta\boldsymbol{r}^i = \boldsymbol{r}^i - \boldsymbol{r}^k \tag{10a}$$

$$\Delta\tilde{\boldsymbol{x}}^{i+1} = \tilde{\boldsymbol{x}}^{i+1} - \tilde{\boldsymbol{x}}^{k+1} \tag{10b}$$

for $i = 0, \ldots, k-1$. The final result will be the same if vectors other than $\boldsymbol{r}^k$ and $\tilde{\boldsymbol{x}}^{k+1}$ are selected as the reference: choosing the first iteration as reference would avoid that all differences have to be recalculated in every coupling iteration; however, choosing the last iteration facilitates the comparison between the IQN-ILS method and Aitken relaxation in Section 6.

Each $\Delta\boldsymbol{r}^i$ corresponds to a $\Delta\tilde{\boldsymbol{x}}^{i+1}$ and these vectors are stored as the columns of the matrices

$$^{n+1}V^k = \begin{bmatrix} \Delta\boldsymbol{r}^{k-1} & \Delta\boldsymbol{r}^{k-2} & \ldots & \Delta\boldsymbol{r}^1 & \Delta\boldsymbol{r}^0 \end{bmatrix} \tag{11a}$$

and

$$^{n+1}W^k = \begin{bmatrix} \Delta\tilde{\boldsymbol{x}}^k & \Delta\tilde{\boldsymbol{x}}^{k-1} & \ldots & \Delta\tilde{\boldsymbol{x}}^2 & \Delta\tilde{\boldsymbol{x}}^1 \end{bmatrix}. \tag{11b}$$

Due to the similarity between subsequent time steps, the information from the previous time steps can be reused. The matrices $^{n+1}V^k$ and $^{n+1}W^k$ are then combined with those from q previous time steps (if at least q time steps have already been performed), giving

$$V^k = \begin{bmatrix} ^{n+1}V^k & ^{n}V & \ldots & ^{n-q+2}V & ^{n-q+1}V \end{bmatrix} \tag{12a}$$

and

$$W^k = \begin{bmatrix} {}^{n+1}W^k & {}^{n}W & \dots & {}^{n-q+2}W & {}^{n-q+1}W \end{bmatrix}. \tag{12b}$$

The columns of the matrices ${}^{n-i+1}V$ and ${}^{n-i+1}W$ are calculated by subtracting the vector of the last iteration of time step $n-i+1$ from all previous vectors in that time step, in the same way as in equation (10). By including the information for q previous time steps, the convergence of the coupling iterations is remarkably accelerated. However, if information from too many time steps is reused, the convergence can slow down again as information from time step $n-q+1$ might no longer be relevant in time step $n+1$. The optimal value of q is problem dependent but the convergence of the coupling iterations does not change significantly near the optimum such that the performance of the method is robust with respect to the parameter q.

The number of columns in V^k and W^k is indicated with v and 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 $\Delta\boldsymbol{r}^i$

$$\Delta\boldsymbol{r} \approx V^k\boldsymbol{c}^k \tag{13}$$

with $\boldsymbol{c}^k \in \mathbb{R}^v$ the coefficients of the decomposition. Because $v \leq u$, equation (13) 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. In [8], the solution of the least-squares problem is calculated as

$$\boldsymbol{c}^k = ({V^k}^{\mathrm{T}}V^k)^{-1}{V^k}^{\mathrm{T}}\Delta\boldsymbol{r} \tag{14}$$

but this implementation becomes unstable if the number of columns in the matrix V^k is high which is especially encountered when information from previous time steps is reused. For that reason, the so-called economy size QR-decomposition of V^k is calculated using Householder transformations [47]

$$V^k = Q^kR^k \tag{15}$$

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

$$R^k\boldsymbol{c}^k = {Q^k}^{\mathrm{T}}\Delta\boldsymbol{r} \tag{16}$$

using back substitution. If a $\Delta\boldsymbol{r}^i$ vector is (almost) a linear combination of other $\Delta\boldsymbol{r}^j$ vectors, one of the diagonal elements of R^k will (almost) be zero. Consequently, the equation corresponding to that row of R^k cannot be solved during the back substitution and the corresponding element of $\boldsymbol{c}^k$ is set to zero. The threshold for the detection of small diagonal elements is set to $10^{-10} \cdot ||R^k||_2$ in this work.

The $\Delta\tilde{\boldsymbol{x}}$ that corresponds to $\Delta\boldsymbol{r}$ is subsequently calculated as a linear combination of the previous $\Delta\tilde{\boldsymbol{x}}^i$, analogous to the equation (13), giving

$$\Delta\tilde{\boldsymbol{x}} = W^k\boldsymbol{c}^k. \tag{17}$$

From equation (7), it follows that

$$\Delta\boldsymbol{r} = \Delta\tilde{\boldsymbol{x}} - \Delta\boldsymbol{x} \tag{18}$$

and substitution of equation (17) in equation (18) results in

$$\Delta\boldsymbol{x} = W^k\boldsymbol{c}^k - \Delta\boldsymbol{r}. \tag{19}$$

Because the coefficients $\boldsymbol{c}^k$ are a function of $\Delta\boldsymbol{r}$, equation (19) shows how $\Delta\boldsymbol{x}$ can be approximated for a given $\Delta\boldsymbol{r}$. Hence, equation (19) can be seen as a procedure to calculate the product of the approximation for the inverse of the Jacobian and a vector $\Delta\boldsymbol{r} = -\boldsymbol{r}^k$

$$\Delta\boldsymbol{x} = \left(\widehat{\left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}_k}}\right)^{-1}\Delta\boldsymbol{r} = W^k\boldsymbol{c}^k + \boldsymbol{r}^k. \tag{20}$$

This matrix-free procedure requires less memory (proportional to the number of rows and columns in V^k, so $u \times v$) and is also faster than the explicit creation of approximation for the inverse of the Jacobian as

$$\left(\widehat{\left.\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}_k}}\right)^{-1} = W^k {R^k}^{-1} {Q^k}^{\mathrm{T}} - I \tag{21}$$

with I the unity matrix in $\mathbb{R}^{u\times u}$. This is a significant advantage for simulations with a high number of degrees-of-freedom on the interface.

The relation between $\Delta\boldsymbol{r}$ and $\Delta\boldsymbol{x}$ is thus found by means of the $\Delta\tilde{\boldsymbol{x}}$ values. One might try to relate the residual $\boldsymbol{r}$ directly to $\boldsymbol{x}$ instead of to $\tilde{\boldsymbol{x}}$, but this obviously will not work as the new input for $\mathcal{S} \circ \mathcal{F}$ would be a linear combination of the previous inputs. The only new information in the input of $\mathcal{S} \circ \mathcal{F}$ would originate from numerical errors and the coupling iterations would not converge.

From equation (16), it can be seen that if part of $\Delta \boldsymbol{r}$ is orthogonal to V^k and thus to Q^k, the decomposition coefficients $\boldsymbol{c}^k$ will be zero for that part of $\Delta \boldsymbol{r}$. Equation (20) shows that this component of the residual is not modified such that Gauss-Seidel iterations between the flow solver and the structural solver are performed for this component of the residual which corresponds to Richardson iterations for the complete FSI problem. Only the components of $\Delta \boldsymbol{r}$ in the span of the columns of V^k are reduced by means of quasi-Newton iterations.

As for any other Newton-Raphson or quasi-Newton scheme, the IQN-ILS iterations will only converge when the initial guess is sufficiently close to the final solution. For example, if the deformation of the structure within one time step is too large and the coupling iterations do not converge, then the time step must be adjusted. The same remark can be made for the IBQN-LS and Interface-GMRES coupling techniques.

The complete IQN-ILS technique is shown in Algorithm 1 and a simplified version is shown in Figure 3(a). Because the matrices V^k and W^k have to contain at least one column, a relaxation with factor ω (line 4) is performed in the second coupling iteration of the first time step if information from the previous time steps is reused ($q > 0$) and in the second coupling iteration of every time step without reuse ($q = 0$).

Algorithm 1 IQN-ILS method

```
 1: $k = 0;\ \tilde{\boldsymbol{x}}^1 = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^0);\ \boldsymbol{r}^0 = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
 2: while $||\boldsymbol{r}^k||_2 > \varepsilon_o$ do
 3:     if $k = 0$ and ($q = 0$ or $n = 0$) then
 4:         $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega \boldsymbol{r}^k$
 5:     else
 6:         construct $V^k$ and $W^k$ as shown in equation (10) to (12)
 7:         calculate QR-decomposition $V^k = Q^k R^k$
 8:         solve $R^k \boldsymbol{c}^k = -{Q^k}^{\mathrm{T}} \boldsymbol{r}^k$
 9:         $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + W^k \boldsymbol{c}^k + \boldsymbol{r}^k$
10:     end if
11:     $k = k + 1;\ \tilde{\boldsymbol{x}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^k);\ \boldsymbol{r}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k$
12: end while
```

5. IBQN-LS

The IBQN-LS method is explained in detail in Algorithm 2 and a simplified representation is shown in Figure 3(b). This coupling technique solves the FSI problem written as

$$\mathcal{F}(\boldsymbol{x}) - \boldsymbol{y} = 0 \tag{22a}$$

$$\mathcal{S}(\boldsymbol{y}) - \boldsymbol{x} = 0 \tag{22b}$$

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

$$\begin{bmatrix} \widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}} & -I \\ -I & \widehat{\frac{\mathrm{d}\mathcal{S}}{\mathrm{d}\boldsymbol{y}}} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \end{bmatrix} = - \begin{bmatrix} \mathcal{F}(\boldsymbol{x}) - \boldsymbol{y} \\ \mathcal{S}(\boldsymbol{y}) - \boldsymbol{x} \end{bmatrix} \tag{23}$$

is thus first solved for $\Delta\boldsymbol{x}$, followed by an update of $\boldsymbol{x}$ and the right-hand side. Subsequently, the modified system is solved for $\Delta\boldsymbol{y}$ and $\boldsymbol{y}$ is updated. As a consequence, the IBQN-LS method modifies the stress distribution that is calculated by the flow solver before transferring it to the structural solver, as opposed to the other techniques described in this paper. With the notation for intermediate values defined in Section 2, the input and output of the flow solver are thus denoted as $\boldsymbol{x}^k$ and $\tilde{\boldsymbol{y}}^{k+1}$ and the input and output of the structural solver are $\boldsymbol{y}^{k+1}$ and $\tilde{\boldsymbol{x}}^{k+1}$.

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

$$\left(I - \widehat{\frac{\mathrm{d}\mathcal{S}}{\mathrm{d}\boldsymbol{y}}}^k \widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k \right) \Delta\boldsymbol{x}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k + \widehat{\frac{\mathrm{d}\mathcal{S}}{\mathrm{d}\boldsymbol{y}}}^k \left(\tilde{\boldsymbol{y}}^{k+1} - \boldsymbol{y}^{k+1} \right) \tag{24}$$

for $\Delta\boldsymbol{x}^k$. As opposed to the original approach of Vierendeels et al. [8], this linear system is solved with an iterative solver like the generalized conjugate residual method [48] or the generalized minimal residual method [49] in a matrix-free way. The matrix on the left-hand side of equation (24) and consequently the approximate Jacobians $\widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k$ and $\widehat{\frac{\mathrm{d}\mathcal{S}}{\mathrm{d}\boldsymbol{y}}}^k$ 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 $\widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k$ or $\widehat{\frac{\mathrm{d}\mathcal{S}}{\mathrm{d}\boldsymbol{y}}}^k$ with a vector is similar to the procedure described in Section 4. The matrix-vector product with $\widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k$ is calculated from the previous inputs

$$\boldsymbol{x}^0, \quad \ldots, \quad \boldsymbol{x}^k \tag{25a}$$

and the corresponding outputs

$$\tilde{\boldsymbol{y}}^1 = \mathcal{F}(\boldsymbol{x}^0), \quad \ldots, \quad \tilde{\boldsymbol{y}}^{k+1} = \mathcal{F}(\boldsymbol{x}^k). \tag{25b}$$

of the flow solver which are converted into differences with respect to the last input and output

$$\Delta\boldsymbol{x}^i = \boldsymbol{x}^i - \boldsymbol{x}^k \tag{26a}$$
$$\Delta\tilde{\boldsymbol{y}}^{i+1} = \tilde{\boldsymbol{y}}^{i+1} - \tilde{\boldsymbol{y}}^{k+1} \tag{26b}$$

for $i = 0, \ldots, k-1$. All $\Delta\boldsymbol{x}^i$ and $\Delta\tilde{\boldsymbol{y}}^{i+1}$ from the current time step (and possibly from previous time steps) are stored as columns of the matrices V_f^k and W_f^k with the subscript f referring to the flow solver. Subsequently, the economy-size QR-decomposition of V_f^k is calculated. To determine the product of $\widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k$ with a vector $\Delta\boldsymbol{x}$, the triangular system

$$R_f^k \boldsymbol{c}_f^k = {Q_f^k}^{\mathrm{T}} \Delta\boldsymbol{x} \tag{27}$$

is solved for $\boldsymbol{c}_f^k$ and the matrix-vector product is then calculated as

$$\widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k \Delta\boldsymbol{x} = W_f^k \boldsymbol{c}_f^k. \tag{28}$$

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

Once $\boldsymbol{x}^{k+1}$ has been obtained, k is increased, the corresponding stress distribution $\tilde{\boldsymbol{y}}^{k+1} = \mathcal{F}(\boldsymbol{x}^k)$ is calculated and the matrices V_f, W_f, Q_f and R_f are updated. To calculate the stress distribution $\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \Delta\boldsymbol{y}^k$ that has to be applied on the structure, the system

$$\left(I - \widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k \widehat{\frac{\mathrm{d}\mathcal{S}}{\mathrm{d}\boldsymbol{y}}}^{k-1}\right) \Delta\boldsymbol{y}^k = \tilde{\boldsymbol{y}}^{k+1} - \boldsymbol{y}^k + \widehat{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{x}}}^k \left(\tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k\right) \tag{29}$$

is solved, again with the matrix-free iterative solver. Every time the solution of either the flow problem or the structural problem is calculated, the approximate Jacobian of the corresponding solver is improved by means of the solver's latest input and output.

Analogous to the IQN-ILS technique, the matrices $V_{s,f}^k$ and $W_{s,f}^k$ 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 position (line 4 in Algorithm 2) and the stress distribution is passed on without modification (line 13).

Algorithm 2 IBQN-LS method [8]

```
$k = 0;\ \tilde{\boldsymbol{y}}^1 = \mathcal{F}(\boldsymbol{x}^0);\ \boldsymbol{y}^1 = \tilde{\boldsymbol{y}}^1;\ \tilde{\boldsymbol{x}}^1 = \mathcal{S}(\boldsymbol{y}^1);\ \boldsymbol{r}^0 = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
while $||\boldsymbol{r}^k||_2 > \varepsilon_o$ do
    if $k = 0$ and ($q = 0$ or $n = 0$) then
        $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega \boldsymbol{r}^k$
    else
        construct $V_s^k$ and $W_s^k$
        calculate QR-decomposition $V_s^k = Q_s^k R_s^k$
        solve equation (24) for $\Delta \boldsymbol{x}^k$
        $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta \boldsymbol{x}^k$
    end if
    $k = k + 1;\ \tilde{\boldsymbol{y}}^{k+1} = \mathcal{F}(\boldsymbol{x}^k)$
    if $k = 1$ and ($q = 0$ or $n = 0$) then
        $\boldsymbol{y}^{k+1} = \tilde{\boldsymbol{y}}^{k+1}$
    else
        construct $V_f^k$ and $W_f^k$
        calculate QR-decomposition $V_f^k = Q_f^k R_f^k$
        solve equation (29) for $\Delta \boldsymbol{y}^k$
        $\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \Delta \boldsymbol{y}^k$
    end if
    $\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S}(\boldsymbol{y}^{k+1});\ \boldsymbol{r}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k$
end while
```

6. Aitken relaxation

Aitken relaxation [30], shown in Algorithm 3 and Figure 3(c), determines a dynamically varying scalar relaxation factor ω^k for the fixed-point iterations within a time step.

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega^k \boldsymbol{r}^k \tag{30a}$$

$$= (1 - \omega^k)\boldsymbol{x}^k + \omega^k \tilde{\boldsymbol{x}}^{k+1} \tag{30b}$$

The next input for $\mathcal{S} \circ \mathcal{F}$ is thus a linear combination of the last output and the previous input. Moreover, the update of the interface's position is in the direction of the residual vector, as opposed to the update from the IQN-ILS method on line 9 of Algorithm 1. The first relaxation in a time step is executed with the relaxation factor from the end of the previous time step, but limited to ω^{max}, so

$\omega^0 = \operatorname{sign}({}^n\omega)\min(|{}^n\omega|, \omega^{max})$ and ${}^0\omega^0 = \omega^{max}$. The value of ω^k is obtained as

$$\omega^k = -\omega^{k-1}\frac{(\boldsymbol{r}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})}{(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})} \tag{31}$$

which is interpreted in [30] as the secant method for scalars directly applied to vectors and projected on $\boldsymbol{r}^k - \boldsymbol{r}^{k-1}$. By combining equations (30) and (31), it can be seen that the update of the interface's position is given by

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \frac{(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})}{(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})}(-\boldsymbol{r}^k) \tag{32a}$$

$$= \boldsymbol{x}^k + \left[\frac{(\tilde{\boldsymbol{x}}^{k+1} - \tilde{\boldsymbol{x}}^k)^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})}{(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})} - 1\right](-\boldsymbol{r}^k) \tag{32b}$$

for $k > 0$. The previous equation is similar to line 9 of Algorithm 1. If the Jacobian were created explicitly in the IQN-ILS algorithm, if the normal equations were used to solve the least-squares problem in equation (13) and if the matrices V^k and W^k were limited to their last column, line 9 of Algorithm 1 would give

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \left[\frac{(\tilde{\boldsymbol{x}}^{k+1} - \tilde{\boldsymbol{x}}^k)(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^{\mathrm{T}}}{(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})} - I\right](-\boldsymbol{r}^k). \tag{33}$$

Equations (32) and (33) are, however, not identical because the coefficient of $-\boldsymbol{r}^k$ is a scalar in the first equation and a matrix in the second one. This proves that Aitken relaxation is fundamentally different from the IQN-ILS method. On the other hand, it demonstrates that implementing the IQN-ILS method is hardly more complex than implementing Aitken relaxation. However, Aitken relaxation can also be seen as an interface quasi-Newton technique: if the inverse of the Jacobian in equation (8) is approximated by $-\omega^k I$, the Aitken relaxation method is retrieved.

Section 8 will demonstrate the gain in performance obtained by using the information from all previous iterations or even previous time steps in the approximate Jacobians instead of only from the last two iterations in Aitken relaxation. Steepest descent relaxation [30] is similar to Aitken relaxation, but it is not studied in this paper as it exhibits stepwise or zig-zag convergence which is typical for steepest descent methods and well understood [50].

7. Interface-GMRES(R)

Interface-GMRES (Algorithm 4 and Figure 3(d)) uses the Newton-Raphson method to solve the nonlinear equation $\mathcal{R}(\boldsymbol{x}) = 0$ for the interface's position. It

Algorithm 3 Aitken relaxation [30]

```
$k = 0$; $\tilde{\boldsymbol{x}}^1 = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^0)$; $\boldsymbol{r}^0 = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
while $||\boldsymbol{r}^k||_2 > \varepsilon_o$ do
    if $k = 0$ then
        $\omega^0 = \text{sign}(^n\omega) \min(|^n\omega|, \omega^{max})$
    else
        $\omega^k = -\omega^{k-1} \frac{(\boldsymbol{r}^{k-1})^{\mathrm{T}}(\boldsymbol{r}^k - \boldsymbol{r}^{k-1})}{||\boldsymbol{r}^k - \boldsymbol{r}^{k-1}||_2}$
    end if
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega^k \boldsymbol{r}^k$
    $k = k + 1$; $\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^k)$; $\boldsymbol{r}^k = \tilde{\boldsymbol{x}}^{k+1} - \boldsymbol{x}^k$
end while
```

is a hybrid Newton-Krylov method [51], applied to FSI. The Newton-Raphson update $\Delta\boldsymbol{x}$ is obtained by solving the linear system in equation (6a). To be able to calculate the Jacobian-vector product, fixed-point iterations are first performed in line 4 to 19 which results in a sequence of interface positions $\boldsymbol{x}^i$ and the corresponding residuals $\boldsymbol{r}^i$ with $i = 1, \ldots, j$. From these sequences, differences $\Delta\boldsymbol{x}^i = \boldsymbol{x}^i - \boldsymbol{x}^0$ and $\Delta\boldsymbol{r}^i = \boldsymbol{r}^i - \boldsymbol{r}^0$ are calculated and stored as columns of V^j and W^j. It has been proved by induction that the Krylov space corresponding with the linear system in equation (6a) is asymptotically similar to $\text{span}\{\Delta\boldsymbol{x}^i\}_{i=1}^{j}$ and the fixed-point iterations thus serve as a preconditioner to the GMRES solution of equation (6a) [25, 28]. The residual at $\boldsymbol{x} + \sum_{i=1}^{j} c_i \Delta\boldsymbol{x}^i$ is approximated as

$$\mathcal{R}(\boldsymbol{x} + \sum_{i=1}^{j} c_i \Delta\boldsymbol{x}^i) \approx \boldsymbol{r}^0 + \sum_{i=1}^{j} c_i \Delta\boldsymbol{r}^i \tag{34}$$

which can be considered as a finite-difference approximation. When the minimization problem in the residual space

$$\min_{\boldsymbol{c}} ||\boldsymbol{r}^0 + \sum_{i=1}^{j} c_i \Delta\boldsymbol{r}^i||_2 \tag{35}$$

of the GMRES method has converged sufficiently, the corresponding Newton-Raphson update is calculated as

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}} ||\boldsymbol{r}^0 + \sum_{i=1}^{j} c_i \Delta \boldsymbol{r}^i||_2 \tag{36a}$$

$$\Delta \boldsymbol{x} = \sum_{i=1}^{j} c_i \Delta \boldsymbol{x}^i. \tag{36b}$$

The coefficient vector $\boldsymbol{c}$ is calculated in the same way as in the IQN-ILS method, as shown on line 17 of Algorithm 4. The expression $\min_{\boldsymbol{c}} ||\boldsymbol{r}^0 + \sum_{i=1}^{j} c_i \Delta \boldsymbol{r}^i||_2$ is called the linearized residual ζ, as opposed to the true residual $||\boldsymbol{r}||_2$.

Algorithm 4 Interface-GMRES method [28]

```
 1: $k = 0;\ \tilde{\boldsymbol{x}}^1 = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^0);\ \boldsymbol{r}^0 = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
 2: while $||\boldsymbol{r}^k||_2 > \varepsilon_o$ do
 3:     $j = 0;\ \zeta = ||\boldsymbol{r}^k||_2$
 4:     while $\zeta > \epsilon_1$ do
 5:         $j = j + 1$
 6:         $\Delta \boldsymbol{x}^j = \tilde{\boldsymbol{x}}^j - \boldsymbol{x}^0$
 7:         for $i = 1$ to $j - 1$ do
 8:             $\Delta \boldsymbol{x}^j = \Delta \boldsymbol{x}^j - \frac{\Delta \boldsymbol{x}^{j\,\mathrm{T}} \Delta \boldsymbol{x}^i}{||\Delta \boldsymbol{x}^i||_2} \Delta \boldsymbol{x}^i$
 9:         end for
10:         $\Delta \boldsymbol{x}^j = \omega \frac{||\boldsymbol{r}^0||_2}{||\Delta \boldsymbol{x}^j||_2} \Delta \boldsymbol{x}^j$
11:         $\boldsymbol{x}^j = \boldsymbol{x}^0 + \Delta \boldsymbol{x}^j$
12:         $\tilde{\boldsymbol{x}}^{j+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^j)$
13:         $\Delta \boldsymbol{r}^j = (\tilde{\boldsymbol{x}}^{j+1} - \boldsymbol{x}^j) - \boldsymbol{r}^k$
14:         construct $V^j = [\Delta \boldsymbol{r}^1 \cdots \Delta \boldsymbol{r}^j]$
15:         construct $W^j = [\Delta \boldsymbol{x}^1 \cdots \Delta \boldsymbol{x}^j]$
16:         calculate QR-decomposition $V^j = Q^j R^j$
17:         solve $R^j \boldsymbol{c} = -Q^{j\,\mathrm{T}} \boldsymbol{r}^k$
18:         $\zeta = ||\boldsymbol{r}^k + V^j \boldsymbol{c}||_2$
19:     end while
20:     $\boldsymbol{x}^0 = \boldsymbol{x}^0 + W^j \boldsymbol{c}$
21:     $k = k + 1;\ \tilde{\boldsymbol{x}}^1 = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^0);\ \boldsymbol{r}^k = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
22: end while
```

For FSI with strong interaction, the relaxation on line 10 of Algorithm 4 is imperative as the inner loop consists of fixed-point iterations which diverge fast without relaxation. The optimal value of the relaxation factor ω is problem dependent and one is limited by the finite accuracy of the solvers for a lower bound and divergence for an upper bound. The orthonormalization of the $\Delta\boldsymbol{x}^i$ from line 7 to 9 is recommended in [28] to improve the accuracy of the solution of equation (35).

Algorithm 4 can be conceived as the construction of a model with the residual as input and the interface's position as output. Instead of using the model to update the interface's position in the inner loop, underrelaxed fixed-point iterations are performed until the model meets the user-defined tolerance ϵ_1. To prevent that equation (35) is solved too accurately during the first few Newton-Raphson iterations, ϵ_1 is set to a fraction λ of the current residual $||\boldsymbol{r}^k||$.

Instead of discarding the $\boldsymbol{x}^i$ and $\boldsymbol{r}^i$ from the previous Newton-Raphson iterations, they can be reused in the next Newton-Raphson iteration, which is called Interface-GMRESR. Algorithm 5 shows the alterations to Algorithm 4 that bring about reuse of the fixed-point iterations from previous Newton-Raphson iterations.

Algorithm 5 Interface-GMRESR method[28]

```
1:  $k = 0;\ \tilde{\boldsymbol{x}}^1 = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^0);\ \boldsymbol{r}^0 = \tilde{\boldsymbol{x}}^1 - \boldsymbol{x}^0$
2:  $j = 0;\ \zeta = ||\boldsymbol{r}^k||_2$
3:  while $||\boldsymbol{r}^k||_2 > \varepsilon_o$ do
4:      $\ldots$
21:     $k = k + 1;\ \tilde{\boldsymbol{x}}^{j+1} = \mathcal{S} \circ \mathcal{F}(\boldsymbol{x}^0);\ \boldsymbol{r}^k = \tilde{\boldsymbol{x}}^{j+1} - \boldsymbol{x}^0$
22:     solve $R^j \boldsymbol{c} = -{Q^j}^{\mathrm{T}} \boldsymbol{r}^k$
23:     $\zeta = ||\boldsymbol{r}^k + V^j \boldsymbol{c}||_2$
24: end while
```

The inner loop is executed at least once before going to the Newton-Raphson update; otherwise no additional $\Delta\boldsymbol{r}$ and $\Delta\boldsymbol{x}$ are known with respect to the previous Newton-Raphson update and the Newton-Raphson update would be ineffective. Every time a Newton-Raphson update is performed on line 21, the reference $\boldsymbol{x}^0$ is modified and the solvers are evaluated. However, this solver evaluation does not result in an additional difference $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{r}$, whereas IBQN-LS and IQN-ILS modify the reference in every iteration and obtain a difference of the input and output in every iteration except for the first one.

Due to the relaxation on line 10, the $\Delta\boldsymbol{x}^j$ in Algorithm 4 all have the same norm, regardless of whether it is the first or the final iteration in the time step. However, the step size is adapted to the initial residual. If the problem is nonlinear, the $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{r}$ can be inaccurate because this finite difference step size is suboptimal. Because the norm of the differences in IBQN-LS and IQN-LS is equal to the step size, it automatically decreases during the coupling iterations as they converge and these methods do not require a preset step size.

Reuse of differences from the previous time steps and not only from the previous Newton-Raphson iterations has been introduced by Michler [52] who mention that reuse increases the efficiency, but that it comes at the expense of robustness and therefore has to be used with some caution. Results for only 25 time steps have been presented in a simulation with a compressible fluid. To enable reuse in Interface-GMRESR, j should not be reset to 0 at the beginning of the time step. The convergence plots in [52] demonstrate that with reuse from the previous time steps, there is a large discrepancy between the linearized residual and the true residual. The inner loop with fixed-point iterations converges after one iteration because the linearized residual already is very small. Consequently, the algorithm repeatedly performs one fixed-point iteration followed by a Newton-Raphson update and in this regime an additional $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{r}$ are obtained at the cost of two solver evaluations.

8. Results

In this section, the performance of the coupling algorithms is compared for two different cases, namely a 2D flexible beam behind a cylinder and the propagation of a pressure pulse in a 3D flexible tube.

8.1. 2D flexible beam

The first cases are the steady FSI1 test and the unsteady FSI2 test from the benchmark in [41]. This test is well-documented in the aforementioned reference and thus it will only be described briefly. As can be seen in Figure 4, the geometry consists of a horizontal channel of 0.41 m high, containing a rigid cylinder with center positioned 0.2 m above the bottom of the channel. A laminar, viscous flow with density 10^3 kg/m^3 and kinematic viscosity 10^{-3} m^2/s enters the channel from the left hand side with a parabolic velocity profile. A constant pressure is imposed at the outlet of the channel and a no-slip boundary condition is applied on the walls. The mean inlet velocity is 0.2 m/s for FSI1 and 1.0 m/s for FSI2. A linearly elastic beam is attached to the right hand side of the cylinder and this beam has a density of 10^3 kg/m^3 or 10^4 kg/m^3, respectively in FSI1 and FSI2, a Young's modulus of $1.40 \cdot 10^6$ N/m^2 and a Poisson's ratio of 0.4.

The finite volume flow solver uses second order discretization for the pressure and second order upwind for the momentum. It solves the Navier-Stokes equations in arbitrary Lagrangian Eulerian (ALE) formulation for the pressure and velocity simultaneously with first order time accuracy. The mesh of the fluid domain is adapted to the position of the fluid-structure interface with a spring analogy and remeshing is performed if the skewness of the cells is too large. The spring analogy means that the edges between the grid nodes are replaced by springs, with a spring constant inversely proportional to the length of the edge. This results in a pseudo-structural problem with the grid displacement as unknowns [53]. The finite element structural solver uses implicit time integration of plain strain elements with 8 nodes and takes into account the geometric nonlinearities due to the large deformation of the structure. The coupling algorithm is executed on one processor, the flow solver and structural solver each on two processors.

After validating the flow solver and the structural solver with the tests listed in [41], the FSI1 and FSI2 tests have been performed. Table 1 and Figure 5 give the displacement in the x and y direction of the point A, which is located at the right end of the beam, and the drag and lift force on the entire structure in the same format as in the benchmark paper [41]. The difference between the simulations of the FSI1 test decreases as the grid is refined and the results correspond well with the benchmark values, even for the FSI2 test considering that the flow solver is only first-order accurate in time on moving meshes. The same results have been found with each coupling method. Pressure contours in the vicinity of the structure are shown in Figure 4.

The number of solver evaluations and the relative duration of the simulations are given in Table 2. The residual $||\boldsymbol{r}_i||_2$ is reduced five orders of magnitude with respect to its initial value and the number between brackets behind the name of an algorithm indicates from how many time steps information is reused. For FSI1, the number of solver evaluations is independent of the grid refinement. Interface-GMRESR has been used with $\lambda = 0.01$ because this resulted in the lowest number of solver evaluations. In the unsteady FSI2 test, Interface-GMRESR requires more coupling iterations than Interface-GMRES, especially when the deformation within a time step is large. Reuse of information from 3 time steps reduces the number of coupling iterations with approximately 30 % for both IQN-ILS and IBQN-LS. For this unsteady simulation, the duration of the simulation is similar for IQN-ILS(3) and IBQN-LS(3), which are significantly faster than Aitken relaxation and Interface-GMRES(R).

8.2. 3D flexible tube

The second test case is a 3D simulation of a straight flexible tube — representing an artery — with radius 0.005 m and length 0.05 m, as described by Fernandez and Moubachir [24], Formaggia et al. [54], Gerbeau and Vidrascu [26]. The same flow solver and structural solver as in Section 8.1 have been used, but the structure has been modeled with one layer of shell elements. The tube's wall is a linear elastic material with density 1200 kg/m^3, Young's modulus $3{\cdot}10^5$ Nm^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 dynamic viscosity of 0.003 Pa·s.

Both the fluid and the structure are initially at rest. During the first $3{\cdot}10^{-3}$ 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. The fluid model contains 37128 degrees-of-freedom and the structure model 29760, with $u = 4896$ degrees-of-freedom in the interface's position and $w = 4992$ degrees-of-freedom in the stress distribution on the fluid-structure interface. Pressure contours on the fluid-structure interface are shown in Figure 6 and they correspond well with the above mentioned references. The same results have been found with each coupling method.

Table 3 gives the number of solver evaluations in a time step, averaged over the entire simulation, and the relative duration of the simulations. In every time step, $||\boldsymbol{r}||_2$ is reduced three orders of magnitude with respect to its initial value in that time step. The results demonstrate that the performance of the coupling methods is different with respect to number of solver evaluations and CPU time. The reason for this difference is that if the coupling algorithm predicts an irregular position of the interface or pressure distribution on the interface, it will take the flow solver and structural solver longer to converge. For this case, the duration of the simulation is again similar for IBQN-LS(10) and IQN-ILS(10), which are faster than the other methods.

9. Conclusion

It has been demonstrated that the complete Jacobian is not required in an interface Newton-Raphson technique for partitioned simulation of fluid-structure interaction. Based on this insight, an interface quasi-Newton technique with an approximation for the product of the inverse of the Jacobian with a vector (IQN-ILS) and reuse of information from previous time steps has been developed [39]. A comparison of the IQN-ILS algorithm with IBQN-LS, Aitken relaxation and Interface-GMRES(R) indicates that the IQN-ILS method can easily be implemented in an FSI framework that currently uses (Aitken) relaxation or Interface-GMRES(R) and that IQN-ILS is more straightforward to implement than IBQN-LS. The results show that the IQN-ILS method performs similarly to the IBQN-LS method and better than Aitken relaxation and Interface-GMRES(R).

Acknowledgments

The authors gratefully acknowledge the funding of Joris Degroote and Peter Bruggeman by a Ph.D. fellowship of the Research Foundation - Flanders (FWO). Joris Degroote acknowledges a grant for a long stay abroad at the Massachusetts Institute of Technology (MIT) of the Research Foundation - Flanders. Jan Vierendeels acknowledges the funding by Research Project G027508 of the Research Foundation - Flanders and the Ghent University Association.

References

[1] 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 (2006) 1973–2001.

[2] L. Cavagna, G. Quaranta, P. Mantegazza, Application of Navier-Stokes simulations for aeroelastic stability assessment in transonic regime, Computers and Structures 85 (11–14) (2007) 818–832.

[3] K. Stein, T. Tezduyar, V. Kumar, S. Sathe, R. Benney, R. Charles, Numerical simulation of soft landing for clusters of cargo parachutes, in: P. Niettaanmäki, T. Rossi, K. Majava, O. Pieronneau (Eds.), European congress on computational methods in applied sciences and engineering ECCOMAS 2004, Jyväskylä, 1–14, 2004.

[4] C. Lee, H. Noguchi, S. Koshizuka, Fluid-shell structure interaction analysis by coupled particle and finite element method, Computers and Structures 85 (11–14) (2007) 688–697.

[5] S. Potapov, B. Maurel, A. Combescure, J. Fabis, Modeling accidental-type fluid-structure interaction problems with the SPH method, Computers and Structures 87 (11–12) (2009) 721–734.

[6] K. Riemslagh, J. Vierendeels, E. Dick, Coupling of a Navier-Stokes solver and an elastic boundary solver for unsteady problems, in: K. Papailiou,

D. Tsahalis, J. Périaux, C. Hirsch, M. Pandolfi (Eds.), 4th European Computational Fluid Dynamics Conference, ECCOMAS, Athens, Greece, 1040–1045, 1998.

[7] 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.

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

[9] N. dos Santos, J.-F. Gerbeau, J.-F. Bourgat, Partitioned FSI strategy for simulations of a thin elastic valve, in: P. Wesseling, E. Oñate, J. Périaux (Eds.), European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006, ECCOMAS, Delft, The Netherlands, 1–10, 2006.

[10] R. van Loon, P. Anderson, F. van de Vosse, A fluid-structure interaction method with solid-rigid contact for heart valve dynamics, Journal of Computational Physics 217 (2006) 806–823.

[11] 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, ASME Journal of Biomechanical Engineering 129 (4) (2007) 558–565.

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

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

[14] H. Jahromi, B. Izzuddin, L. Zdravkovic, Partitioned analysis of nonlinear soil-structure interaction using iterative coupling, Interaction and Multiscale Mechanics 1 (1) (2007) 33–51.

[15] W. Elleithy, M. Tanaka, Interface relaxation algorithms for BEM-BEM coupling and FEM-BEM coupling, Computer Methods in Applied Mechanics and Engineering 192 (2003) 2977–2992.

[16] E. Oñate, S. Idelsohn, F. Del Pin, R. Aubry, The particle finite element method. An overview, International Journal of Computational Methods 1 (2004) 267–307.

[17] J. Monaghan, Simulating free surface flows with SPH, Journal of Computational Physics 110 (2) (1994) 399–406.

[18] C. Peskin, D. McQueen, A three-dimensional computational method for blood flow in the heart I. immersed elastic fibers in a viscous incompressible fluid, Journal of Computational Physics 81 (1989) 372–405.

[19] R. van Loon, P. Anderson, F. van de Vosse, S. Sherwin, Comparison of various fluid-structure interaction methods for deformable bodies, Computers and Structures 85 (11–14) (2007) 833–843.

[20] 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, ISBN 3-540-34595-7, 2006.

[21] H. Matthies, J. Steindorf, Partitioned strong coupling algorithms for fluid-structure interaction, Computers and Structures 81 (2003) 805–812.

[22] H. Matthies, R. Niekamp, J. Steindorf, Algorithms for strong coupling procedures, Computer Methods in Applied Mechanics and Engineering 195 (2006) 2028–2049.

[23] M. Fernandez, M. Moubachir, An exact block-Newton algorithm for solving fluid-structure interaction problems, Comptes Rendus de l'Academie des Sciences - Series I: Mathematics 336 (8) (2003) 681–686.

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

[25] 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 (2006) 2124–2148.

[26] 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.

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

[28] E. van Brummelen, C. Michler, R. de Borst, Interface-GMRES(R) Acceleration of Subiteration for Fluid-Structure-Interaction Problems, Report DACS-05-001, available from: http://www.em.lr.tudelft.nl/downloads/DACS-05-001.pdf., 2005.

[29] 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.

[30] U. Küttler, W. Wall, Fixed-point fluid-structure interaction solvers with dynamic relaxation, Computational Mechanics 43 (1) (2008) 61–72, DOI: 10.1007/s00466-008-0255-5.

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

[32] 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 (1995) 79–112.

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

[34] A. v. Zuijlen, S. Bosscher, H. Bijl, Two level algorithms for partitioned fluid-structure interaction computations, Computer Methods in Applied Mechanics and Engineering 196 (2007) 1458–1470.

[35] 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 (2007) 414–430.

[36] J. Vierendeels, Implicit coupling of partitioned fluid-structure interaction solvers using reduced-order models, 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, 1–18, ISBN 3-540-34595-7, 2006.

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

[38] J. Degroote, S. Annerel, J. Vierendeels, Stability analysis of Gauss-Seidel iterations in a partitioned simulation of fluid-structure interaction, Computers and Structures In press, doi:10.1016/j.compstruc.2009.09.003.

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

[40] R. Haelterman, J. Degroote, D. Van Heule, J. Vierendeels, The quasi-Newton least squares method: a new and fast secant method analyzed for linear systems, SIAM Journal on Numerical Analysis 47 (3) (2009) 2347–2368.

[41] S. Turek, J. Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, 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, 371–385, ISBN 3-540-34595-7, 2006.

[42] K.-J. Bathe, Encyclopedia of Computer Science and Engineering, chap. The Finite Element Method, John Wiley and Sons, 1, b. Wah (ed.), dx.doi.org/10.1002/9780470050118.ecse159, 2008.

[43] C. Farhat, M. Lesoinne, P. Le Tallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity, Computer Methods in Applied Mechanics and Engineering 157 (1998) 95–114.

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

[45] A. de Boer, M. van der Schoot, H. Bijl, Mesh deformation based on radial basis function interpolation, Computers and Structures 85 (11–14) (2007) 784–795.

[46] J. Vierendeels, K. Dumont, E. Dick, P. Verdonck, Analysis and stabilization of fluid-structure interaction algorithm for rigid-body motion, AIAA Journal 43 (12) (2005) 2549–2557.

[47] G. H. Golub, C. F. V. Loan, Matrix computations, Johns Hopkins University Press, Baltimore, MD, USA, 3rd edn., 1996.

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

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

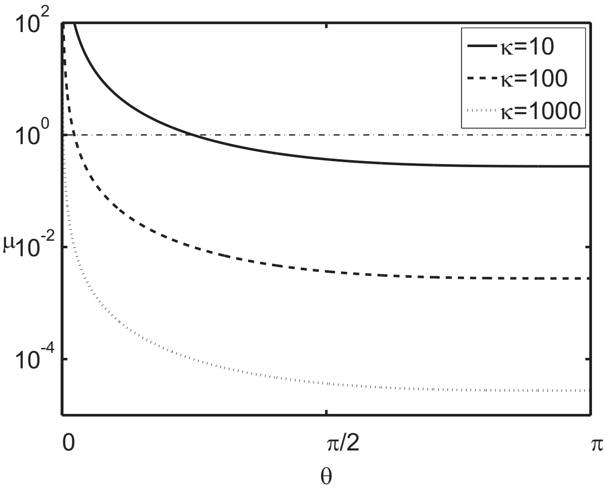
[50] J. Nocedal, S. Wright, Numerical Optimization, Springer Series in Operations Research, Springer, 1999.

[51] P. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM Journal on Scientific and Statistical Computing 11 (3) (1990) 450–481.

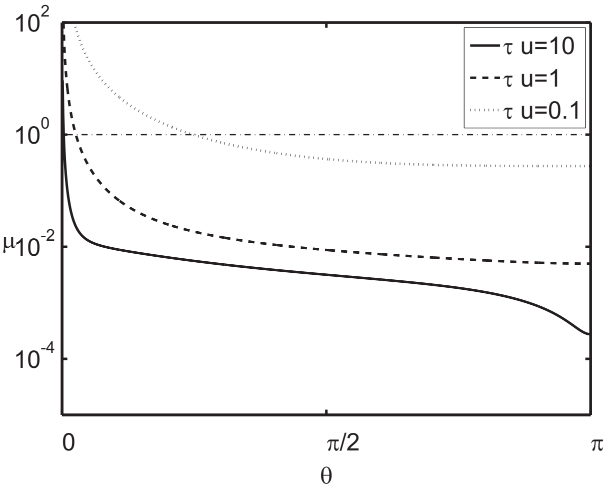
[52] S. Michler, Efficient numerical methods for Fluid-Structure Interaction, Phd thesis, TUDelft, 2005.

[53] J. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA Journal 28 (8) (1990) 1381–1388.

[54] 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 (2001) 561–582.



(a)



(b)

Figure 1: Amplification of the error on the interface's position as a function of the wave number (θ). This result has been obtained by means of Fourier error analysis on the Gauss-Seidel iterations between the flow solver and structural solver in a simulation of 1D unsteady, incompressible flow in a flexible tube [37] with (a) constant product of the dimensionless time step τ and the number of degrees-of-freedom u in the discretization of the interface ($\tau u = 0.1$) and (b) constant dimensionless stiffness of the tube ($\kappa = 10$).

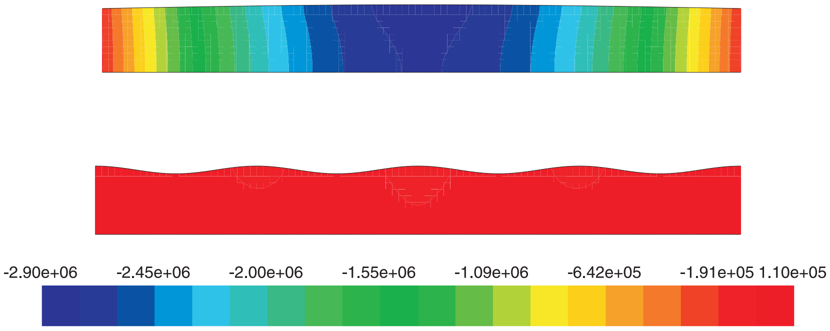


Figure 2: Pressure contours in an axisymmetric tube due to two displacements of the tube's wall with the same amplitude but a different wave number. Initially, the fluid is at rest and the tube has a constant cross section. The straight horizontal line is the axis of symmetry, the curved horizontal line is the tube's wall. A displacement of the tube's wall with a low wave number (top) creates much larger pressure variations than a displacement with the same amplitude but a higher wave number (bottom). Only the difference between the two calculations and not the values as such are important.

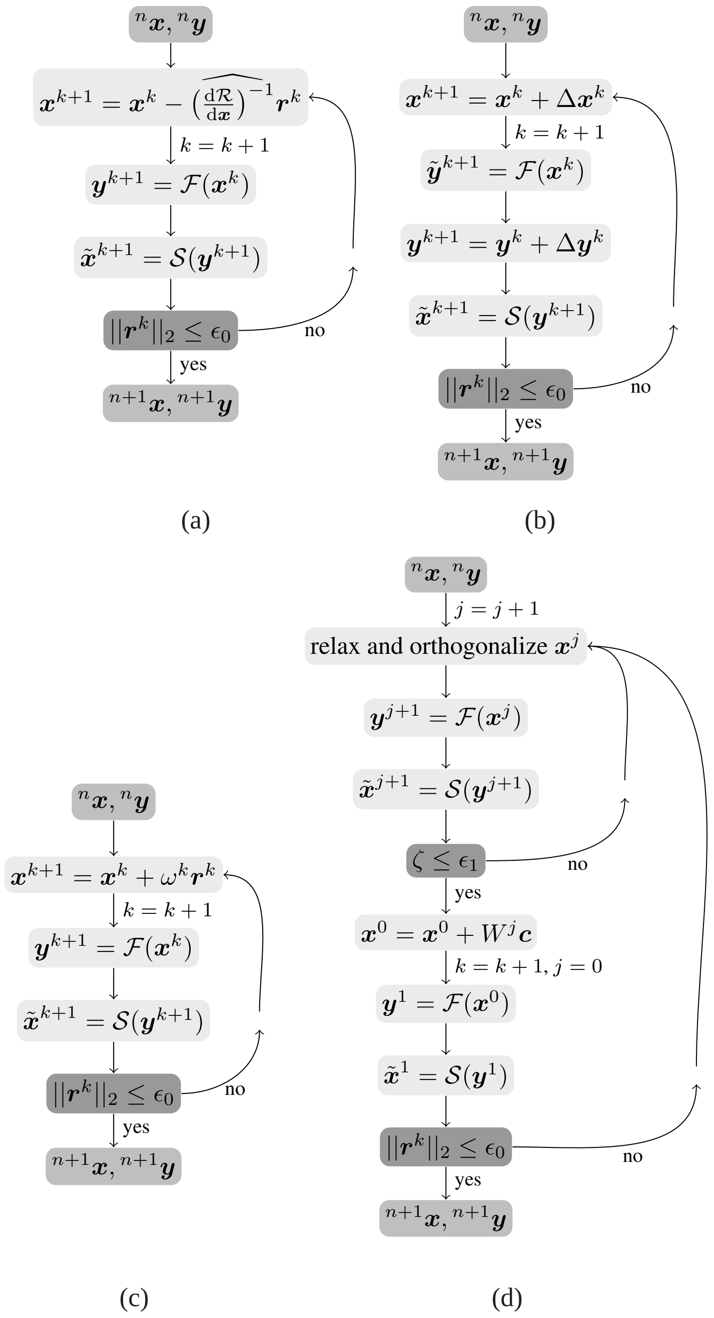
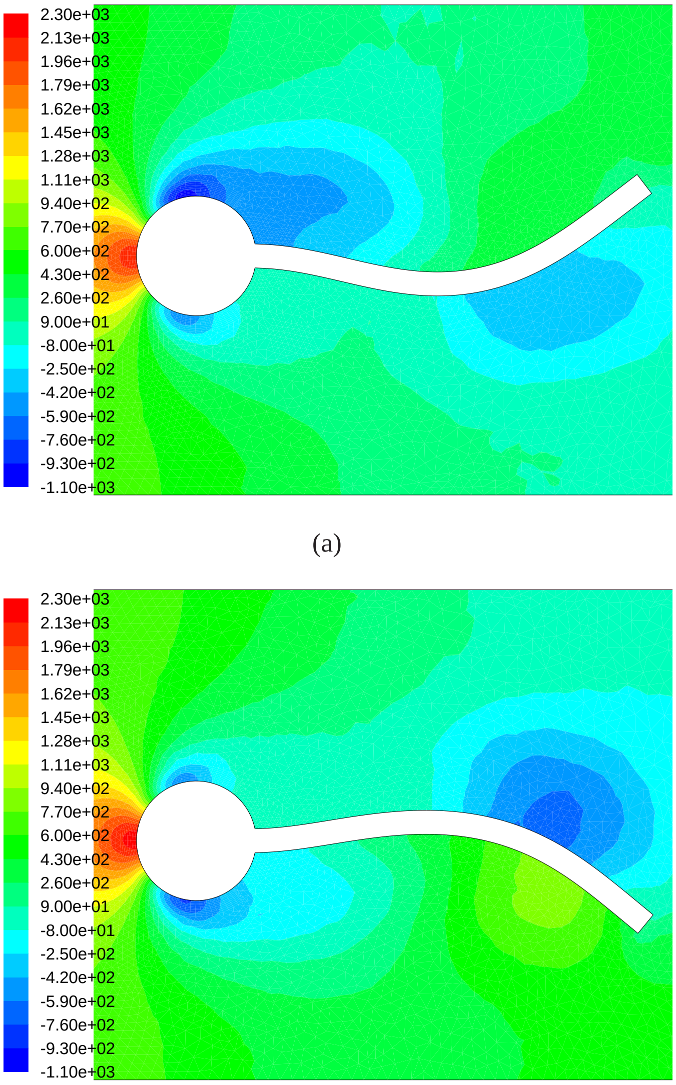

$^{n}\boldsymbol{x}, {}^{n}\boldsymbol{y}$
$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \left(\widehat{\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\boldsymbol{x}}}\right)^{-1}\boldsymbol{r}^k$
$k = k+1$
$\boldsymbol{y}^{k+1} = \mathcal{F}(\boldsymbol{x}^k)$
$\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S}(\boldsymbol{y}^{k+1})$
$||\boldsymbol{r}^k||_2 \leq \epsilon_0$
no
yes
$^{n+1}\boldsymbol{x}, {}^{n+1}\boldsymbol{y}$
(a)
$^{n}\boldsymbol{x}, {}^{n}\boldsymbol{y}$
$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \Delta\boldsymbol{x}^k$
$k = k+1$
$\tilde{\boldsymbol{y}}^{k+1} = \mathcal{F}(\boldsymbol{x}^k)$
$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \Delta\boldsymbol{y}^k$
$\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S}(\boldsymbol{y}^{k+1})$
$||\boldsymbol{r}^k||_2 \leq \epsilon_0$
no
yes
$^{n+1}\boldsymbol{x}, {}^{n+1}\boldsymbol{y}$
(b)
$^{n}\boldsymbol{x}, {}^{n}\boldsymbol{y}$
$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \omega^k\boldsymbol{r}^k$
$k = k+1$
$\boldsymbol{y}^{k+1} = \mathcal{F}(\boldsymbol{x}^k)$
$\tilde{\boldsymbol{x}}^{k+1} = \mathcal{S}(\boldsymbol{y}^{k+1})$
$||\boldsymbol{r}^k||_2 \leq \epsilon_0$
no
yes
$^{n+1}\boldsymbol{x}, {}^{n+1}\boldsymbol{y}$
(c)
$^{n}\boldsymbol{x}, {}^{n}\boldsymbol{y}$
$j = j+1$
relax and orthogonalize $\boldsymbol{x}^j$
$\boldsymbol{y}^{j+1} = \mathcal{F}(\boldsymbol{x}^j)$
$\tilde{\boldsymbol{x}}^{j+1} = \mathcal{S}(\boldsymbol{y}^{j+1})$
$\zeta \leq \epsilon_1$
no
yes
$\boldsymbol{x}^0 = \boldsymbol{x}^0 + W^j\boldsymbol{c}$
$k = k+1, j = 0$
$\boldsymbol{y}^1 = \mathcal{F}(\boldsymbol{x}^0)$
$\tilde{\boldsymbol{x}}^1 = \mathcal{S}(\boldsymbol{y}^1)$
$||\boldsymbol{r}^k||_2 \leq \epsilon_0$
no
yes
$^{n+1}\boldsymbol{x}, {}^{n+1}\boldsymbol{y}$
(d)

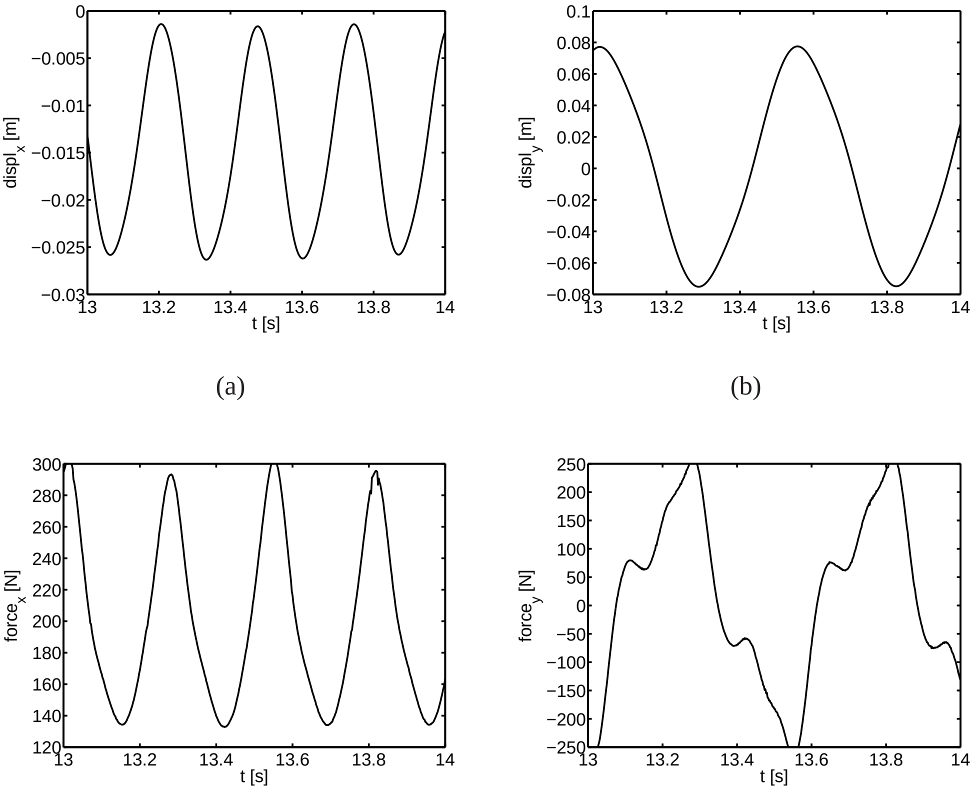
Figure 3: Simplified representation of the (a) IQN-ILS, (b) IBQN-LS, (c) (Aitken) relaxation and (d) Interface-GMRES algorithm for partitioned FSI simulations with black-box solvers.



(a)

(b)

Figure 4: Pressure contours in the 2D unsteady FSI2 test with the flexible beam after (a) 12 s; (b) 16 s.



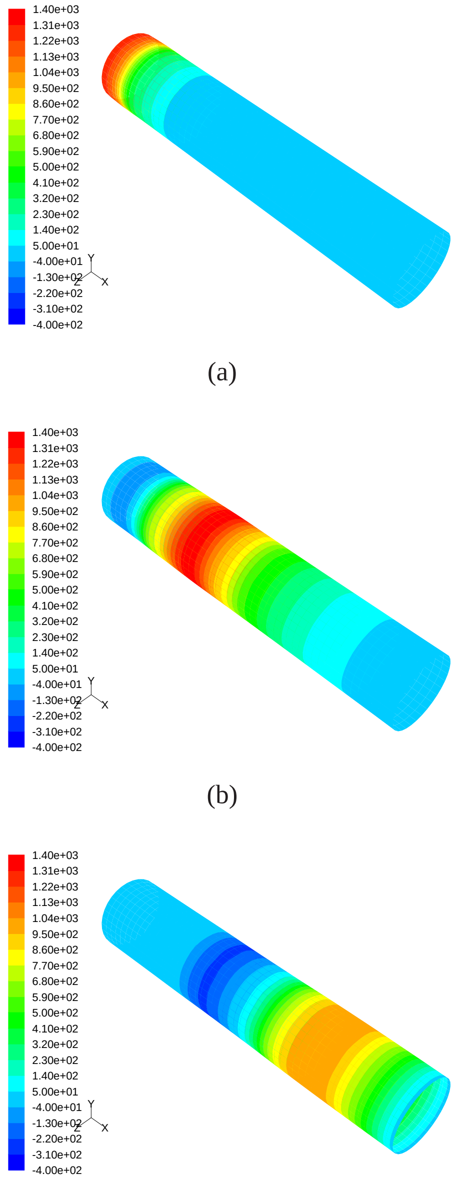
(a)

(b)

(c)

(d)

Figure 5: Displacement and force in the 2D unsteady FSI2 test with the flexible beam. Displacement of node A in (a) the x-direction; (b) the y-direction. Force on the beam and cylinder in (c) the x-direction; (d) the y-direction.



(a)

(b)

(c)

Figure 6: Pressure contours on the fluid-structure interface in a 3D simulation of the flexible tube after (a) 10^{-3} s; (b) $5{\cdot}10^{-3}$ s; (c) $9{\cdot}10^{-3}$ s.

Results for the 2D flexible beam

FSI1				
DOF ($\mathcal{F} + \mathcal{S}$)	displ$_x$ [10^{-3} m]	displ$_y$ [10^{-3} m]	force$_x$ [N]	force$_y$ [N]
74418+7650	0.022643	0.83672	14.234	0.75416
164394+16874	0.022645	0.83489	14.254	0.75616
290574+29698	0.022651	0.83478	14.266	0.75753
Benchmark	0.022700	0.82090	14.295	0.76380

FSI2				
DOF ($\mathcal{F} + \mathcal{S}$)	displ$_x$ [10^{-3} m]	displ$_y$ [10^{-3} m]	force$_x$ [N]	force$_y$ [N]
89760+7650	-14.07±12.37[3.7]	1.18±76.5[1.9]	217.52±84.65[3.7]	-0.74±267.6[1.9]
Benchmark	-14.58±12.44[3.8]	1.23±80.6[2.0]	208.83±73.75[3.8]	0.88±234.2[2.0]

Table 1: Displacement and force in the 2D steady FSI1 and unsteady FSI2 test with the flexible beam for different grids. The number of degrees-of-freedom (DOF) is indicated for each grid, separately for the flow solver and the structural solver, and this number varies slightly in the unsteady test due to remeshing. For the FSI2 test, the values are given as mean±amplitude[frequency].

Performance of the algorithms for the 2D flexible beam

FSI1		
Algorithm	Evaluations	Duration
IBQN-LS	5	1.18
IQN-ILS	5	1.00
Aitken relaxation	8	1.42
Interface-GMRES	7	1.35
Interface-GMRESR	6	1.00

FSI2		
Algorithm	Evaluations	Duration
IBQN-LS	7.2	1.54
IBQN-LS(3)	4.8	1.00
IQN-ILS	9.4	1.84
IQN-ILS(3)	6.1	1.07
Aitken relaxation	9.9	1.81
Interface-GMRES	10.5	1.94
Interface-GMRESR	14.4	2.71
Interface-GMRESR(3)	12.4	2.71

Table 2: Number of solver evaluations and relative duration for the 2D steady FSI1 test and unsteady FSI2 test with the flexible beam. The data are independent of the grid refinement in the steady simulation and the number of solver evaluations per time step in the unsteady simulation has been averaged over the last period of the oscillation.

Performance of the algorithms for the 3D flexible tube

Algorithm	Evaluations	Duration
IBQN-LS(2)	8.2	1.29
IBQN-LS(10)	6.3	1.00
IQN-ILS(2)	8.4	1.30
IQN-ILS(10)	6.6	1.03
Aitken relaxation	26.7	4.69
Interface-GMRES	16.1	2.71
Interface-GMRESR(2)	11.7	1.79
Interface-GMRESR(10)	9.5	1.45

Table 3: Number of solver evaluations per time step and relative duration for the simulation of the 3D flexible tube. The number of solver evaluations per time step has been averaged over the entire simulation.