

# A multi-solver quasi-Newton method for the partitioned simulation of fluid-structure interaction

**J Degroote, S Annerel and J Vierendeels**

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

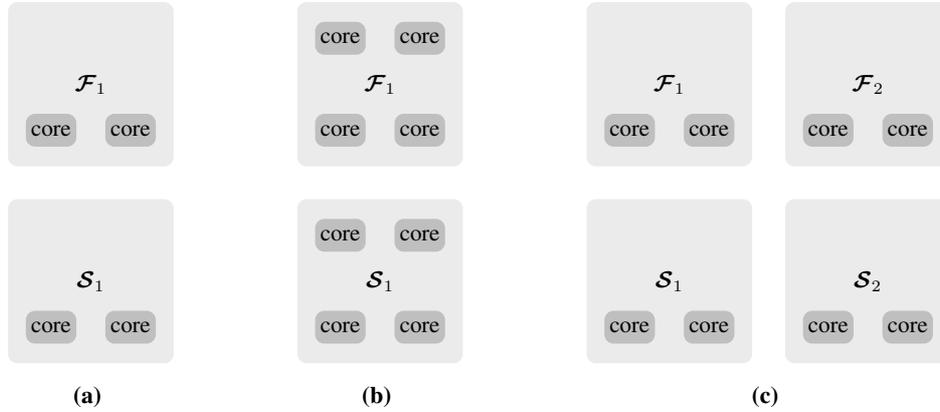
E-mail: {Joris.Degroote, Sebastiaan.Annerel, Jan.Vierendeels}@UGent.be

**Abstract.** In partitioned fluid-structure interaction simulations, the flow equations and the structural equations are solved separately. Consequently, the stresses and displacements on both sides of the fluid-structure interface are not automatically in equilibrium. Coupling techniques like Aitken relaxation and the Interface Block Quasi-Newton method with approximate Jacobians from Least-Squares models (IBQN-LS) enforce this equilibrium, even with black-box solvers. However, all existing coupling techniques use only one flow solver and one structural solver. To benefit from the large number of multi-core processors in modern clusters, a new Multi-Solver Interface Block Quasi-Newton (MS-IBQN-LS) algorithm has been developed. This algorithm uses more than one flow solver and structural solver, each running in parallel on a number of cores. One-dimensional and three-dimensional numerical experiments demonstrate that the run time of a simulation decreases as the number of solvers increases, albeit at a slower pace. Hence, the presented multi-solver algorithm accelerates fluid-structure interaction calculations by increasing the number of solvers, especially when the run time does not decrease further if more cores are used per solver.

## 1. Introduction

Fluid-structure interaction (FSI) is the mutual interaction between a fluid flow and a deforming structure. In monolithic fluid-structure interaction simulations, all governing equations are solved simultaneously. By contrast, in partitioned fluid-structure interaction simulations, the flow equations are solved by a flow solver and the structural equations are solved by a structural solver. When using this partitioned approach, the stresses and displacements on both sides of the fluid-structure interface are not automatically in equilibrium. Coupling techniques like Aitken relaxation [1], Interface-GMRES [2, 3], the Interface Block Quasi-Newton method with approximate Jacobians from Least-Squares models (IBQN-LS [4]) and the Interface Quasi-Newton technique with an approximation for the Inverse of the Jacobian from a Least-Squares model (IQN-ILS [5]) enforce this equilibrium, even with black-box solvers.

Fluid-structure interaction simulations can be computationally intensive. Nowadays, mostly clusters are used for high-performance computing (HPC). These computers consist of a large number of cluster nodes, each containing a small number of multi-core processors and an amount of memory. Although fast interconnects between the cluster nodes exist, they are still slower than the communication lines inside the cluster nodes. By running the flow solver and the structural solver in parallel, i.e. on more than one core of the cluster, a fluid-structure interaction calculation can generally be accelerated. Optimally, the speed-up from parallelization is linear: doubling the number of cores should halve the calculation's duration, and doubling it a second time should again halve the duration. However, very few parallel codes achieve optimal speed-up. Most of them have a near-linear speed-up for small numbers of cores,



**Figure 1:** In (a), the flow solver  $\mathcal{F}_1$  and structural solver  $\mathcal{S}_1$  each run on two cores. With twice as many cores, either (b) the number of cores per solver can be increased to four or (c) an additional flow solver  $\mathcal{F}_2$  and structural solver  $\mathcal{S}_2$  can be used.

which flattens out into a constant value or even decreases for large numbers of cores. The end of the near-linear speed-up depends on several factors, such as the interconnect, the parallelization technique, the time required for communication compared to the time required for computation, the number of degrees of freedom, etc.

Fluid-structure interaction simulations are often unsteady calculations with a large number of time steps that have to be calculated consecutively and a modest number of degrees of freedom to current standards. As mentioned above, the near-linear speed-up ends at a certain number of cores. At that point, increasing the number of cores per solver no longer leads to a reduction of the calculation time. However, it is also possible to increase the number of flow solvers and structural solvers, while keeping the number of cores per solver constant. Figure 1 illustrates this multi-solver approach.

The new idea of using more than one flow solver and more than one structural solver can be applied to several existing coupling techniques. This paper presents the Multi-Solver Interface Block Quasi-Newton technique with approximate Jacobians from Least-Squares models (MS-IBQN-LS) for the partitioned simulation of strongly coupled fluid-structure interaction problems. The new multi-solver algorithm is derived from the Interface Block Quasi-Newton technique with approximate Jacobians from Least-Squares models (IBQN-LS [4]). This new multi-solver algorithm uses more than one flow solver and more than one structural solver at the same time, as opposed to all existing partitioned algorithms which use only one solver of each type. In this multi-solver algorithm, data from the least-squares models in the previous time step is recalculated in the current time step by the additional flow solvers and structural solvers. A similar approach can be used to construct a multi-solver version of IQN-ILS.

After the necessary definitions in Section 2, Section 3 briefly summarizes how a least-squares approximation for the Jacobian of a black-box solver is constructed in the standard IBQN-LS technique. Section 4 then explains how data from the least-squares models can be recalculated by additional solvers, followed by the numerical experiments in Section 5.

## 2. Definitions

The function

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) \quad (1)$$

is referred to as the flow solver and it concisely represents the following operations. The displacement  $\mathbf{x} \in \mathbb{R}^u$  of the fluid-structure interface with respect to the initial position is given to the flow code.

Subsequently, the flow equations are solved for the fluid state in the entire fluid domain, which also results in a stress distribution  $\mathbf{y} \in \mathbb{R}^w$  on the interface.

The structural solver is represented by the function

$$\mathbf{x} = \mathcal{S}(\mathbf{y}). \quad (2)$$

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

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

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

In the remainder of this paper, all values and functions are at the new time level  $n + 1$ , unless indicated otherwise. A superscript  $k$  indicates the coupling iteration within time step  $n + 1$ . 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. The coupling iterations in a time step begin from an extrapolation of the interface's position based on the previous time steps, given by

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

if the time step is constant. Lower order extrapolations are used for the first two time steps.

### 3. Interface Block Quasi-Newton technique with approximate Jacobians from Least-Squares models (IBQN-LS)

The IBQN-LS method solves the fluid-structure interaction problem written as

$$\begin{cases} \mathcal{F}(\mathbf{x}) - \mathbf{y} = \mathbf{0} \\ \mathcal{S}(\mathbf{y}) - \mathbf{x} = \mathbf{0} \end{cases} \quad (5)$$

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

$$\begin{bmatrix} \widehat{\mathcal{F}'} & -I \\ -I & \widehat{\mathcal{S}'} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathcal{F}(\mathbf{x}) - \mathbf{y} \\ \mathcal{S}(\mathbf{y}) - \mathbf{x} \end{bmatrix} \quad (6)$$

is thus first solved for  $\Delta \mathbf{x}$ , followed by an update of  $\mathbf{x}$  and the right-hand side. Subsequently, the modified system is solved for  $\Delta \mathbf{y}$  and afterwards  $\mathbf{y}$  is updated. In agreement with the notation for intermediate values defined in Section 2, the input and output of the flow solver are denoted as  $\mathbf{x}^k$  and  $\tilde{\mathbf{y}}^{k+1}$  and the input and output of the structural solver as  $\mathbf{y}^{k+1}$  and  $\tilde{\mathbf{x}}^{k+1}$ .

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

$$\left( I - \widehat{\mathcal{S}'^k} \widehat{\mathcal{F}'^k} \right) \Delta \mathbf{x}^k = \tilde{\mathbf{x}}^{k+1} - \mathbf{x}^k + \widehat{\mathcal{S}'^k} (\tilde{\mathbf{y}}^{k+1} - \mathbf{y}^{k+1}) \quad (7)$$

for  $\Delta \mathbf{x}^k$ . The prime denotes the Jacobian matrix of a function. As opposed to the original approach in reference [4], this linear system is solved in a matrix-free way with an iterative Krylov solver like the generalized conjugate residual (GCR) method [6]. The matrix on the left-hand side of Eq. (7) and thus

the approximate Jacobians  $\widehat{\mathcal{F}}^{i^k}$  and  $\widehat{\mathcal{S}}^{i^k}$  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  $\widehat{\mathcal{F}}^{i^k}$  or  $\widehat{\mathcal{S}}^{i^k}$  is now explained. The matrix-vector product with  $\widehat{\mathcal{F}}^{i^k}$  is calculated from the previous inputs

$$\mathbf{x}^0, \dots, \mathbf{x}^k \quad (8a)$$

and the corresponding outputs

$$\tilde{\mathbf{y}}^1 = \mathcal{F}(\mathbf{x}^0), \dots, \tilde{\mathbf{y}}^{k+1} = \mathcal{F}(\mathbf{x}^k). \quad (8b)$$

of the flow solver. These data are converted into differences (also called modes) with respect to some reference, in this case the first input and output in the current time step

$$\Delta \mathbf{x}^i = \mathbf{x}^{i+1} - \mathbf{x}^0 \quad (9a)$$

$$\Delta \tilde{\mathbf{y}}^{i+1} = \tilde{\mathbf{y}}^{i+2} - \tilde{\mathbf{y}}^1 \quad (9b)$$

for  $0 = 1, \dots, k-1$ . These differences are stored as the columns of the matrices

$$\underline{\mathbf{V}}_f^k = [\Delta \mathbf{x}^{k-1} \quad \Delta \mathbf{x}^{k-2} \quad \dots \quad \Delta \mathbf{x}^1 \quad \Delta \mathbf{x}^0] \quad (10a)$$

and

$$\underline{\mathbf{W}}_f^k = [\Delta \tilde{\mathbf{y}}^k \quad \Delta \tilde{\mathbf{y}}^{k-1} \quad \dots \quad \Delta \tilde{\mathbf{y}}^2 \quad \Delta \tilde{\mathbf{y}}^1], \quad (10b)$$

with the subscript  $f$  referring to the flow solver. Not only differences obtained during the coupling iterations in the current time step can be used but also differences obtained in the previous time step(s) can be reused. If only differences from the previous time step are reused, the matrices  $\underline{\mathbf{V}}_f^k$  and  $\underline{\mathbf{W}}_f^k$  have to be combined with those from the previous time step (if a previous time step has already been performed), giving

$$\mathbf{V}_f^k = [\underline{\mathbf{V}}_f^k \quad \underline{\mathbf{V}}_f^n] \quad (11a)$$

and

$$\mathbf{W}_f^k = [\underline{\mathbf{W}}_f^k \quad \underline{\mathbf{W}}_f^n]. \quad (11b)$$

A similar approach has been used in references [2, 3] to reuse Krylov vectors. Subsequently, the economy-size QR-decomposition of  $\mathbf{V}_f^k$  is calculated. To determine the product of  $\widehat{\mathcal{F}}^{i^k}$  with a vector  $\Delta \mathbf{x}$ , the triangular system

$$\mathbf{R}_f^k \mathbf{c}_f^k = \mathbf{Q}_f^{kT} \Delta \mathbf{x} \quad (12)$$

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

$$\widehat{\mathcal{F}}^{i^k} \Delta \mathbf{x} = \mathbf{W}_f^k \mathbf{c}_f^k. \quad (13)$$

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

Once  $\mathbf{x}^{k+1}$  has been obtained,  $k$  is increased. Then, the corresponding stress distribution  $\tilde{\mathbf{y}}^{k+1} = \mathcal{F}(\mathbf{x}^k)$  is calculated and the matrices  $\mathbf{V}_f^k$ ,  $\mathbf{W}_f^k$ ,  $\mathbf{Q}_f^k$  and  $\mathbf{R}_f^k$  are updated. To calculate the stress distribution  $\mathbf{y}^{k+1} = \mathbf{y}^k + \Delta \mathbf{y}^k$  that has to be applied on the structure, the system

$$\left( \mathbf{I} - \widehat{\mathcal{F}}^{i^k} \widehat{\mathcal{S}}^{i^{k-1}} \right) \Delta \mathbf{y}^k = \tilde{\mathbf{y}}^{k+1} - \mathbf{y}^k + \widehat{\mathcal{F}}^{i^k} (\tilde{\mathbf{x}}^k - \mathbf{x}^k) \quad (14)$$

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

For changes of  $\mathbf{x}$  and  $\mathbf{y}$  that can be written as a linear combination of the columns in the matrices  $\mathbf{V}_f^k$  and  $\mathbf{V}_s^k$ , the interaction between the fluid and the structure is treated implicitly during the coupling iterations. In reference [7], it is shown that only a fraction of the Fourier modes in  $\mathbf{x}$  and  $\mathbf{y}$  needs implicit treatment of the interaction during the coupling iterations. Hence, the IBQN-LS iterations converge once those modes are included in  $\mathbf{V}_f^k$  and  $\mathbf{V}_s^k$ .

#### 4. Multi-Solver Interface Block Quasi-Newton technique with approximate Jacobians from Least-Squares models (MS-IBQN-LS)

In the IBQN-LS algorithm with reuse of data from the previous time step, the approximation for a matrix-vector product uses differences obtained during the coupling iterations in the current time step combined with differences obtained during the coupling iterations in the previous time step, as show in Eqs. (11). However, the data from the previous time step are only approximately correct for the current time step, even though the reuse of data from previous time steps results in faster convergence of the coupling iterations in most numerical experiments [8]. Cases with large differences between the time steps, for example, do not benefit from this reuse.

The relation between the columns of  $\mathbf{V}_f^n$  and  $\mathbf{W}_f^n$  is only approximate at  $t^{n+1}$ . Nevertheless, the columns of  $\mathbf{V}_f^n$  can be used to determine which modes should be treated implicitly during the coupling iterations at  $t^{n+1}$  to obtain fast convergence of the coupling iterations. The response of a solver at  $t^{n+1}$  to the modes that have been treated implicitly during the coupling iterations at  $t^n$  can be calculated exactly by applying these modes again at  $t^{n+1}$ . Moreover, the recalculation of modes from the previous time step can be done in parallel with normal coupling iterations if  $g > 1$  flow solvers and  $h > 1$  structural solvers are used. In this section, a subscript  $i$  distinguishes the different solvers and their respective input and output.

Algorithm 1 on page 6 describes the Multi-Solver IBQN-LS (MS-IBQN-LS) algorithm with parallel recalculation of modes. For this algorithm, the number of flow solvers and structural solvers does not have to be the same. Solvers  $\mathcal{F}_1$  and  $\mathcal{S}_1$  calculate the solution of the coupled problem with the standard IBQN-LS algorithm, while solvers  $\mathcal{F}_i$  and  $\mathcal{S}_j$  ( $i = 2, \dots, g$  and  $j = 2, \dots, h$ ) recalculate modes from the previous time step. Lines 10 to 19 and lines 22 to 30 describe the standard IBQN-LS algorithm, with the exception of the 'start' on line 19 and line 30. This command means that the calculation has to be started, without waiting for the result to continue the execution of the algorithm. On line 7, the coupling algorithm checks whether  $\mathcal{F}_1$  and  $\mathcal{S}_1$  are 'ready', i.e. whether  $\mathcal{F}_1$  has completed its previous calculation and  $\mathcal{S}_1$  can begin the following calculation or vice versa. The variable  $\ell$  alternates between 0 and 1 to ensure that  $\mathcal{F}_1$  and  $\mathcal{S}_1$  take turns.

Because the coupling code is not waiting on line 19 and line 30 until  $\mathcal{F}_1$  and  $\mathcal{S}_1$  are ready, it can control the other solvers in the meantime. On line 33 and line 40, the coupling algorithm loops over the additional solvers  $\mathcal{F}_i$  and  $\mathcal{S}_j$  ( $i = 2, \dots, g$  and  $j = 2, \dots, h$ ). In a first step of the recalculation of modes by solver  $\mathcal{F}_i$ , a column  $\Delta\mathbf{x}$  of  $\mathbf{V}^n$  and the corresponding column  $\Delta\tilde{\mathbf{y}}$  of  $\mathbf{W}^n$  are selected. In this case, these columns are chosen in the order in which they were created, meaning the oldest one first. As a result, the rightmost columns of  $\mathbf{V}^n$  and  $\mathbf{W}^n$  are selected first for  $\mathcal{F}_2$ , followed by the second rightmost columns for  $\mathcal{F}_3$ , etc. Other selection procedures are also possible: newest first, largest  $\|\Delta\mathbf{x}\|_2$ , largest  $\|\Delta\tilde{\mathbf{y}}\|_2/\|\Delta\mathbf{x}\|_2$ , etc.

Subsequently, the selected  $\Delta\mathbf{x}$  and  $\Delta\tilde{\mathbf{y}}$  from  $t^n$  have to be recalculated at  $t^{n+1}$ . These vectors are differences with respect to the reference vectors  $\mathbf{x}_1^{n,0}$  and  $\tilde{\mathbf{y}}_1^{n,1}$  which originate from the first coupling iteration at  $t^n$ . To be recalculated at  $t^{n+1}$ , the vector  $\Delta\mathbf{x}$  has to be added to a reference at  $t^{n+1}$ . Here, the vectors  $\mathbf{x}_1^0$  and  $\tilde{\mathbf{y}}_1^1$  are used as reference at  $t^{n+1}$ . These vectors are calculated by  $\mathcal{F}_1$  during the first coupling iteration, so they are the first  $\mathbf{x}$  and  $\tilde{\mathbf{y}}$  at  $t^{n+1}$ . As a result, the recalculation of information can begin after the first coupling iteration between  $\mathcal{F}_1$  and  $\mathcal{S}_1$ .

---

**Algorithm 1** The Multi-Solver Interface Block Quasi-Newton technique with approximate Jacobians from Least-Squares models (MS-IBQN-LS) (Part 1).

---

```

1:  $k = 0; \ell = 0$ 
2:  $\tilde{\mathbf{y}}_1^1 = \mathcal{F}_1(\mathbf{x}_1^0)$ 
3:  $\mathbf{y}_1^1 = \tilde{\mathbf{y}}_1^1$ 
4:  $\tilde{\mathbf{x}}_1^1 = \mathcal{S}_1(\mathbf{y}_1^1)$ 
5:  $\mathbf{r}_1^0 = \tilde{\mathbf{x}}_1^1 - \mathbf{x}_1^0$ 
6: while  $\|\mathbf{r}_1^k\|_2 > \varepsilon_o$  do
7:   if  $\mathcal{F}_1$  and  $\mathcal{S}_1$  are ready then
8:     if  $\ell = 0$  then
9:        $\ell = 1$ 
10:      if  $k = 0$  and ( $q = 0$  or  $n = 0$ ) then
11:         $\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + \omega \mathbf{r}_1^k$ 
12:      else
13:        construct  $\mathbf{V}_s^k$  and  $\mathbf{W}_s^k$ 
14:        calculate QR-decomposition  $\mathbf{V}_s^k = \mathbf{Q}_s^k \mathbf{R}_s^k$ 
15:        calculate  $\Delta \mathbf{x}^k$ 
16:         $\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + \Delta \mathbf{x}^k$ 
17:      end if
18:       $k = k + 1$ 
19:      start  $\tilde{\mathbf{y}}_1^{k+1} = \mathcal{F}_1(\mathbf{x}_1^k)$ 
20:    else
21:       $\ell = 0$ 
22:      if  $k = 1$  and ( $q = 0$  or  $n = 0$ ) then
23:         $\mathbf{y}_1^{k+1} = \tilde{\mathbf{y}}_1^{k+1}$ 
24:      else
25:        construct  $\mathbf{V}_f^k$  and  $\mathbf{W}_f^k$ 
26:        calculate QR-decomposition  $\mathbf{V}_f^k = \mathbf{Q}_f^k \mathbf{R}_f^k$ 
27:        calculate  $\Delta \mathbf{y}^k$ 
28:         $\mathbf{y}_1^{k+1} = \mathbf{y}_1^k + \Delta \mathbf{y}^k$ 
29:      end if
30:      start  $\mathbf{r}_1^k = \mathcal{S}_1(\mathbf{y}_1^{k+1}) - \mathbf{x}_1^k$ 
31:    end if
32:  end if
33:  for  $i = 2$  to  $g$  do
34:    if  $\mathcal{F}_i$  is ready then
35:      select  $\Delta \mathbf{x}$ 
36:       $\mathbf{x}_i = \mathbf{x}_1^0 + \Delta \mathbf{x}$ 
37:      start  $\tilde{\mathbf{y}}_i = \mathcal{F}_i(\mathbf{x}_i)$ 
38:    end if
39:  end for
40:  for  $i = 2$  to  $h$  do
41:    if  $\mathcal{S}_i$  is ready then
42:      select  $\Delta \mathbf{y}$ 
43:       $\mathbf{y}_i = \mathbf{y}_1^1 + \Delta \mathbf{y}$ 
44:      start  $\tilde{\mathbf{x}}_i = \mathcal{S}_i(\mathbf{y}_i)$ 
45:    end if
46:  end for
47: end while

```

---

---

**Algorithm 1** Multi-Solver Interface Block Quasi-Newton technique with approximate Jacobians from Least-Squares models (MS-IBQN-LS) (Part 2).

---

```

48: for  $i = 2$  to  $g$  do
49:   start synchronizing  $\mathcal{F}_i$  with  $\mathcal{F}_1$ 
50: end for
51: for  $i = 2$  to  $h$  do
52:   start synchronizing  $\mathcal{S}_i$  with  $\mathcal{S}_1$ 
53: end for

```

---

So, the input for  $\mathcal{F}_i$  is given by

$$\mathbf{x}_i = \mathbf{x}_1^0 + \Delta \mathbf{x}. \quad (15)$$

The coupling code checks on line 34 whether  $\mathcal{F}_i$  has completed its calculation. When this is the case, both the displacement  $\mathbf{x}_i$  and the corresponding stress distribution  $\tilde{\mathbf{y}}_i$  are known. By subtracting the references, respectively  $\mathbf{x}_1^0$  and  $\tilde{\mathbf{y}}_1^1$ , a new mode of the flow solver in the current time step becomes available.

$$\Delta \mathbf{x} = \mathbf{x}_i - \mathbf{x}_1^0 \quad (16a)$$

$$\Delta \tilde{\mathbf{y}} = \tilde{\mathbf{y}}_i - \tilde{\mathbf{y}}_1^1 \quad (16b)$$

All modes from the previous time step that have been recalculated are stored as columns of the matrices  $\underline{\mathbf{V}}_f^n$  and  $\underline{\mathbf{W}}_f^n$ , instead of using the modes calculated in the previous time step. These matrices are then combined with  $\underline{\mathbf{V}}_f^k$  and  $\underline{\mathbf{W}}_f^k$ , which contain the modes calculated by  $\mathcal{F}_1$ , to form  $\mathbf{V}_f^k$  and  $\mathbf{W}_f^k$  as

$$\mathbf{V}_f^k = [\underline{\mathbf{V}}_f^k \quad \underline{\mathbf{V}}_f^n] \quad (17a)$$

and

$$\mathbf{W}_f^k = [\underline{\mathbf{W}}_f^k \quad \underline{\mathbf{W}}_f^n]. \quad (17b)$$

Subsequently, the QR-decomposition  $\mathbf{V}_f^k = \mathbf{Q}_f^k \mathbf{R}_f^k$  is calculated, as indicated on line 26. The columns of  $\underline{\mathbf{V}}_f^k$  and  $\underline{\mathbf{W}}_f^k$  contain the modes calculated by  $\mathcal{F}_1$  in the current time step. Data from the previous time step that have not been recalculated (yet) can be included in the matrices  $\underline{\mathbf{V}}_f^n$  and  $\underline{\mathbf{W}}_f^n$  as well, but this is not done in this paper. In that case, the old data should be removed once it has been recalculated so that it is not present twice. Because no information from the previous time step that has not been recalculated at  $t^{n+1}$  is included in  $\mathbf{V}_f^k$  and  $\mathbf{W}_f^k$ , these matrices are empty at the beginning of each time step. An analogous procedure is followed for the structural solvers.

All modes calculated by  $\mathcal{F}_1$  and  $\mathcal{S}_1$  in the current time step and all modes that have been recalculated in the current time step are candidates to be recalculated in the following time step. Two mechanisms avoid an ever increasing number of modes. The first one is that once  $\mathcal{F}_1$  and  $\mathcal{S}_1$  have found the correct solution, the coupling discards old modes that have not been recalculated at  $t^{n+1}$ . However, before discarding these modes, the coupling waits until all solvers that are recalculating data have completed their current calculation. It is of course possible to select a wider window for the modes that can be recalculated, for example modes that have been calculated in the last two or three time steps. The second mechanism is a tolerance  $\varepsilon_s$  for the detection of small diagonal elements in  $\mathbf{R}_{f,s}^k$ . If a small diagonal element is detected, the corresponding columns in  $\mathbf{V}_{f,s}^k$  and  $\mathbf{W}_{f,s}^k$  are removed, which means that the column cannot be recalculated in the following time step.

The matrices  $\mathbf{V}_{s,f}^k$  and  $\mathbf{W}_{s,f}^k$  have to contain at least one column to calculate the quasi-Newton update; otherwise a relaxation with factor  $\omega$  is used for the interface's displacement (line 11) and the stress distribution is passed on without modification (line 23).

A last important aspect of the MS-IBQN-LS algorithm is the synchronization of the solvers  $\mathcal{F}_i$  and  $\mathcal{S}_j$  ( $i = 2, \dots, g$  and  $j = 2, \dots, h$ ) with  $\mathcal{F}_1$  and  $\mathcal{S}_1$ , as mentioned on lines 48 to 53. At the end of each time step, the values of the degrees of freedom inside the fluid and solid domain have to be the same in all solvers. Otherwise, unphysical results will be obtained in the following time step. The different solvers have to receive identical initial conditions for the next time step, which implies that the solution at the end of a time step should be the same in each of the solvers. For example, without synchronization, the stress distribution on the interface for a given displacement would depend on which flow solver is used ( $\mathcal{F}_i(\mathbf{x}) \neq \mathcal{F}_j(\mathbf{x})$  if  $i \neq j$ ) because the solution at  $t^n$  influences the solution at  $t^{n+1}$ . These difficulties can be avoided by copying the degrees of freedom in the entire fluid and solid domain from  $\mathcal{F}_1$  and  $\mathcal{S}_1$  to all other flow solvers and structural solvers, once the coupling iterations have converged. If the implementation does not allow to copy the degrees of freedom from one solver to another one or to read a file with all values from another solver, the same result can be obtained by solving the equations once more in  $\mathcal{F}_i$  and  $\mathcal{S}_j$  ( $i = 2, \dots, g$  and  $j = 2, \dots, h$ ) with  $\mathbf{x}_1^{last}$  and  $\mathbf{y}_1^{last}$  as input. This implementation will of course reduce the gain of the parallel recalculation of modes.

The synchronization of the different solvers can also be done in parallel. The coupling code does not have to wait until one solver has completed the synchronization to start synchronizing the following solver. The coupling code does not even have to wait until all synchronizations are complete at the end of the time step. However, the synchronization has to be completed before the solver can start to recalculate modes in the following time step. Therefore, the ‘ready’ on line 34 and line 41 also means that the synchronization of the solvers has to be completed. In practice, the synchronization of the additional solvers is performed at the same time as the first coupling iteration between  $\mathcal{F}_1$  and  $\mathcal{S}_1$ .

## 5. Numerical experiments

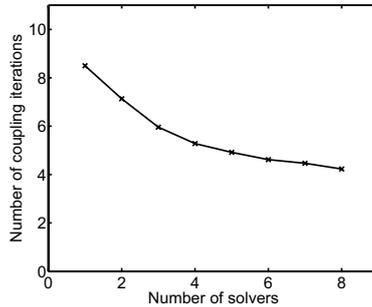
All numerical experiments in this section have been performed on dedicated cluster nodes, each with two Intel Xeon X5355 2.66 GHz quad-core processors and 8 GB of working memory. Both one-dimensional and three-dimensional simulations of the flow in a tube are presented.

### 5.1. 1D Tube

The first numerical experiment with the MS-IBQN-LS algorithm is the simulation of the unsteady flow in a one-dimensional flexible tube, without inertia in the structure. The governing equations and the discretization of this problem are described in reference [7]. In the simulations presented in this section, the tube consists of  $N = 1000$  segments, the dimensionless stiffness is  $\kappa = 10$  and the dimensionless time step is  $\tau = 0.01$ . The initial conditions are a dimensionless velocity of  $v_o = 0.1$ , a dimensionless cross-sectional area of  $a_o = 1$  and a dimensionless pressure of  $p_o = 0$ . At the inlet, a sinusoidally varying velocity is imposed while a non-reflecting boundary condition is applied at the outlet. One period of the inlet boundary condition is simulated.

Figure 2 depicts the average number of coupling iterations between  $\mathcal{F}_1$  and  $\mathcal{S}_1$  per time step for the MS-IBQN-LS algorithm. The convergence criterion for the coupling iterations is  $\|\mathbf{r}_1^k\|_2 < 10^{-5} \|\mathbf{r}_1^0\|_2$ . Although it is possible to have a different number of flow solvers and structural solvers in the MS-IBQN-LS algorithm, this option is not used. The MS-IBQN-LS algorithm with eight flow solvers and eight structural solvers requires only half of the number of coupling iterations per time step compared to the standard IBQN-LS algorithm. If the duration of the communication and synchronization is negligible compared to the duration of the calculation, this will result in a reduction of the run time by almost 50%. Not more than eight solvers of each type have been used because the curves flatten out as the number of solvers increases.

As opposed to most other coupling algorithms, multi-solver algorithms will need a slightly different number of coupling iterations each time the same simulation is performed. The parallel recalculation involves ‘start’ and ‘ready’ commands, so the order of the various calculations can change. Depending on whether a recalculated mode becomes available before or after  $\mathcal{F}_1$  and  $\mathcal{S}_1$  start with a new calculation, the convergence will be faster or slower. Therefore, all simulations have been performed 100 times. The



**Figure 2:** The average number of coupling iterations between  $\mathcal{F}_1$  and  $\mathcal{S}_1$  per time step for the flow in a 1D flexible tube using the MS-IBQN-LS algorithm as a function of the number of solvers. The same number of flow solvers and structural solvers has been used and each solver uses one core.

**Table 1:** The average number of coupling iterations per time step and the relative duration for the propagation of a pressure wave in a 3D flexible tube. The notation MS-IBQN-LS( $g, h$ ) denotes that  $g$  flow solvers and  $h$  structural solvers are used.

Algorithm	Iterations	Duration
IBQN-LS	7.9	1.52
MS-IBQN-LS(4,4)	4.8	1.00

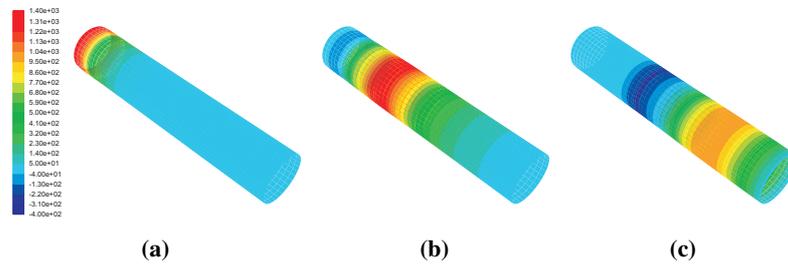
number of coupling iterations per time step has first been averaged over all time steps in a simulation with a given number of solvers and over all 100 runs of that simulation. The difference between the average number of coupling iterations per time step in two different runs with the same number of solvers was never more than one iteration.

### 5.2. 3D Tube

The second numerical experiment is the propagation of a pressure wave in a three-dimensional straight flexible tube [9, 10]. The finite volume flow solver uses second-order discretization for the pressure and first-order upwind for the momentum. It solves the Navier-Stokes equations in arbitrary Lagrangian-Eulerian (ALE) formulation with the PISO scheme and first-order backward Euler time integration. The grid of the fluid domain is adapted to the displacement of the fluid-structure interface with a spring analogy. 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 has a length of 0.05 m and an inner radius of 0.005 m. The material of its wall is a linear elastic with density  $1200 \text{ kg/m}^3$ , Young's modulus  $3 \times 10^5 \text{ N/m}^2$ , 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  $1000 \text{ kg/m}^3$  and a viscosity of 0.003 Pas. Both the fluid and the structure are initially at rest. During the first  $3 \times 10^{-3} \text{ s}$ , an overpressure of  $1333.2 \text{ N/m}^2$  is applied at the inlet. The wave propagates through the tube during  $10^{-2} \text{ s}$ , simulated with time steps of  $10^{-4} \text{ s}$ . Pressure contours on the fluid-structure interface are shown in Figure 3 and they correspond well with those in [9, 10].

Table 1 lists the number of coupling iterations per time step, averaged over the entire simulation, as well as the relative duration of the simulations. For this test case, the simulation has been performed



**Figure 3:** The pressure contours (in Pa) on the fluid-structure interface for the propagation of a pressure wave in a 3D flexible tube after (a)  $10^{-3}$  s; (b)  $5 \times 10^{-3}$  s; (c)  $9 \times 10^{-3}$  s.

only once with each coupling algorithm. The convergence criterion for the coupling iterations is  $\|\mathbf{r}_1^k\|_2 < 10^{-3} \|\mathbf{r}_1^0\|_2$ . With four flow solvers and four structural solvers, the MS-IBQN-LS algorithm is approximately 30 % faster than the standard IBQN-LS algorithm. The reduction of the number of coupling iterations can be considered as a quality measure for the coupling algorithm while the reduction of the simulation's duration can be seen as a quality measure for the implementation. In this calculation, each flow solver runs on six cores and each structural solver on two cores.

## 6. Conclusions

More than one flow solver and more than one structural solver can be used in coupling algorithms for the partitioned simulation of fluid-structure interaction. This new idea has been used to develop the Multi-Solver Interface Block Quasi-Newton algorithm with approximate Jacobians from Least-Squares models (MS-IBQN-LS). This multi-solver algorithm uses additional solvers to recalculate modes from the previous time level at the current time level. It has been demonstrated that this new algorithm can reduce the duration of a simulation. However, the MS-IBQN-LS algorithm requires significantly more cores than the IBQN-LS algorithm for a comparatively small reduction of the duration of the simulation, resulting in a low efficiency. Therefore, the MS-IBQN-LS algorithm should only be applied when the duration of the simulation cannot be reduced any further by increasing the number of cores allocated to a single flow solver and a single structural solver.

## Acknowledgments

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

## References

- [1] Küttler U and Wall W 2008 *Comput. Mech.* **43** 61–72
- [2] Michler C, van Brummelen E and de Borst R 2005 *Int. J. Numer. Meth. Fluid.* **47** 1189–95
- [3] Michler C, van Brummelen E and de Borst R 2006 *Comput. Meth. Appl. Mech. Eng.* **195** 2124–48
- [4] Vierendeels J, Lanoye L, Degroote J and Verdonck P 2007 *Comput. Struct.* **85** 970–6
- [5] Degroote J, Bathe K J and Vierendeels J 2009 *Comput. Struct.* **87** 793–801
- [6] Eisenstat S, Elman H and Schultz M 1983 *SIAM J. Numer. Anal.* **20** 345–57
- [7] Degroote J, Bruggeman P, Haelterman R and Vierendeels J 2008 *Comput. Struct.* **86** 2224–34
- [8] Degroote J, Haelterman R, Annerel S, Bruggeman P and Vierendeels J 2009 *Comput. Struct.* **88** 446–57
- [9] Fernandez M and Moubachir M 2005 *Comput. Struct.* **83** 127–42
- [10] Gerbeau J F and Vidrascu M 2003 *ESAIM Math. Model. Numer. Anal.* **37** 631–48