Computers and Structures 87 (2009) 793–801

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction

Joris Degroote [a,*], Klaus-Jürgen Bathe [b], Jan Vierendeels [a]

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

[b] *Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge MA 02139-4307, USA*

ARTICLE INFO

Article history:
Received 13 October 2008
Accepted 26 November 2008
Available online 8 January 2009

Keywords:
Fluid–structure interaction
Partitioned
Quasi-Newton
Monolithic
Newton–Raphson

ABSTRACT

Fluid–structure interaction (FSI) can be simulated in a monolithic way by solving the flow and structural equations simultaneously and in a partitioned way with separate solvers for the flow equations and the structural equations. A partitioned quasi-Newton technique which solves the coupled problem through nonlinear equations corresponding to the interface position is presented and its performance is compared with a monolithic Newton algorithm. Various structural configurations with an incompressible fluid are solved, and the ratio of the time for the partitioned simulation, when convergence is reached, to the time for the monolithic simulation is found to be between 1/2 and 4. However, in this comparison of the partitioned and monolithic simulations, the flow and structural equations have been solved with a direct sparse solver in full Newton–Raphson iterations, only relatively small problems have been solved and this ratio would likely change if large industrial problems were considered or if other solution strategies were used.

1. Introduction

The simulation of fluid–structure interaction (FSI) and other coupled problems has gained in importance over the last decade. Nowadays, FSI simulations are used to avoid flutter of airplanes [1] and turbomachinery [2] and to study wave-energy converters [3]. Parachute dynamics is another interesting field as large deformations appear [4]. The design and analysis of artificial heart valves [5,6], the prediction of the rupture of aneurysms or of the outcome of surgery [7–10] also rely extensively on FSI simulations in patient-specific geometries.

An FSI problem can be solved in a monolithic or partitioned way. In the monolithic approach, the flow equations and structural equations are solved simultaneously such that their mutual influence can be taken into account directly which is favorable for the stability of the calculation. On the other hand, the partitioned approach is to solve the flow equations and the structural equations separately which means that the flow does not change while the solution of the structural equations is calculated and vice versa. The partitioned approach thus requires a coupling algorithm to allow for the interaction and to determine the solution of the coupled problem but software modularity is preserved and different, possibly more efficient solution techniques can be used for the flow equations and structural equations.

Newton methods can be used to solve the flow equations and structural equations in the entire fluid and solid domain with Newton–Raphson iterations. These methods have been applied in monolithic codes [11–15] but also in partitioned implementations with a matrix-free solver and finite-difference approximations in the calculation of the matrix–vector product [16,17].

The FSI problem can be reformulated as a problem – generally nonlinear – in the discrete position of the fluid–structure interface only, with all other variables internal to the residual operator of this problem. Interface-Newton methods solve these nonlinear equations with Newton–Raphson iterations while the flow equations and the structural equations can be solved with any method inside the residual operator. Gerbeau et al. [18,19] utilize an approximation of the Jacobian from a linear reduced-physics model whereas Van Brummelen et al. [20–22] employ a matrix-free Krylov solver for the linear system within each Newton–Raphson iteration combined with an approximation of the Jacobian-vector product based on a linear combination of the previous residual vectors.

Gauss–Seidel iterations between the flow solver and structural solver can also be used to solve this equation for the interface position in a partitioned way, but these iterations converge slowly or diverge if the interaction between the fluid and the structure is strong due to a high fluid/structure density ratio or an incompressible fluid [23,24]. The convergence can however be accelerated by Aitken's Δ^2 method [25]. If the interaction between the fluid and

* Corresponding author.
E-mail addresses: Joris.Degroote@UGent.be (J. Degroote), kjb@mit.edu (K.J. Bathe), Vierendeels@UGent.be (J. Vierendeels).
URL: http://www.FloHeaCom.UGent.be/ (J. Degroote).

0045-7949/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2008.11.013

the structure is weak, e.g. in aeroelastic simulations, only one coupling iteration is required within each time step [1,26–28] but these so-called staggered or loosely-coupled methods do not enforce the equilibrium conditions on the fluid–structure interface within each time step.

The FSI problem can also be written as a nonlinear problem in both the discrete interface position and the stress on the interface, with again all other variables internal to the residual operator of the problem. This system of equations has been solved with an interface-block-quasi-Newton technique with two least-squares reduced-order models (IBQN–LS) [29,30]. The reduced-order models and approximate Jacobians of the "black-box" flow solver and the "black-box" structural solver are constructed with the information on the fluid–structure interface during all the previous iterations in the time step or even information from previous time steps.

In this work, the technique to create an approximate Jacobian of a black-box solver from Ref. [29] is employed to develop a new partitioned interface-quasi-Newton technique. The particularity of the new technique is that an approximation for the *inverse* of the Jacobian that appears in the Newton linearization is obtained such that no linear system with the Jacobian has to be solved in every Newton–Raphson iteration and hence it is named IQN–ILS (interface-quasi-Newton with inverse Jacobian from a least-squares model). This new technique couples a black-box flow solver and black-box structural solver and can be implemented easily in an FSI framework that currently uses Gauss–Seidel iterations with or without relaxation.

New partitioned solution techniques are usually compared with other partitioned techniques at their introduction but the difference in terms of the duration of the simulation compared to monolithic techniques remains unclear. Therefore, the performance of the new partitioned IQN–ILS method is compared with a monolithic Newton method for several FSI problems with an incompressible fluid. To analyze the difference in performance between both solution techniques without other causes for differences, ADINA (Adina R&D Inc., Watertown, MA, USA) has been used as this program is capable of both monolithic Newton–Raphson iterations and partitioned iterations between the flow and structural solvers. Only a small modification of the partitioned technique in ADINA was necessary to enable the new IQN–ILS algorithm. Consequently, both the mathematical model and the solver for the resulting equations are identical and also the convergence of the FSI problem is controlled by ADINA such that both techniques solve the problem to the same accuracy.

Previous comparisons between the monolithic solution of an FSI problem and the partitioned solution with black-box solvers seem to have been limited to 1D problems [31]. In this paper, 2D and 3D cases with incompressible fluids from several authors [32–34] are simulated. The limitations of the partitioned technique are demonstrated and the influence of problem size, large displacements and fluid/solid density ratio on the difference in performance are analyzed.

The remainder of this paper is organized as follows: In Section 2, the equilibrium conditions and the governing equations of the FSI problem are introduced. The new partitioned algorithm and the monolithic algorithm of ADINA are explained in Section 3 and their performance is subsequently compared for five cases in Section 4. Finally, the conclusions are given in Section 5.

2. Governing equations

In this section, the equilibrium conditions on the interface and the governing equations for the fluid flow and the structure are outlined briefly. The fluid domain and structural domain are indicated as Ω_f and Ω_s, respectively, and their boundaries as Γ_f and Γ_s. The fluid–structure interface $\Gamma_i = \Gamma_f \cap \Gamma_s$ is the common boundary of these domains. Fluid and solid are indicated with subscripts f and s and values on the fluid–structure interface are underlined [35,36].

2.1. Equilibrium conditions

The equilibrium conditions on the fluid–structure interface are the kinematic condition

$$\underline{\boldsymbol{d}}_f = \underline{\boldsymbol{d}}_s \tag{1}$$

and the dynamic condition

$$\boldsymbol{n}_f \cdot \underline{\tau}_f = -\boldsymbol{n}_s \cdot \underline{\tau}_s \tag{2}$$

with $\underline{\boldsymbol{d}}$ the displacement, $\underline{\tau}$ the stress tensor and $\boldsymbol{n}$ the unit normal vector that points outwards from the domain Ω. The Dirichlet–Neumann formulation of the FSI problem is employed which means that the flow equations are solved for a given velocity of the fluid–structure interface whereas a stress is imposed on the fluid–structure boundary of the solid domain.

2.2. Flow equations

The unsteady flow of an incompressible, isothermal fluid is governed by the conservation of mass and the Navier–Stokes equations, given by

$$\nabla \cdot \boldsymbol{v}_f = 0 \tag{3a}$$

$$\rho_f \frac{\partial \boldsymbol{v}_f}{\partial t} + \rho_f \nabla \cdot (\boldsymbol{v}_f \boldsymbol{v}_f) - \nabla \cdot \tau_f = \boldsymbol{f}_f \tag{3b}$$

for $\boldsymbol{x} \in \Omega_f$. The flow velocity is denoted by $\boldsymbol{v}_f$, ρ_f is the fluid density, t the time and $\boldsymbol{f}_f$ the body forces per unit of volume on the fluid. For a Newtonian fluid, the stress tensor is defined as $\tau_f = -p\mathbf{I} + 2\mu\gamma$ with $\gamma = \frac{1}{2}(\nabla \boldsymbol{v}_f + (\nabla \boldsymbol{v}_f)^T)$ the rate of strain tensor and μ the fluid viscosity. In the Dirichlet–Neumann formulation, the kinematic condition (1) on a no-slip interface results in

$$\underline{\boldsymbol{v}}_f = \dot{\underline{\boldsymbol{d}}}_s \tag{4}$$

for $\boldsymbol{x} \in \Gamma_i$. Due to the time discretization, this velocity boundary condition becomes an imposed displacement on the fluid–structure interface. Appropriate conditions have to be imposed on the remainder of the boundary $\Gamma_f \setminus \Gamma_i$.

In FSI calculations, the flow equations have to be solved on a moving mesh due to the deformation of the structure and therefore they are discretized in the arbitrary Lagrangian–Eulerian formulation. For the discrete equations in a control volume to be conservative in time, the volume swept by the control volume's boundaries must be calculated in such a way that it is consistent with the time discretization of the change of its volume. Therefore, every discretization has its own requirement with respect to a consistent calculation of the time-dependent geometric quantities [14] which is also referred to as the geometric conservation law [37].

2.3. Structural equations

The deformation of the structure is determined by the conservation of momentum

$$\rho_s \frac{\partial^2 \boldsymbol{d}_s}{\partial t^2} - \nabla \cdot \tau_s = \boldsymbol{f}_s \tag{5}$$

for $\boldsymbol{x} \in \Omega_s$ with ρ_s the structural density and $\boldsymbol{f}_s$ the body force per unit volume on the structure. The relation between the stress tensor τ_s and the strains is given by the constitutive equation of the material. Again using the Dirichlet–Neumann formulation, the stress is imposed by Eq. (2) for $\boldsymbol{x} \in \Gamma_i$ and suitable boundary conditions have to be applied on $\Gamma_s \setminus \Gamma_i$.

2.4. Algebraic equations

The discretization of the mathematical model in space with finite elements and in time with the time integration algorithm of Ref. [38] for both the structural and the fluid flow response, which makes the integration 'consistent', results in a system of algebraic equations. The solution vectors corresponding to the nodes of the fluid and solid domain are defined as $\boldsymbol{X}_f$ and $\boldsymbol{X}_s$ and the interface values are consequently given by $\underline{\boldsymbol{d}}_s = \underline{\boldsymbol{d}}_s(\boldsymbol{X}_s)$ and $\underline{\tau}_f = \underline{\tau}_f(\boldsymbol{X}_f)$. The algebraic equations for the fluid–structure interaction problem are written concisely as

$$\mathbf{F}(\boldsymbol{X}) \equiv \begin{bmatrix} \mathbf{F}_f(\boldsymbol{X}_f, \underline{\boldsymbol{d}}_s(\boldsymbol{X}_s)) \\ \boldsymbol{F}_s(\boldsymbol{X}_s, \underline{\tau}_f(\boldsymbol{X}_f)) \end{bmatrix} = \mathbf{0} \tag{6a}$$

with

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_f \\ \boldsymbol{X}_s \end{bmatrix} \tag{6b}$$

the solution vector of the coupled problem and $\mathbf{F}_f$ and $\mathbf{F}_s$ the discrete equations in Ω_f and Ω_s, respectively. When the flow problem and the structural problem are not discretized in the same way on the fluid–structure interface, the 'consistent' transfer of variables between both discretizations is required [14,33].

3. Solution techniques

In this section, we describe the partitioned and monolithic solution algorithms.

3.1. Partitioned interface-quasi-Newton method

As stated in the introduction, interface-Newton techniques reformulate the FSI problem (6a) as an equation in the discrete position of the fluid–structure interface only

$$\mathbf{R}(\underline{\boldsymbol{d}}_s) = \mathbf{0} \tag{7}$$

with $\mathbf{R}$ the residual operator. The number of degrees-of-freedom in the position of the interface d $\underline{\boldsymbol{d}}_s$ is denoted as p. If solving the flow equations $\mathbf{F}_f(\boldsymbol{X}_f, \underline{\boldsymbol{d}}_s) = \mathbf{0}$ for $\boldsymbol{X}_f$ with given $\underline{\boldsymbol{d}}_s$ is represented by the function $\underline{\tau}_f = F_f(\underline{\boldsymbol{d}}_s)$ and similarly solving the structural equations $\mathbf{F}_s(\boldsymbol{X}_s, \underline{\tau}_f) = \mathbf{0}$ for $\boldsymbol{X}_s$ with given $\underline{\tau}_f$ is denoted by $\underline{\boldsymbol{d}}_s = F_s(\underline{\tau}_f)$, then the residual of the FSI problem is defined as

$$\mathbf{R}(\underline{\boldsymbol{d}}_s) = F_s \circ F_f(\underline{\boldsymbol{d}}_s) - \underline{\boldsymbol{d}}_s \tag{8}$$

To solve Eq. (7) using Newton–Raphson iterations

$$\frac{\partial \mathbf{R}^k}{\partial \underline{\boldsymbol{d}}_s} \Delta \underline{\boldsymbol{d}}_s^k = -\mathbf{R}^k \tag{9a}$$

$$\underline{\boldsymbol{d}}_s^{k+1} = \underline{\boldsymbol{d}}_s^k + \Delta \underline{\boldsymbol{d}}_s^k \tag{9b}$$

with $\mathbf{R}^k \equiv \mathbf{R}(\underline{\boldsymbol{d}}_s^k) \in \mathbb{R}^{p \times 1}$, a linear system (9a) has to be solved in every Newton–Raphson iteration. The Jacobian matrix $\partial \mathbf{R}^k / \partial \underline{\boldsymbol{d}}_s$ has to be known explicitly if a direct solver is used for this linear system or it has to be possible to calculate the product of the Jacobian matrix with a vector if this linear system is solved iteratively. However, explicit calculation of the Jacobian of the residual operator would require knowledge of the Jacobians of the solvers for the flow equations and structural equations while partitioned algorithms are most useful if they couple existing flow solvers and structural solvers without access to their Jacobian matrices. The matrix–vector product can be approximated using finite-differences but this requires an evaluation of the residual operator in every iteration of the iterative solver.

In Ref. [29], a technique to approximate the Jacobian of a function based on a set of inputs and outputs of that function has been introduced. In the present paper, it is shown that with a special choice of the inputs and outputs, an approximation for the inverse of the Jacobian $\partial \mathbf{R}^k / \partial \underline{\boldsymbol{d}}_s$ can be obtained. With this approximated inverse of the Jacobian, an interface-quasi-Newton technique is constructed which does not require the solution of the linear system (9a) in every Newton–Raphson iteration. The complete procedure is given in the Box and is explained in detail below.

First an extrapolation of the interface position based on the previous time steps is calculated on line 1 where a left superscript identifies the time level. If no left superscript is used, the current time level $n + 1$ is meant. The flow equations and structural equations are subsequently solved which results in a first residual vector $\mathbf{R}^0$. Because at least two residual vectors are required to construct an approximate Jacobian, one iteration with a fixed relaxation factor ω is performed on line 3. The relaxation has to prevent divergence such that $\mathbf{R}(\underline{\boldsymbol{d}}_s^1)$ can still be calculated.

Once at least two vectors $\mathbf{R}$ and $\tilde{\underline{\boldsymbol{d}}}_s$ are known, differences between the previous values (superscript i) and the last value (superscript k) can be calculated

$$\Delta \mathbf{R}^i = \mathbf{R}^i - \mathbf{R}^k \tag{10a}$$

$$\Delta \tilde{\underline{\boldsymbol{d}}}_s^i = \tilde{\underline{\boldsymbol{d}}}_s^i - \tilde{\underline{\boldsymbol{d}}}_s^k \tag{10b}$$

for $i = 0, \ldots, k-1$. Each $\Delta \mathbf{R}^i$ corresponds to a $\Delta \tilde{\underline{\boldsymbol{d}}}_s^i$ and these vectors are stored as the columns of the matrices

$$V^k = \begin{bmatrix} \Delta \mathbf{R}^{k-1} & \Delta \mathbf{R}^{k-2} & \cdots & \Delta \mathbf{R}^1 & \Delta \mathbf{R}^0 \end{bmatrix} \tag{11a}$$

$$W^k = \begin{bmatrix} \Delta \tilde{\underline{\boldsymbol{d}}}_s^{k-1} & \Delta \tilde{\underline{\boldsymbol{d}}}_s^{k-2} & \cdots & \Delta \tilde{\underline{\boldsymbol{d}}}_s^1 & \Delta \tilde{\underline{\boldsymbol{d}}}_s^0 \end{bmatrix} \tag{11b}$$

The number of columns in the matrices V^k and W^k is indicated by q. If q would exceed p after a very high number of iterations, the number of columns q in V^k and W^k is limited to p by discarding the rightmost columns. The desired value of the residual is of course a vector with zeros and the difference between this desired value and the current value $\Delta \mathbf{R} = \mathbf{0} - \mathbf{R}^k$ is subsequently approximated as a linear combination of the known $\Delta \mathbf{R}^i$

$$\Delta \mathbf{R} \approx \sum_{i=0}^{k-1} \alpha_i^k \Delta \mathbf{R}^i = V^k \boldsymbol{\alpha}^k \tag{12}$$

with α_i^k element i of $\boldsymbol{\alpha}^k \in \mathbb{R}^{q \times 1}$. Because $q \leqslant p$, this problem is overdetermined and it is solved in the least squares sense [39]. To solve the linear least squares problem, the so-called economy size QR-decomposition of V^k is calculated using Householder transformations [39]

$$V^k = Q^k R^k \tag{13}$$

with $Q^k \in \mathbb{R}^{p \times q}$ orthogonal and $R^k \in \mathbb{R}^{q \times q}$ upper triangular. The coefficient vector $\boldsymbol{\alpha}^k$ is obtained by solving the triangular system

$$R^k \boldsymbol{\alpha}^k = Q^{k^T} \Delta \mathbf{R} \tag{14}$$

for α^k using back substitution. The $\Delta \tilde{\underline{\boldsymbol{d}}}_s$ that corresponds to $\Delta \mathbf{R}$ is then calculated as a linear combination of the previous $\Delta \tilde{\underline{\boldsymbol{d}}}_s^i$, similar to Eq. (12), giving

$$\Delta \tilde{\underline{\boldsymbol{d}}}_s = \sum_{i=0}^{k-1} \alpha_i^k \Delta \tilde{\underline{\boldsymbol{d}}}_s^i = W^k \boldsymbol{\alpha}^k \tag{15}$$

Since $\mathbf{R}(\underline{\boldsymbol{d}}_s) = \tilde{\underline{\boldsymbol{d}}}_s(\underline{\boldsymbol{d}}_s) - \underline{\boldsymbol{d}}_s$ and since $\boldsymbol{\alpha}^k$ is a function of $\Delta \mathbf{R}$, this results in the relation

$$\Delta \underline{\boldsymbol{d}}_s = W^k \boldsymbol{\alpha}^k - \Delta \mathbf{R} \tag{16}$$

between $\Delta \mathbf{R}$ and $\Delta \underline{\boldsymbol{d}}_s$. The inverse approximate Jacobian is thus not calculated explicitly but Eq. (16) is a procedure to calculate the product of the approximation (indicated with a hat) for the inverse Jacobian and a vector $\Delta\mathbf{R} = -\mathbf{R}^k$

$$\Delta\underline{\boldsymbol{d}}_s = (\partial\hat{\mathbf{R}}^k/\partial\underline{\boldsymbol{d}}_s)^{-1}(-\mathbf{R}^k) = W^k\boldsymbol{\alpha}^k + \mathbf{R}^k \tag{17}$$

The position of the interface is finally updated on line 12.

If a residual vector is identical to another one or a linear combination of other residual vectors, one of the diagonal elements of R^k will be zero. Consequently, the equation corresponding with that row of R^k cannot be solved during the back substitution (14) and the corresponding element of $\boldsymbol{\alpha}^k$ is set to zero.

Algorithm. Algorithm of the interface-quasi-Newton method

```
 1: $\underline{\boldsymbol{d}}_s^0 = 5/2({}^n\underline{\boldsymbol{d}}_s) - 2({}^{n-1}\underline{\boldsymbol{d}}_s) + 1/2({}^{n-2}\underline{\boldsymbol{d}}_s)$
 2: $\tilde{\underline{\boldsymbol{d}}}_s^0 = F_s \circ F_f(\underline{\boldsymbol{d}}_s^0)$ and $\mathbf{R}^0 = \tilde{\underline{\boldsymbol{d}}}_s^0 - \underline{\boldsymbol{d}}_s^0$
 3: $\underline{\boldsymbol{d}}_s^1 = \underline{\boldsymbol{d}}_s^0 + \omega\mathbf{R}^0$
 4: $k = 1$
 5: **While** not converged **do**
 6:     $\tilde{\underline{\boldsymbol{d}}}_s^k = F_s \circ F_f(\underline{\boldsymbol{d}}_s^k)$ and $\mathbf{R}^k = \tilde{\underline{\boldsymbol{d}}}_s^k - \underline{\boldsymbol{d}}_s^k$
 7:     $V^k = [\Delta\mathbf{R}^{k-1} \ldots \Delta\mathbf{R}^0]$ with $\Delta\mathbf{R}^i = \mathbf{R}^i - \mathbf{R}^k$
 8:     $W^k = [\Delta\tilde{\underline{\boldsymbol{d}}}_s^{k-1} \ldots \Delta\tilde{\underline{\boldsymbol{d}}}_s^0]$ with $\Delta\tilde{\underline{\boldsymbol{d}}}_s^i = \tilde{\underline{\boldsymbol{d}}}_s^i - \tilde{\underline{\boldsymbol{d}}}_s^k$
 9:     $V^k = Q^k R^k$
10:     solve $R^k\alpha^k = Q^{k^T}(-\mathbf{R}^k)$ for $\alpha^k$ using back substitution
11:     $\Delta\underline{\boldsymbol{d}}_s = W^k\boldsymbol{\alpha}^k + \mathbf{R}^k$
12:     $\underline{\boldsymbol{d}}_s^{k+1} = \underline{\boldsymbol{d}}_s^k + \Delta\underline{\boldsymbol{d}}_s$
13:     $k = k + 1$
14: **Endwhile**
```

From Eq. (14), it can be seen that if part of $\Delta\mathbf{R}$ is orthogonal to V^k and thus to Q^k, the decomposition coefficients α^k will be zero for that part of $\Delta\mathbf{R}$. Eq. (16) 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. Only the components of $\Delta\mathbf{R}$ in the span of the columns of V^k are reduced with Newton–Raphson iterations. In Ref. [24], a stability analysis of Gauss–Seidel iterations between a flow solver and a structural solver for the partitioned solution of the flow in a straight flexible tube is presented and it is shown that particular components of the residual with a low wave number are unstable during Gauss–Seidel iterations. Hence, only these components have to be reduced with Newton–Raphson iterations while the remainder of the residual disappears during Gauss–Seidel iterations. This analysis forms the theoretical basis for the presented quasi-Newton method.

Generally, $q \ll p$ and thus the computational cost of this quasi-Newton method is limited because it mainly consists of the QR-decomposition of a $p \times q$ matrix and the solution of a $q \times q$ triangular system. To reduce the number of quasi-Newton iterations per time step, the matrices V^k and W^k as defined in Eq. (11b) can be combined with those from r previous time steps

$$V'^k = \begin{bmatrix} V^k & {}^nV^k & \ldots & {}^{n-r+2}V^k & {}^{n-r+1}V^k \end{bmatrix} \tag{18a}$$

$$W'^k = \begin{bmatrix} W^k & {}^nW^k & \ldots & {}^{n-r+2}W^k & {}^{n-r+1}W^k \end{bmatrix} \tag{18b}$$

The value of r that results in the fastest convergence is problem dependent but the optimum is flat such that the performance of the method is robust with respect to this parameter. The notation IQN–ILS (r) indicates that information from r time steps is reused. When information is reused, the relaxation on line 3 is not required because at least two residual vectors are already known at that point. The relaxation is then only performed in the first time step.

3.2. Monolithic Newton method

In the monolithic Newton (MN) method, the flow equations and structural Eq. (6a) are solved simultaneously with Newton–Raphson iterations

$$\begin{bmatrix} \frac{\partial\mathbf{F}_f^k}{\partial\boldsymbol{X}_f} & \frac{\partial\mathbf{F}_f^k}{\partial\boldsymbol{X}_s} \\ \frac{\partial\mathbf{F}_s^k}{\partial\boldsymbol{X}_f} & \frac{\partial\mathbf{F}_s^k}{\partial\boldsymbol{X}_s} \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{X}_f^k \\ \Delta\boldsymbol{X}_s^k \end{bmatrix} = -\begin{bmatrix} \mathbf{F}_f^k \\ \mathbf{F}_s^k \end{bmatrix} \tag{19a}$$

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \Delta\boldsymbol{X}^k \tag{19b}$$

with

$$\mathbf{F}_f^k \equiv \mathbf{F}_f(\boldsymbol{X}_f^k, \underline{\boldsymbol{d}}_s(\boldsymbol{X}_s^k)) \tag{20a}$$

$$\mathbf{F}_s^k \equiv \mathbf{F}_s(\boldsymbol{X}_s^k, \underline{\boldsymbol{\tau}}_f(\boldsymbol{X}_f^k)) \tag{20b}$$

This straightforward approach ensures that the interaction between the fluid and the structure is taken into account during the solution process due to the off-diagonal blocks of the Jacobian matrix in Eq. (19a).

3.3. Solvers

In this study, the solution of the flow equations and the structural equations inside the residual operator (7) of the partitioned approach was performed with a direct sparse solver and Newton–Raphson iterations but any other solver which is faster for a specific flow or structural problem could be used.

4. Numerical studies

In this section, the partitioned IQN–ILS method and the MN method are compared in five different FSI cases with incompressible fluids. The average number of Newton–Raphson iterations per time step (denoted as QN iterations) is only shown for the IQN–ILS method because the MN method has to solve the flow equations, the structural equations and their interaction with the same Newton–Raphson iterations while the quasi-Newton iterations in the IQN–ILS method only solve the fluid–structure interaction. The convergence criterion in the Euclidean norm for the solution of the FSI problem is 10^{-2} for both the interface's displacement and the force on the interface, unless indicated otherwise. All calculations have been performed on a dedicated machine with two Intel Xeon X5355 quad-core processors.

4.1. Wave propagation

The first case is the wave propagation in a straight, three-dimensional elastic tube [32]. The dimensions, boundary conditions and the coarsest mesh are shown in Fig. 1. The fluid has a density of 1000 kg/m^3 and a dynamic viscosity of 0.003 Pa · s and it is modeled by 8-node flow-condition-based interpolation (FCBI) elements [40]. The linear-elastic solid with density 1200 kg/m^3, Young's modulus 300, 000 N/m^2 and Poisson's ratio 0.3 is modeled using 9-node shell elements. Hundred time steps of 0.0001 s are

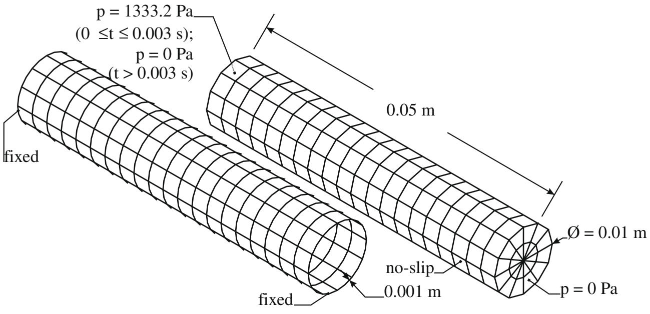
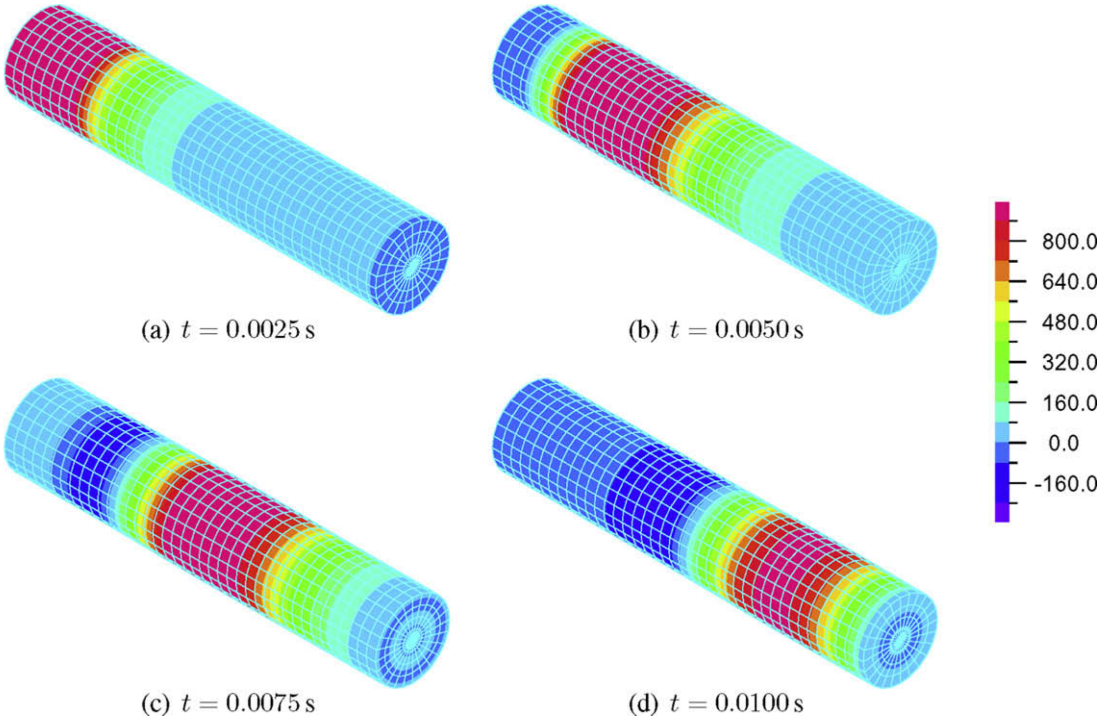


Fig. 1. Wave propagation in the straight elastic tube. Geometry, boundary conditions and coarsest mesh for the solid (left) and fluid (right) model.



(a) $t = 0.0025\,\mathrm{s}$ (b) $t = 0.0050\,\mathrm{s}$

(c) $t = 0.0075\,\mathrm{s}$ (d) $t = 0.0100\,\mathrm{s}$

Fig. 2. Wave propagation in the straight elastic tube. Pressure contours on the fluid–structure interface.

performed and the pressure pulse applied at the inlet propagates through the tube as shown in Fig. 2.

The goal of this problem is to show the influence of the problem size on the difference in performance between the IQN–ILS and MN method and the limitations of the partitioned technique. The calculation of the wave propagation has been performed on three meshes with respectively 480, 3840 and 12,960 fluid elements and 240, 960 and 2160 shell elements. Table 1 lists the average number of quasi-Newton iterations per time step, the time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12) in the entire simulation. The latter is negligible with respect to the time spent on the flow and structural equations. The table also indicates that the duration of the simulation with IQN–ILS is approximately twice the duration of the one with the MN method, regardless of the problem size. Without reuse of the previous time steps, more than 10 quasi-Newton iterations are required per time step of the partitioned simulation but this has been reduced to approximately 3 by the reuse of information from the 8 previous time steps.

Table 1
Wave propagation in the straight elastic tube. Average number of quasi-Newton iterations per time step, total time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12 in the Box on page 9) in the entire simulation and also the ratio of the time for the IQN–ILS simulation to the time for the MN simulation. The coarse, medium and fine meshes consist of respectively 480, 3840 and 12,960 fluid elements and 240, 960 and 2160 shell elements.

Mesh	Coarse	Medium	Fine
IQN–ILS (8)			
QN iterations [–]	3.23	3.28	3.35
Time for $\mathbf{F}_f$ [s]	2121	27399	189692
Time for $\mathbf{F}_s$ [s]	32	137	370
Time for IQN	[s] 1.63	6.44	15.29
MN			
Time for $\mathbf{F}_f$ [s]	1005	14527	93689
Time for $\mathbf{F}_s$ [s]	123	559	1258
Ratio time IQN–ILS to time MN [–]	1.91	1.83	2.00

Table 2 shows that a partitioned simulation becomes more difficult as the tube length increases which is explained by the stability analysis in Refs. [23,24]. The number of quasi-Newton iterations in the first time step ($n = 1$) increases significantly with increasing length of the tube. However, the number of quasi-Newton iterations averaged over all time steps and the ratio of the time for the IQN–ILS simulation to the time for the MN simulation increase more slowly until the tube length reaches 0.20 m due to the reuse from previous time steps. For a tube of 0.40 m, the partitioned simulation failed in the first time step due to too large displacements and forces during the quasi-Newton iterations. This indicates the limitations of a partitioned FSI simulation.

4.2. Mass conservation

Because the deformations in the previous numerical example are small, a similar experiment with large deformations is per-

Table 2
Wave propagation in the straight elastic tube. Average number of quasi-Newton iterations per time step, number of quasi-Newton iterations in the first time step ($n = 1$), total time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12) in the entire simulation and also the ratio of the time for the IQN–ILS simulation to the time for the MN simulation. The meshes consist of respectively 480, 960, 1920 and 3840 fluid elements and 240, 480, 960 and 1920 shell elements. For the longest tube, the partitioned simulation failed in the first time step.

Length [m]	0.05	0.10	0.20	0.40
IQN–ILS(8)				
QN iterations [–]	3.23	3.68	4.62	–
QN iterations ($n = 1$) [–]	11	16	23	–
Time for $\mathbf{F}_f$ [s]	2121	4113	10176	–
Time for $\mathbf{F}_s$ [s]	32	67	169	–
Time for IQN [s]	1.63	4.39	15.07	–
MN				
Time for $\mathbf{F}_f$ [s]	1005	1475	3611	6326
Time for $\mathbf{F}_s$ [s]	123	251	504	1008
Ratio time IQN–ILS to time MN [–]	1.91	2.42	2.52	–

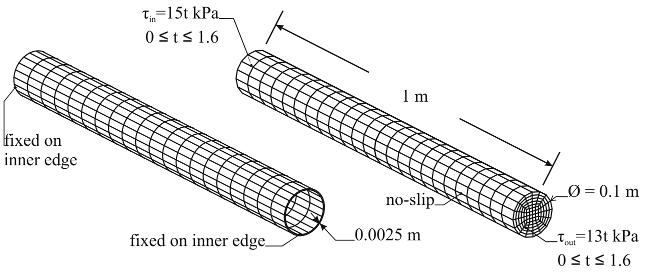


Fig. 3. Mass conservation test. Geometry, boundary conditions and mesh for the solid (left) and fluid (right) model.

formed. In the mass conservation test [33], the pressure at the inlet and outlet of the tube is increased in 16 equal steps and the steady solution is calculated in every step. The dimensions, boundary conditions and the mesh for this test are shown in Fig. 3. The mass flow rate at the inlet, middle and outlet of the tube in both the monolithic and partitioned simulation is equal to the benchmark results [33]. The fluid with density 1000 kg/m^3 and dynamic viscosity 1 Pa $\cdot$ s is also modeled by 8-node FCBI elements and the Mooney–Rivlin solid with parameters $\kappa_s = 10^9$ Pa, $C_1 = 2 \times 10^5$ Pa and $C_2 = 10^5$ Pa is modeled using 27/4 u/p solid elements [41,42]. In this case, the convergence criterion for the solution of the FSI problem is 10^{-3}.

Fig. 4 shows the large deformations at the end of the simulation. According to Table 3, the simulation with IQN–ILS takes 33% longer than the simulation with the MN method but both algorithms are capable of calculating the response. No information from the previous steps is reused by the IQN–ILS method because the boundary conditions of subsequent steps are significantly different such that information from previous steps is no longer relevant in the current step. Again, the time spent on lines 7–12 of the IQN–ILS algorithm is negligible in comparison with the time spent on the flow and structural equations.

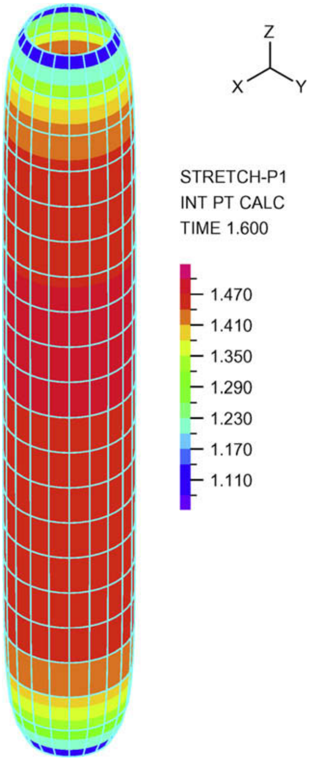


Fig. 4. Mass conservation test. Maximum principal stretch.

Table 3
Mass conservation test. Average number of quasi-Newton iterations per time step, total time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12) in the entire simulation and also the ratio of the time for the IQN–ILS simulation to the time for the MN simulation.

IQN–ILS	
QN iterations [–]	8.13
Time for $\mathbf{F}_f$ [s]	512
Time for $\mathbf{F}_s$ [s]	355
Time for IQN [s]	2.13
MN	
Time for $\mathbf{F}_f$ [s]	582
Time for $\mathbf{F}_s$ [s]	71
Ratio time IQN–ILS to time MN [–]	1.33

4.3. Strong coupling

The strong coupling test [33] is an unsteady test which is difficult due to the strong interaction between fluid and solid. The dimensions, boundary conditions and the mesh for this test are shown in Fig. 5. The fluid domain is discretized with 8-node FCBI elements with density 1 kg/m^3 and dynamic viscosity 1 Pa s. The Mooney–Rivlin solid with parameters $\kappa_s = 3.3333$ Pa, $C_1 = 1.6667$ Pa and $C_2 = 0$ Pa does not have inertia and consists of 8/1 u/p solid elements [41,42]. The simulation with time step 0.02 s continues until the fluid domain has almost zero thickness. The coarse mesh is constructed with 10 fluid elements and 1 solid element and the fine mesh with 100 fluid elements and 4 solid elements. This case has been analyzed with a convergence criterion of 10^{-4} for the solution of the FSI problem.

The IQN–ILS method passes the strong coupling test; the displacement and velocity of the fluid–structure interface are shown in Fig. 6. Table 4 indicates that the partitioned simulation is 1.76 times more expensive than the monolithic simulation on the coarse mesh and 3.16 times more expensive on the fine mesh although the number of quasi-Newton iterations is approximately 4.6 in both cases. As opposed to the wave propagation in Section 4.1, the problem size increases from the coarse to the fine mesh but the number of degrees of freedom on the fluid–structure interface remains constant and consequently the number of quasi-Newton iterations is similar on both meshes.

4.4. Shell in steady-state cross-flow

The shell in steady-state cross-flow is a benchmark with large displacements [33]. The dimensions and boundary conditions for this test are shown in Fig. 7 and the velocity at the inlet is increased in 10 equal steps. The fluid has a density of 1000 kg/m^3 and a dynamic viscosity of 0.1 Pa s and it is modeled by respectively $13 \times 18 \times 8$ and $24 \times 36 \times 15$ 8-node FCBI elements in the

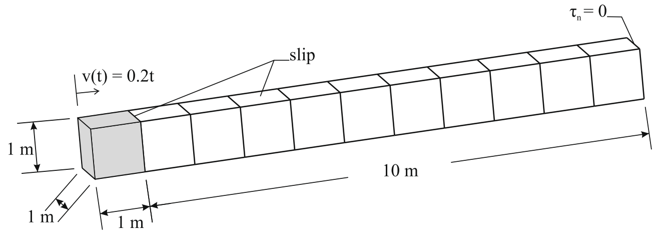


Fig. 5. Strong coupling test. Geometry, boundary conditions and coarsest mesh for the solid (grey) and fluid (white) model. The solid is confined to stay in the channel.

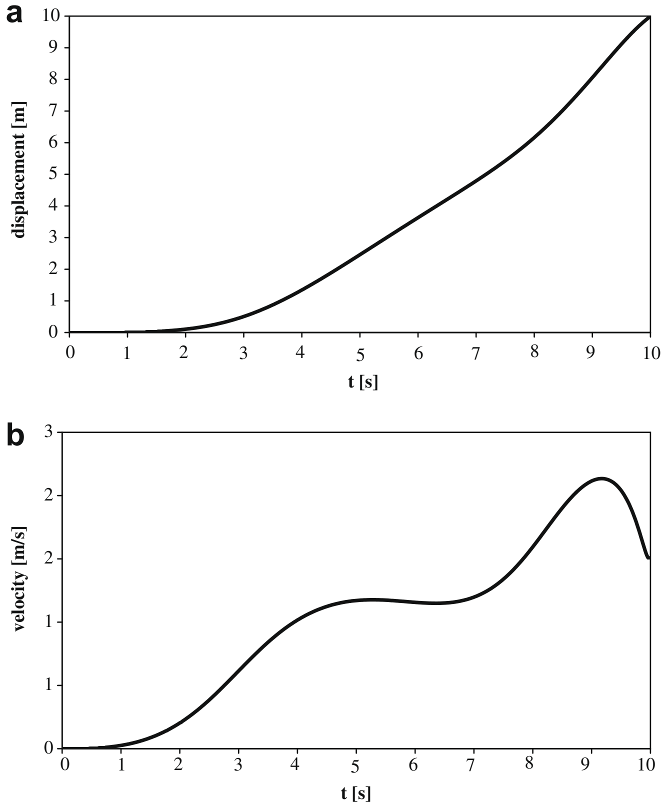


Fig. 6. Strong coupling test. Displacement (a) and velocity (b) of the fluid–structure interface.

Table 4
Strong coupling test. Average number of quasi-Newton iterations per time step, total time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12) in the entire simulation and also the ratio of the time for the IQN–ILS simulation to the time for the MN simulation. The coarse and fine meshes consist of respectively, 10 and 100 equal fluid elements and 1 and 4 equal solid elements.

Mesh	Coarse	Fine
IQN–ILS		
QN iterations [–]	4.58	4.65
Time for $\mathbf{F}_f$ [s]	112	691
Time for $\mathbf{F}_s$ [s]	1	2
Time for IQN [s]	0.24	0.31
MN		
Time for $\mathbf{F}_f$ [s]	62	210
Time for $\mathbf{F}_s$ [s]	2	9
Ratio time IQN–ILS to time MN [–]	1.76	3.16

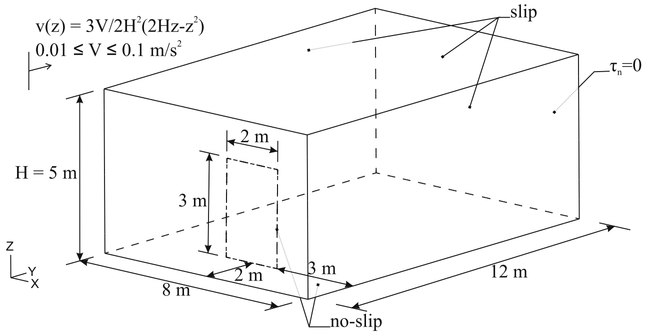


Fig. 7. Shell in steady-state cross-flow test. Geometry and boundary conditions for the solid and fluid model.

Table 5
Shell in steady-state cross-flow test. Average number of quasi-Newton iterations per time step, total time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12) in the entire simulation and also the ratio of the time for the IQN–ILS simulation to the time for the MN simulation. The coarse and fine meshes consist of respectively $13 \times 18 \times 8$ and $24 \times 36 \times 15$ fluid elements and 6×12 solid elements.

Mesh	Coarse	Fine
IQN–ILS(1)		
QN iterations [–]	3.22	3.22
Time for $\mathbf{F}_f$ [s]	119	2506
Time for $\mathbf{F}_s$ [s]	12	14
Time for IQN [s]	0.01	0.06
MN		
Time for $\mathbf{F}_f$ [s]	186	4510
Time for $\mathbf{F}_s$ [s]	3	3
Ratio time IQN–ILS to time MN [–]	0.69	0.56

coarse and fine meshes. The linear-elastic solid with Young's modulus 70×10^9 N/m^2 and Poisson's ratio 0.3 is modeled using 6×12 equal MITC4 shell elements with thickness 1.25×10^{-3} m.

The partitioned simulation yields the benchmark results obtained with the monolithic method. Table 5 indicates that the IQN–ILS method is faster for this simulation, on the coarse mesh and even more on the fine mesh. The IQN–ILS algorithm required on average 3.22 iterations per step and reused information from the previous step to accelerate the convergence.

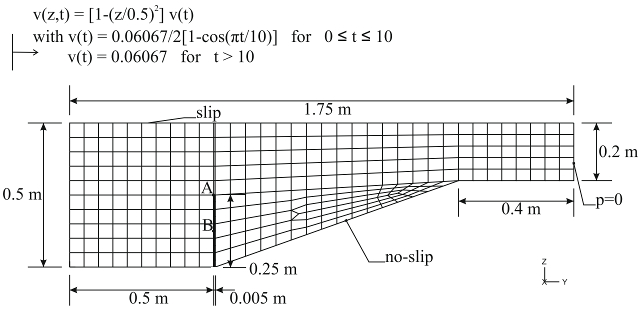
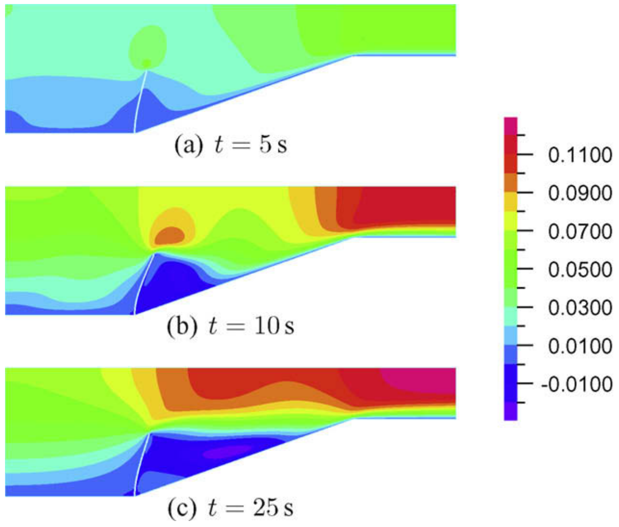

v(z,t) = [1-(z/0.5)²] v(t)
with v(t) = 0.06067/2[1-cos(πt/10)] for 0 ≤ t ≤ 10
v(t) = 0.06067 for t > 10
slip
1.75 m
0.2 m
0.5 m
A
B
0.4 m
p=0
no-slip
0.25 m
0.5 m
0.005 m
z
Y

Fig. 8. Flexible restrictor in the converging channel. Geometry, boundary conditions and mesh for the solid (black) and fluid (white) model.


(a) $t = 5$ s

(b) $t = 10$ s

(c) $t = 25$ s

Fig. 9. Flexible restrictor in the converging channel. Horizontal velocity.

4.5. Restrictor in converging channel

The last example is a flexible restrictor flap in a converging channel [34] which is used to demonstrate the effect of the fluid/solid density ratio on the performance difference. The dimensions, boundary conditions and mesh for this simulation are shown in Fig. 8, as well as the points A and B which are located on the left hand side of the restrictor. Fig. 8 represents half of the channel and a symmetry boundary condition is applied on the top edge. The fluid with dynamic viscosity 0.1 Pa s is modeled with 4-node FCBI elements. The linear-elastic solid with density 1500 kg/m^3, Young's modulus 2.3×10^6 N/m^2 and Poisson's ratio 0.45 is modeled with 9-node solid elements. A total of 250 time steps of 0.1 s are calculated. For the reference fluid density of 956 kg/m^3, the horizontal velocity in the entire field is depicted in Fig. 9 and the pressure and horizontal displacement of the points A and B are shown in Fig. 10.

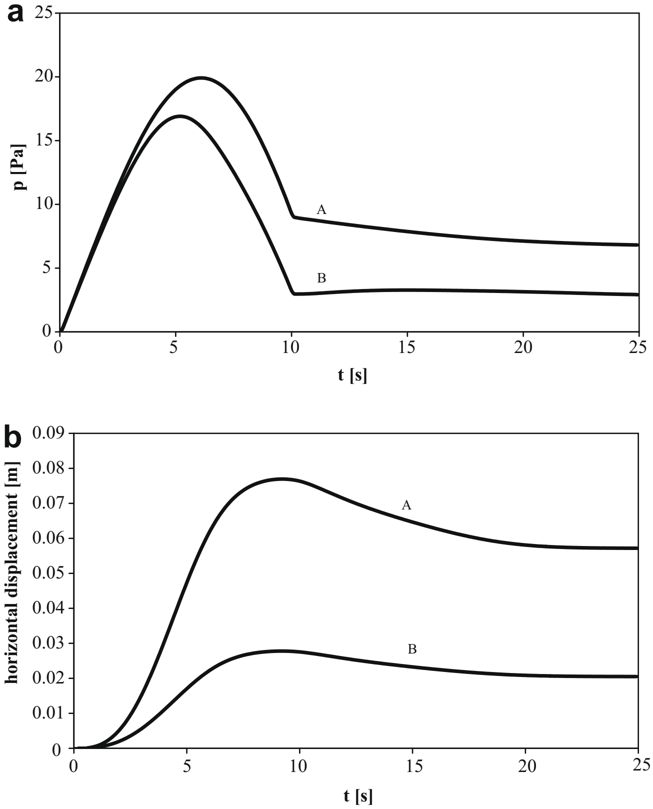


Fig. 10. Flexible restrictor in the converging channel. Pressure (a) and horizontal displacement (b) in point A and B.

Table 6
Flexible restrictor flap in a converging channel. Average number of quasi-Newton iterations per time step, total time spent on the flow equations $\mathbf{F}_f$, the structural equations $\mathbf{F}_s$ and on the IQN algorithm (lines 7–12) in the entire simulation and also the ratio of the time for the IQN–ILS simulation to the time for the MN simulation. The density of the fluid is varied from 250 kg/m^3 to 1750 kg/m^3, the density of the solid is 1500 kg/m^3.

ρ_f [kg/m^3]	250	500	750	956	1300	1500	1750
IQN–ILS (2)							
QN iterations [–]	1.51	1.97	2.30	2.48	2.46	2.53	2.80
Time for $\mathbf{F}_f$ [s]	174	273	355	404	433	460	516
Time for $\mathbf{F}_s$ [s]	1	4	4	5	6	5	9
Time for IQN [s]	0.16	0.16	0.33	0.39	0.48	0.40	0.71
MN							
Time for $\mathbf{F}_f$ [s]	125	155	177	186	200	207	213
Time for $\mathbf{F}_s$ [s]	8	10	11	12	13	12	13
Ratio time IQN–ILS to time MN [–]	1.32	1.67	1.91	2.07	2.07	2.13	2.33

The fluid density is subsequently varied from 250 kg/m^3 to 1750 kg/m^3 and the comparison between the IQN–ILS and MN method is given in Table 6. As the fluid density increases, the ratio of the time for the IQN–ILS simulation to the time for the MN simulation increases from 1.32 to 2.33 and the average number of quasi-Newton iterations per time step increases from 1.51 to 2.80. This effect of the fluid/solid density ratio is expected and explained by the stability analysis of partitioned FSI algorithms in Refs. [23,24]. Information from the two previous time steps is reused by the IQN–ILS method.

5. Conclusions

A new interface-quasi-Newton (IQN) algorithm for partitioned FSI simulations has been presented. The algorithm can easily be implemented as a replacement for existing algorithms because as in other schemes the flow solver and structural solver are treated as "black-boxes". The performance has been compared with monolithic Newton (MN) simulations 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 simulation to the time for the MN simulation was between 1/2 and 4, but of course there are cases for which the partitioned simulations do not converge.

The conclusions of a comparison as given here are difficult to generalize. First, while problems of various characteristics have been solved, still, only specific problems have been considered and in general rather small problems in number of equations. Second, the solutions of the structural equations and the flow equations have been obtained using a direct sparse solver with full Newton–Raphson iterations. However, different solver schemes, in particular much more efficient for the fluid equations when the number of elements becomes very large, are frequently used, in ADINA and otherwise, see for example, [43]. The performance comparisons may look different when different problems are solved and other solver schemes are used.

We should also mention that the data transfer to a separate program, in which the IQN algorithm has been implemented, has been neglected. This data transfer would not exist if the algorithm were implemented together with the solver programs, and needs to be optimized otherwise.

However, while there are these shortcomings of the study, the new algorithm is valuable, and the general observations given in the paper can be used to assess whether a monolithic or partitioned solution of a fluid–structure interaction problem might be more effective.

Acknowledgements

J. Degroote gratefully acknowledges a Ph.D. fellowship and a grant for a long stay abroad at the Massachusetts Institute of Technology of the Research Foundation – Flanders (FWO). Special thanks are also given to Dr. Nagi Elabbasi from ADINA R&D Inc. for the ADINA program support.

References

[1] Farhat C, van der Zee K, Geuzaine P. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. Comput Methods Appl Mech Eng 2006;195:1973–2001.

[2] Willcox K, Paduano J, Peraire J. Low order aerodynamic models for aeroelastic control of turbomachines. In: Proceedings of 40th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials (SDM) Conference. St Louis, MO, USA; 1999. p. 1–11.

[3] Agamloh E, Wallace A, von Jouanne A. Application of fluid–structure interaction simulation of an ocean wave energy extraction device. In: Proceedings of 44th AIAA aerospace sciences meeting and exhibit, vol. AIAA 2006-1001. Reno, NV, USA; 2006. p. 1–14.

[4] Stein K, Tezduyar T, Kumar V, Sathe S, Benney R, Charles R. Numerical simulation of soft landing for clusters of cargo parachutes. In: Niettaanmäki P, Rossi T, Majava K, Pieronneau O, editors. European congress on computational methods in applied sciences and engineering ECCOMAS 2004. Finland: Jyväskylä; 2004. p. 1–4.

[5] dos Santos N, Gerbeau JF, Bourgat JF. Partitioned FSI strategy for simulations of a thin elastic valve. In: Wesseling P, Oñate E, Périaux J, editors. European conference on computational fluid dynamics ECCOMAS CFD 2006. Delft, The Netherlands: ECCOMAS; 2006. p. 1–10.

[6] Dumont K, Vierendeels J, Kaminsky R, Van Nooten G, Verdonck P, Bluestein D. Comparison of the hemodynamic and thrombogenic performance of two bileaflet mechanical heart valves using a CFD/FSI model. J Biomech Eng – Trans ASME 2007;129(4):558–65.

[7] Taylor C, Draney M, Ku J, Parker D, Steele B, Wang K, et al. Predictive medicine: computational techniques in therapeutic decision-making. Comput Aid Surg 1999;4:231–47.

[8] Bathe M, Kamm R. A fluid–structure interaction finite element analysis of pulsatile blood flow through a compliant stenotic artery. J Biomech Eng 1999;121(4):361–9.

[9] Kaazempur-Mofrad M, Bathe M, Karcher H, Younis H, Seong H, Shim E, et al. The role of simulation in understanding biological systems. Comput Struct 2003;81(8–11):715–26.

[10] Wilson N, Arko F, Taylor C. Predicting changes in blood flow in patient-specific operative plans for treating aortoiliac occlusive disease. Comput Aid Surg 2005;10(4):257–77.

[11] Bathe KJ, Zhang H, Wang M. Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions. Comput Struct 1995;56(2–3):193–213.

[12] Rugonyi S, Bathe KJ. On the finite element analysis of fluid flows fully coupled with structural interactions. Comput Model Eng Sci 2001;2:195–212.

[13] Heil M. An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. Comput Methods Appl Mech Eng 2004;193:1–23.

[14] Bathe KJ, Zhang H. Finite element developments for general fluid flows with structural interactions. Int J Numer Methods Eng 2004;60:213–32.

[15] Hron J, Turek S. A monolithic FEM/multigrid solver for ALE formulation of fluid structure interaction with application in biomechanics. In: Fluid–structure interaction – modelling, simulation, optimisation. In: Bungartz HJ, Schäfer M, editors. Lecture notes in computational science and engineering, vol. 53. Berlin: Springer; 2006, ISBN 3-540-34595-7. p. 146–70.

[16] Matthies H, Steindorf J. Partitioned strong coupling algorithms for fluid–structure interaction. Comput Struct 2003;81:805–12.

[17] Matthies H, Niekamp R, Steindorf J. Algorithms for strong coupling procedures. Comput Methods Appl Mech Eng 2006;195:2028–49.

[18] Gerbeau JF, Vidrascu M. A quasi-Newton algorithm based on a reduced model for fluid–structure interaction problems in blood flows. ESAIM: Math Model Numer Anal 2003;37(4):631–48.

[19] Gerbeau JF, Vidrascu M, Frey P. Fluid–structure interaction in blood flows on geometries based on medical imaging. Comput Struct 2005;83(2-3):155–65.

[20] van Brummelen E, Michler C, de Borst R. Interface-GMRES(R) acceleration of subiteration for fluid–structure-interaction problems. Report DACS-05-001 2005. Available from: http://www.em.lr.tudelft.nl/downloads/DACS-05-001.pdf.

[21] Michler C, van Brummelen E, de Borst R. Error-amplification analysis of subiteration-preconditioned GMRES for fluid–structure interaction. Comput Methods Appl Mech Eng 2006;195:2124–48.

[22] Michler C, van Brummelen E, de Borst R. An interface Newton–Krylov solver for fluid–structure interaction. Int J Numer Methods Fluids 2005;47(10–11):1189–95.

[23] Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. Comput Methods Appl Mech Eng 2005;194(42-44):4506–27.

[24] Degroote J, Bruggeman P, Haelterman R, Vierendeels J. Stability of a coupling technique for partitioned solvers in FSI applications. Comput Struct 2008;86(23–24):2224–34.

[25] Küttler U, Wall W. Fixed-point fluid–structure interaction solvers with dynamic relaxation. Comput Mech 2008.

[26] Lesoinne M, Farhat C. A higher-order subiteration free staggered algorithm for non-linear transient aeroelastic problems. AIAA J 1998;36(9):1754–6.

[27] Piperno S, Farhat C, Larrouturou B. Partitioned procedures for the transient solution of coupled aeroelastic problems – Part I: model problem, theory and two-dimensional application. Comput Methods Appl Mech Eng 1995;124:79–112.

[28] van Zuijlen A, de Boer A, Bijl H. Higher-order time integration through smooth mesh deformation for 3D fluid structure interaction simulations. J Comput Phys 2007;224:414–30.

[29] Vierendeels J, Lanoye L, Degroote J, Verdonck P. Implicit coupling of partitioned fluid–structure interaction problems with reduced order models. Comput Struct 2007;85(11–14):970–6.

[30] Vierendeels J. Implicit coupling of partitioned fluid–structure interaction solvers using reduced-order models. In: Bungartz HJ, Schäfer M, editors. Fluid–structure interaction – modelling simulation, optimisation. Lecture notes in computational science and engineering, vol. 53. Berlin: Springer; 2006, ISBN 3-540-34595-7. p. 1–18.

[31] Michler C, van Brummelen E, Hulshoff S, de Borst R. The relevance of conservation for stability and accuracy of numerical methods for fluid structure interaction. Comput Methods Appl Mech Eng 2003;192:4195–215.

[32] Fernandez M, Moubachir M. A Newton method using exact Jacobians for solving fluid–structure coupling. Comput Struct 2005;83:127–42.

[33] Bathe KJ, Ledezma G. Benchmark problems for incompressible fluid flows with structural interactions. Comput Struct 2007;85:628–44.

[34] Mok D, Wall W, Ramm E. Accelerated iterative substructuring schemes for instationary fluid–structure interaction. In: Bathe KJ, editor. Computational fluid and solid mechanics. Elsevier; 2001. p. 1325–8.

[35] Zhang H, Bathe KJ. Direct and iterative computing of fluid flows fully coupled with structures. In: Proceedings of the first MIT conference on computational fluid and solid mechanics. In: Bathe KJ, editor. Computational fluid and solid mechanics, vol. 2. Cambridge, MA, USA: Elsevier; 2001.

[36] ADINA R&D Inc., Watertown, MA, USA. Theory and modeling guide volume III: ADINA CFD & FSI; 2008. Report ARD 08-9.

[37] Lesoinne M, Farhat C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes and their impact on aeroelastic computations. Comput Methods Appl Mech Eng 1996;134:71–90.

[38] Bathe KJ. Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. Comput Struct 2007;85:437–45.

[39] Golub GH, Loan CFV. Matrix computations. 3rd ed. Baltimore, MD, USA: Johns Hopkins University Press; 1996.

[40] Bathe KJ, Zhang H. A flow-condition-based interpolation finite element procedure for incompressible fluid flows. Comput Struct 2002;80:1267–77.

[41] Sussman T, Bathe KJ. A finite element formulation for nonlinear incompressible elastic and inelastic analysis. Comput Struct 1987;26:357–409.

[42] Bathe KJ. Finite element procedures. Prentice Hall; 1996.

[43] Bathe KJ. Encyclopedia of computer science and engineering. In: Wah B, editor. Chapter the finite element method 1. John Wiley & Sons; 2008. doi:10.1002/9780470050118.ecse159.