A PARTITIONED QUASI-NEWTON SOLUTION TECHNIQUE FOR FLUID-STRUCTURE INTERACTION PROBLEMS USING A COARSENED GRID TO ACCELERATE THE CONVERGENCE OF THE COUPLING ITERATIONS

JORIS DEGROOTE*[*] AND JAN VIERENDEELS[*]

[*]Department of Flow, Heat and Combustion Mechanics
Ghent University
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
e-mail: {Joris.Degroote,Jan.Vierendeels}@UGent.be, www.FSI.UGent.be

Key words: partitioned, fluid-structure interaction, quasi-Newton, least-squares, grid

Abstract. Previous stability analyses on Gauss-Seidel coupling iterations in partitioned fluid-structure interaction simulations have demonstrated that Fourier modes with a low wave-number in the difference between the current and correct interface displacement are unstable. To stabilize these modes, the IQN-ILS technique automatically constructs a least-squares model of the flow solver and structural solver. In this work, the multi-level IQN-ILS technique (ML-IQN-ILS) is presented, which uses a coarsened grid of the fluid and structure subdomains to initialize this least-squares model. As the modes that need to be present in this least-squares model have a low wave-number, they can be resolved on a coarsened grid. Therefore, in each time step, a number of cheap coupling iterations is first performed on the coarsened grid to construct the model, followed by a smaller number of coupling iterations on the fine grid. As the iterations on the coarse grid are fast and fewer iterations are performed on the fine grid, the total duration of the simulation decreases compared to a simulation on the fine grid only.

1 INTRODUCTION

Partitioned fluid-structure interaction (FSI) simulation techniques solve the flow equations and the structural equations separately. In this article, the focus lies on partitioned 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. Implicit (or strongly coupled) partitioned techniques enforce the equilibrium of the stress and velocity (or displacement) on the fluid-structure interface in each time step. Several strongly coupled partitioned techniques are able to couple ‘black-box’ solvers, for example Gauss-Seidel iterations with Aitken relaxation [1], the Interface Generalized Minimal Residual method (Interface-GMRES) [2], the Interface Block Quasi-Newton technique with an approximation for the Jacobians from Least-Squares models (IBQN-LS) [3] and the Interface Quasi-Newton technique with an approximation for the Inverse of the Jacobian from a Least-Squares model (IQN-ILS) [4].

Several stability analyses on coupling algorithms have been performed for the incompressible, inviscid flow in a straight, elastic tube [5–8]. Fourier analysis has been performed on the difference between the current and the correct interface displacement during Gauss-Seidel coupling iterations in [7, 8]. From these analyses, two lessons can be learned. While the standard, one-level IQN-ILS technique only takes advantage of the first one, the new multi-level IQN-ILS (ML-IQN-ILS) technique takes advantage of both of them.

The first lesson is that only a fraction of the Fourier modes is unstable during Gauss-Seidel iterations. If a quasi-Newton technique is used, then only a low-rank approximation for the exact Jacobian is required, as long as it represents the behaviour of these unstable and slowly converging modes, which explains the performance of quasi-Newton methods like IQN-ILS. For combinations of Fourier modes that are covered by the least-squares model, IQN-ILS performs Newton iterations; for the other modes, IQN-ILS corresponds to Gauss-Seidel iterations.

The second lesson is that the unstable modes have a low wave number, so their behaviour 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 IQN-ILS while doing so. Then coupling iterations are performed on the second grid level, during which the approximation for the inverse of the Jacobian obtained on the coarsest grid level is further improved. This procedure is repeated until the solution on the finest grid level has been found.

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. This new multi-level approach is depicted in Figure 1 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 [9], which has been used for fluid-structure interaction simulations in for example [10, 11] and which is different from the ML-IQN-ILS technique.

The remainder of this article is organized as follows. Section 2 gives a brief overview of the governing equations, before the detailed explanation of ML-IQN-ILS in Section 3. Numerical results in Section 4 illustrate the performance of ML-IQN-ILS compared to the standard, one-level IQN-ILS. Finally, Section 5 offers the conclusions.

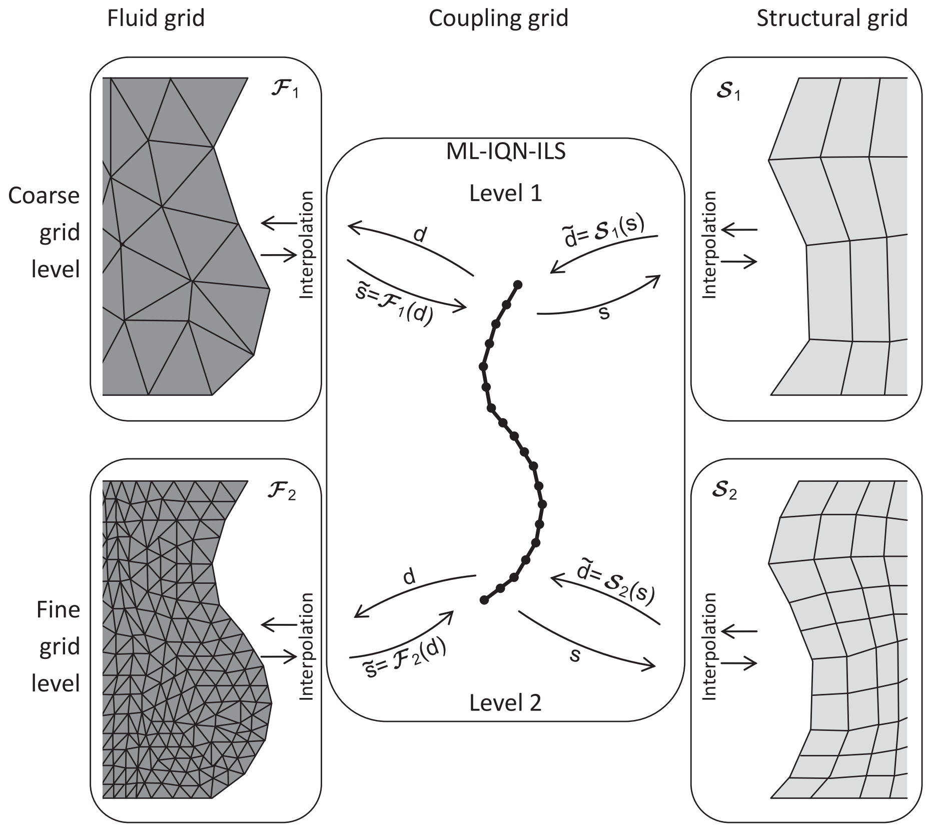

Figure 1: The coarse and fine fluid grid (left) and the coarse and fine structural grid (right), together with the unique coupling grid (centre) in a multi-level simulation with two grid levels. $\boldsymbol{d}$ represents the displacement of the interface while $\boldsymbol{s}$ represents the stress 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. 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.

2 GOVERNING EQUATIONS

The fluid and structure subdomains are indicated as Ω_f and Ω_s and their boundaries as Γ_f and Γ_s, respectively. The fluid-structure interface $\Gamma_{fs} = \Gamma_f \cap \Gamma_s$ is the common boundary of these subdomains. This article only considers incompressible fluids as they prove to be most challenging for the partitioned fluid-structure interaction techniques. The unsteady flow of an incompressible fluid is governed by the conservation of mass and the Navier-Stokes equations

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

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

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

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

with ρ_s the structural density, $\bar{\sigma}_s$ the Cauchy stress tensor and $\vec{f}_s$ the body forces per unit volume on the structure.

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

$$\vec{v} = \frac{\mathrm{d}\vec{u}}{\mathrm{d}t} \quad \text{and} \quad \bar{\sigma}_f \cdot \vec{n}_f = -\bar{\sigma}_s \cdot \vec{n}_s, \tag{3}$$

which stipulate that the velocity and the stress 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). 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 stress on the interface.

The flow equations and the structural equations are discretized in space and time with a method of choice. The discrete flow equations are represented by $\boldsymbol{\mathfrak{F}}$, the discrete structural equations by $\boldsymbol{\mathfrak{S}}$. 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 stress 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 stress on the interface as a function of the flow degrees of freedom ($\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{v})$). 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 $\boldsymbol{\mathfrak{F}}(\boldsymbol{v}, \boldsymbol{d}(\boldsymbol{u})) = \boldsymbol{0}$ for a given interface displacement $\boldsymbol{d}$. From the flow field $\boldsymbol{v}$, the stress on the interface $\boldsymbol{s}$ is extracted. Therefore, the flow solver is represented by the function

$$\tilde{\boldsymbol{s}} = \boldsymbol{\mathcal{F}}(\boldsymbol{d}). \tag{4}$$

Similarly, the structural solver calculates the structural variables $\boldsymbol{u}$ that satisfy $\boldsymbol{\mathfrak{S}}(\boldsymbol{u}, \boldsymbol{s}(\boldsymbol{v})) = \boldsymbol{0}$ for a given stress 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

$$\tilde{\boldsymbol{d}} = \boldsymbol{\mathcal{S}}(\boldsymbol{s}). \tag{5}$$

As the multi-level coupling technique uses 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 1). In this work, this coupling grid is identical to the interface discretization of the finest fluid 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. A local interpolant is constructed in the neighbourhood of each point on the interface using a basis function introduced by Wendland [12], namely

$$\phi(||\vec{x}||/r) = (1 - ||\vec{x}||/r)^4_+(4||\vec{x}||/r + 1), \tag{6}$$

with r the radius and $||\vec{x}|| = \sqrt{x_1^2 + \ldots + x_d^2}$ the Euclidean distance of dimension d. The plus-sign behind the first term denotes that this term is zero if $1 - ||\vec{x}||/r < 0$ such that ϕ has a compact support.

3 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 this value is not always directly given as input to the other solver. 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

$$\boldsymbol{\mathcal{R}}(\boldsymbol{d}) = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}) - \boldsymbol{d} = \boldsymbol{0} \tag{7}$$

can be solved by means of Newton-Raphson iterations

$$\text{solve}\quad \boldsymbol{\mathcal{R}}'^{k}\Delta\boldsymbol{d}^{k} = -\boldsymbol{r}^{k} \tag{8a}$$

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^{k} + \Delta\boldsymbol{d}^{k} \tag{8b}$$

with the residual calculated as

$$\boldsymbol{r}^{k} = \boldsymbol{\mathcal{R}}(\boldsymbol{d}^{k}) = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{k}) - \boldsymbol{d}^{k} = \tilde{\boldsymbol{d}}^{k} - \boldsymbol{d}^{k}. \tag{9}$$

$\boldsymbol{\mathcal{R}}'^{k}$ 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. (8a) 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. (8a) 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. [3] 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

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^{k} + \widehat{\Delta\boldsymbol{d}}^{k} = \boldsymbol{d}^{k} + \widehat{\left(\boldsymbol{\mathcal{R}}'^{k}\right)^{-1}}\left(-\boldsymbol{r}^{k}\right). \tag{10}$$

It can be seen from Eq. (10) 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 $\widehat{\Delta\boldsymbol{r}}^{k} = \boldsymbol{0} - \boldsymbol{r}^{k} = -\boldsymbol{r}^{k}$. The matrix-vector product in Eq. (10) is calculated from information obtained during the previous quasi-Newton iterations. Eq. (9) 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

$$\boldsymbol{r}^{k},\quad \boldsymbol{r}^{k-1},\quad \ldots,\quad \boldsymbol{r}^{1},\quad \boldsymbol{r}^{0} \tag{11a}$$

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

$$\tilde{\boldsymbol{d}}^{k},\quad \tilde{\boldsymbol{d}}^{k-1},\quad \ldots,\quad \tilde{\boldsymbol{d}}^{1},\quad \tilde{\boldsymbol{d}}^{0} \tag{11b}$$

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 using

$$\Delta\boldsymbol{r}^{k-1} = \boldsymbol{r}^{k} - \boldsymbol{r}^{k-1} \quad \text{and} \quad \Delta\tilde{\boldsymbol{d}}^{k-1} = \tilde{\boldsymbol{d}}^{k} - \tilde{\boldsymbol{d}}^{k-1}. \tag{12}$$

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 $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$. 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 $\widehat{\Delta\boldsymbol{r}}^k = \boldsymbol{0} - \boldsymbol{r}^k$ is approximated as a linear combination of the known $\Delta\boldsymbol{r}^j$

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

with $\boldsymbol{c}^k \in \mathbb{R}^{v\times 1}$ the coefficients of the decomposition. Because $v \leq u$, Eq. (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. For that reason, the so-called economy-size QR-decomposition of $\boldsymbol{V}^k$ is calculated using Householder transformations

$$\boldsymbol{V}^k = \boldsymbol{Q}^k\boldsymbol{R}^k, \tag{14}$$

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

$$\boldsymbol{R}^k\boldsymbol{c}^k = \boldsymbol{Q}^{k\mathrm{T}}\widehat{\Delta\boldsymbol{r}}^k \tag{15}$$

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 $\boldsymbol{R}^k$ 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. (14)) 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 $\widehat{\Delta\tilde{\boldsymbol{d}}}^k$ that corresponds to $\widehat{\Delta\boldsymbol{r}}^k$ is subsequently calculated as a linear combination of the previous $\Delta\tilde{\boldsymbol{d}}^j$, analogous to Eq. (13), giving

$$\widehat{\Delta\tilde{\boldsymbol{d}}}^k = \boldsymbol{W}^k\boldsymbol{c}^k. \tag{16}$$

From Eq. (9), it follows that $\Delta\boldsymbol{r} = \Delta\tilde{\boldsymbol{d}} - \Delta\boldsymbol{d}$ and substitution of Eq. (16) results in

$$\widehat{\Delta\boldsymbol{d}}^k = \boldsymbol{W}^k\boldsymbol{c}^k - \widehat{\Delta\boldsymbol{r}}^k. \tag{17}$$

Because the coefficients $\boldsymbol{c}^k$ are a function of $\widehat{\Delta\boldsymbol{r}}^k$, Eq. (17) shows how $\widehat{\Delta\boldsymbol{d}}^k$ can be approximated for a given $\widehat{\Delta\boldsymbol{r}}^k$. Hence, Eq. (17) can be seen as a procedure to calculate the product of the approximation for the inverse of the Jacobian and a vector $\widehat{\Delta\boldsymbol{r}}^k = -\boldsymbol{r}^k$

$$\widehat{\Delta\boldsymbol{d}}^k = \left(\widehat{\boldsymbol{\mathcal{R}}'^k}\right)^{-1}\widehat{\Delta\boldsymbol{r}}^k = \boldsymbol{W}^k\boldsymbol{c}^k + \boldsymbol{r}^k. \tag{18}$$

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.

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 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. 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. 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, but only on the coarsest grid level.

The numerical experiments in Section 4 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 $i+1$,

$$\Delta\boldsymbol{r}^{j-1} = \boldsymbol{\mathcal{R}}_{i+1}(\boldsymbol{d}^j) - \boldsymbol{\mathcal{R}}_i(\boldsymbol{d}^{j-1}) \tag{19a}$$

$$\Delta\tilde{\boldsymbol{d}}^{j-1} = \boldsymbol{\mathcal{S}}_{i+1}\circ\boldsymbol{\mathcal{F}}_{i+1}(\boldsymbol{d}^j) - \boldsymbol{\mathcal{S}}_i\circ\boldsymbol{\mathcal{F}}_i(\boldsymbol{d}^{j-1}), \tag{19b}$$

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

```
 1: $k = \ell = 0$
 2: $\boldsymbol{d}^k = \frac{5}{2}\boldsymbol{d}^n - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$
 3: $\tilde{\boldsymbol{d}}^k = \boldsymbol{\mathcal{S}}_1 \circ \boldsymbol{\mathcal{F}}_1(\boldsymbol{d}^k)$
 4: $\boldsymbol{r}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k$
 5: for $i = 1$ to $g$ do
 6:     while $||\boldsymbol{r}^k||_2 > \varepsilon_i$ or $\ell = 0$ do
 7:         $\ell = 1$
 8:         if $k = 0$ then
 9:             $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \omega \boldsymbol{r}^k$
10:         else
11:             construct $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$
12:             calculate QR-decomposition $\boldsymbol{V}^k = \boldsymbol{Q}^k \boldsymbol{R}^k$
13:             solve $\boldsymbol{R}^k \boldsymbol{c}^k = -{\boldsymbol{Q}^k}^{\mathrm{T}} \boldsymbol{r}^k$
14:             $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \boldsymbol{W}^k \boldsymbol{c}^k + \boldsymbol{r}^k$
15:         end if
16:         $\tilde{\boldsymbol{d}}^{k+1} = \boldsymbol{\mathcal{S}}_i \circ \boldsymbol{\mathcal{F}}_i(\boldsymbol{d}^{k+1})$
17:         $\boldsymbol{r}^{k+1} = \tilde{\boldsymbol{d}}^{k+1} - \boldsymbol{d}^{k+1}$
18:         $k = k + 1$
19:     end while
20:     $\ell = 0$
21: end for
22: for $i = 1$ to $g - 1$ do
23:     synchronize $\boldsymbol{\mathcal{F}}_i$ and $\boldsymbol{\mathcal{S}}_i$ with $\boldsymbol{\mathcal{F}}_g$ and $\boldsymbol{\mathcal{S}}_g$
24: end for
```

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. When the differences in Eqs. (19) are not used, the number of columns in $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ at the end of the time step is less than or equal to the number of coupling iterations minus the number of 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 stress 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

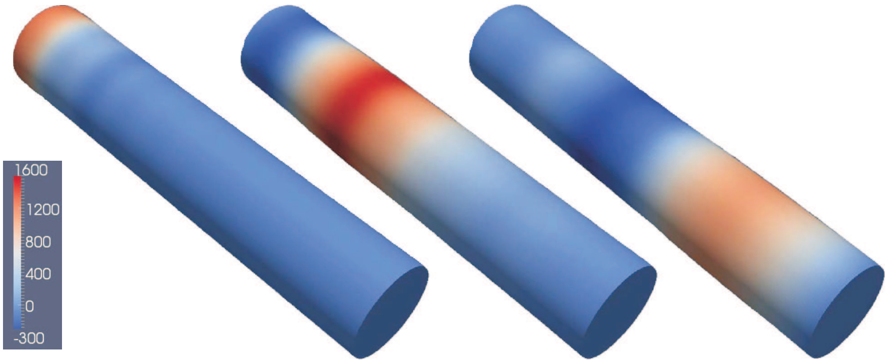

Figure 2: The pressure contours (in Pa) on the fluid-structure interface of the finest grid level for the propagation of a pressure wave in a 3D tube after 10^{-3} s (left), 5×10^{-3} s (centre) and 9×10^{-3} s (right).

and structural equations can be solved once more on all but the finest grid level, with this displacement and stress as boundary condition.

4 NUMERICAL RESULTS

As an example, the propagation of a pressure wave in a straight flexible tube is simulated [13]. This tube with radius 0.005 m and length 0.05 m is a simplified model for a large artery. The finite volume flow solver uses linear interpolation for the pressure and first-order upwind discretization 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. The grid of the fluid domain is adapted to the displacement of the fluid-structure interface by replacing the cell edges with springs. 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 Pa·s. 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. The pressure contours on the fluid-structure interface as shown in Figure 2 correspond well with those in [13].

For this simulation, two grid levels are used and the convergence tolerance is $\varepsilon_i = 10^{-3}||\boldsymbol{r}^0||_2$ for both grid levels. 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. IQN-ILS required on average 13.2 coupling iterations per time step on the fine grid, whereas ML-IQN-ILS performed on average 12.1 coupling iterations on the coarse grid and 7.0 on the fine grid. The number of coupling iterations on the fine grid is thus reduced by approximately 50 % in the simulation with two grid levels, compared to a simulation with a fine grid only. 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 %.

5 CONCLUSIONS

A new multi-level coupling technique for partitioned simulation of fluid-structure interaction has been presented. This technique is based on the fundamental insight from stability analyses on Gauss-Seidel coupling iterations that in the difference between the current and the correct interface displacement, the Fourier modes with a low wave number are most unstable. ML-IQN-ILS 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 numerical results show that this multi-level algorithm can reduce the duration of a partitioned fluid-structure interaction simulation.

ACKNOWLEDGMENT

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

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] A. van Zuijlen, S. Bosscher, H. Bijl, Two level algorithms for partitioned fluid-structure interaction computations, Computer Methods in Applied Mechanics and Engineering 196 (8) (2007) 1458–1470.

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

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