Quasi-Newton Methods for Implicit Black-Box FSI Coupling

A.E.J. Bogaers[a,b], S. Kok[c], B.D. Reddy[d,e], T. Franz[d,f,g]

[a] *Advanced Mathematical Modelling, Modelling and Digital Sciences, CSIR, South Africa*
[b] *Department of Mechanical Engineering, University of Cape Town,South Africa*
[c] *Department of Mechanical and Aeronautical Engineering, University of Pretoria, South Africa*
[d] *Centre for Research in Computational and Applied Mechanics,University of Cape Town, South Africa*
[e] *Department of Mathematics and Applied Mathematics, Universityof Cape Town, South Africa*
[f] *Cardiovascular Research Unit, Chris Barnard Division of Cardiothoracic Surgery, University of Cape Town, South Africa*
[g] *Research Office, University of Cape Town, South Africa*

Abstract

In this paper we introduce a new multi-vector update quasi-Newton (MVQN) method for implicit coupling of partitioned, transient FSI solvers. The new quasi-Newton method facilitates the use of 'black-box' field solvers and under certain circumstances can be demonstrated to provide Newton-like convergence behaviour for strongly coupled FSI benchmark problems. We demonstrate the superior convergence behaviour and robust nature of the MVQN method compared to other well known quasi-Newton coupling schemes, including the least squares reduced order modelling (IBQN-LS) scheme, the classical rank-1 update Broyden's method and fixed point iterations with dynamic relaxation. The quasi-Newton methods are analysed on a suite of strongly coupled FSI problems, including but not limited to, internal, incompressible flow through a flexible tube where the solid density is an order of magnitude lower than the fluid density.

Keywords: Fluid-structure interactions, partitioned solver, black-box solver, quasi-Newton methods, implicit coupling

1. Introduction

With an ever increasing availability and size of computational resources at the disposal of researchers, multiphysics simulations are becoming an ever present reality. Fluid-structure interactions (FSI), a subclass of multiphysics problems, have in particular received much attention over the past several years. The main reason for this is the wide range of problems within the fields of engineering and life sciences that involve FSI. These problems, to mention but a few, range from flutter prediction in aeroelasticity [15, 25], parachute dynamics [26, 28] and blood flow through the vascular system [29, 32, 33].

There are two main approaches to solving the FSI problem, namely the monolithic approach or the partitioned approach. The monolithic approach is to solve all equations, including those related to the interface and computational domains in a single unified solver, typically involving some variant of Newton's method [4, 21]. Alternatively, a partitioned solver utilises separate field solvers for the fluid and solid domains, with these separate domains then coupled along the interface. Partitioned schemes can then be solved explicitly or allow for sub-iterations or implicit coupling in order to guarantee satisfaction of the governing equations along the interface.

One particular subclass of FSI problems which have been the focus of large bodies of research is the simulation of incompressible flows. Incompressible FSI present a number of numerical challenges, particularly when the solid and fluid densities are of equal orders of magnitude (for example the simulation of blood flow through arteries). In these situations, the numerical coupling between the solid and fluid domains are non-trivial, leading to what has been coined as the "added-mass" effect [7, 17]. The general trend in literature, in order to obtain a stable and robust solution procedure for these class of problems, has been that monolithic solvers are required [9, 20, 23]. While there are merits to this argument, it does however remain difficult to justify the enormous initial investment required in developing a monolithic solver. This is especially relevant considering the large availability of optimised fluid and solid solvers, which the use of monolithic solution procedures completely preclude. Equally, segregated solvers allow for the solution of smaller systems as opposed to one large unified set of equations, which unless properly preconditioned can be very expensive [20]. Partitioned solvers further allow for each of the sub-domain fields to be solved using field specific discretisation and solution schemes.

Several useful contributions have been made with regards to partitioned incompressible FSI technology, ranging from artificial compressibility [22, 11], robin boundary conditions [2], and computing the exact Jacobians or sensitivities [13, 16] of the cross-coupling terms between the

Email addresses: abogaers@csir.co.za (A.E.J. Bogaers), schalk.kok@up.ac.za (S. Kok), daya.reddy@uct.ac.za (B.D. Reddy), thomas.franz@uct.ac.za (T. Franz)

fluid and solid domains. While these contributions represent distinct steps in allowing for the re-use of existing solvers, they are limited in the sense that they require access to the source of one or both of the domain solvers.

In this paper we are interested in the application of FSI using 'black-box' solvers, where no access to source code is available. FSI solvers based on black-box solvers mean that most fluid or solid solvers, including commercially available solvers can be used. The general understanding however is that black-box coupling algorithms have only limited applicability to the range of complexity of problems to which they can be applied, and are often insufficient when dealing with FSI problems where the solid to fluid mass ratios are close to unity.

Several coupling algorithms have been proposed which facilitate the use of black-box field solvers. They typically involve fixed point iterations, with augmentations including dynamic relaxation or other similar acceleration techniques [23]. For strongly coupled problems the convergence behaviour is often poor, if convergence is obtained at all. Quasi-Newton methods, whereby the coupling Jacobians are approximated rather than explicitly computed have shown promising results. Most notably of these is the use of finite-differencing (FD) to approximate the matrix-vector product for Newton-Krylov solution methods [24]. While Newton-like convergence behaviour has been observed, the FD methods have been demonstrated to be sensitive to the choice of step size. They also require several solid and fluid solver calls within a given sub-iteration, often making them too expensive, especially when compared to monolithic solution schemes [23].

In this paper we will introduce a new quasi-Newton method designed for the coupling of transient, black-box FSI solvers. The coupling method is based on a multi-vector iterative updating scheme requiring only observations of the interface tractions and displacements. It allows for the efficient and robust solution of strongly coupled transient FSI problems. The new proposed coupling scheme is heuristic independent and can be applied to a wide range of complex FSI problems, which can be shown under certain circumstances to provide Newton-like convergence behaviour.

We will compare the new proposed scheme to the reduced order modelling (ROM) scheme introduced by Vierendeels *et al.* [30], one of the more promising black-box coupling schemes. We will illustrate how the ROM coupling scheme can be viewed as a minimum norm solution scheme. This provides insights into why the method works comparatively well and provides some explanations as to its behaviour with regards to the number of retained observations in time. We will further compare these methods to the classical Broyden's quasi-Newton method and the popular (though somewhat limited) Aitken's dynamic relaxation.

The outline of the remainder of the paper is as follows. In Section 2 we describe the general black-box FSI problem within a block-Newton framework. Section 4 introduces the multi-vector Jacobian update scheme and briefly outlines the other quasi-Newton methods investigated in this paper. Lastly, the performance of the quasi-Newton methods is analysed in Section 5 using four popular benchmark problems covering a wide range of complexity, including incompressible flow through a flexible tube where the solid density is an order of magnitude lower than the fluid density.

2. Coupled FSI Problem

In this paper we focus only on FSI coupling using black-box field solvers. We therefore refrain from limiting the discussion to a given choice of discretised equations, and rather denote each of the respective field solvers as interface operators that map interface displacements and forces. The field solvers operate independently on non-overlapping fluid and structural domains Ω^F and Ω^S which share a common interface Γ. The solid solver is therefore an interface operator $\boldsymbol{S}$ mapping a given interface force vector $\boldsymbol{f}_\Gamma^{n+1}$ to interface displacements

$$\boldsymbol{d}_\Gamma^{S,n+1} = \boldsymbol{S}\left(\boldsymbol{f}_\Gamma^{F,n+1}\right) \tag{1}$$

where $\boldsymbol{d}_\Gamma^{n+1}$is the interface displacement vector at time step $n+1$. Similarly, the fluid field solver is represented by a mapping operator $\boldsymbol{F}$ such that

$$\boldsymbol{f}_\Gamma^{F,n+1} = \boldsymbol{F}\left(\boldsymbol{d}_\Gamma^{S,n+1}\right). \tag{2}$$

The fluid field operator $\boldsymbol{F}$ denotes both the solution step of the fluid field variables as well as the mesh movement of the fluid domain nodal coordinates.

It is important to realise here, while we operate under the assumption of 'black-box' field solvers, that there are some important restrictions on the choice of potential solvers. Both solvers need to allow for full access to interface information as well as the ability to prescribe the relevant interface boundary conditions. For strong coupling algorithms, where several coupling iterations are required, it is further necessary that each of the field solvers allow for the primary variables to be reset to the converged solution from the previous time step (while still allowing the user to prescribe the new boundary conditions). Furthermore, since the focus of this paper is on interface coupling algorithms only, we assume that each of the field solve mapping operators include the other necessary building blocks required for successful FSI simulations. In this regard, we refer to interface information transfer schemes (especially if non-matching meshes along the interface are employed), mesh deformation schemes and if necessary, depending on the solver choice, that the fluid field solver incorporates an arbitrary Lagrangian-Eulerian (ALE) formulation accounting for convective velocity relative to the domain velocity.

For the FSI problem, it is essential that both the kinematic and dynamic continuity be satisfied at all times along the interface. In the case of no-slip boundary conditions on the moving interface the kinematic continuity states that the fluid flow velocity at the interface equals the boundary displacement

$$\boldsymbol{u}_\Gamma = \frac{\partial \boldsymbol{d}_\Gamma}{\partial t} \tag{3}$$

and dynamic continuity states that the interface stress states are equal at the interface,

$$\boldsymbol{\sigma}_\Gamma^S \cdot \boldsymbol{n} = \boldsymbol{\sigma}_\Gamma^F \cdot \boldsymbol{n} \tag{4}$$

where $\boldsymbol{n}$ is the respective interface normals.

3. Block-Newton Equations

In order to solve the coupled system defined by (1) and (2), we re-write the equations as a root finding problem, such that

$$\boldsymbol{r}_F = \boldsymbol{F}(\boldsymbol{d}(\boldsymbol{f})) - \boldsymbol{f} = \boldsymbol{0} \tag{5}$$
$$\boldsymbol{r}_S = \boldsymbol{S}(\boldsymbol{f}(\boldsymbol{d})) - \boldsymbol{d} = \boldsymbol{0} \tag{6}$$

where the functional dependency of forces on displacements and vice versa is indicated by $\boldsymbol{d}(\boldsymbol{f})$ and $\boldsymbol{f}(\boldsymbol{d})$.

The coupled system defined by equations (5) and (6) can then be solved by computing the system Jacobian and solving for an update in the Newton direction:

$$\begin{bmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I} & \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} \\ \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} & \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} - \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{f} \\ \Delta \boldsymbol{d} \end{Bmatrix} = \begin{bmatrix} -\boldsymbol{F}(\boldsymbol{d}(\boldsymbol{f})) + \boldsymbol{f}(\boldsymbol{d}) \\ -\boldsymbol{S}(\boldsymbol{f}(\boldsymbol{d})) + \boldsymbol{d}(\boldsymbol{f}) \end{bmatrix} \tag{7}$$

where the Newton update for coupling iteration $k+1$ is then computed by

$$\boldsymbol{f}_{k+1}^{n+1} = \boldsymbol{f}_k^{n+1} + \Delta \boldsymbol{f} \tag{8}$$
$$\boldsymbol{d}_{k+1}^{n+1} = \boldsymbol{d}_k^{n+1} + \Delta \boldsymbol{d}. \tag{9}$$

However, since we are using partitioned solvers, executed in a staggered fashion, we will solve the system of linear equations in a block-Newton fashion. Following a call to the fluid solver (that returns an interface traction f), we update the interface traction iterate

$$\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I}\right)\Delta \boldsymbol{f} = -\left(\boldsymbol{F}(\boldsymbol{d}(\boldsymbol{f})) - \boldsymbol{f}(\boldsymbol{d})\right) - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\Delta \boldsymbol{d}. \tag{10}$$

Similarly, following a call to the solid solver (that returns a displacement d), we update this displacement iterate

$$\left(\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} - \boldsymbol{I}\right)\Delta \boldsymbol{d} = -\left(\boldsymbol{S}(\boldsymbol{f}(\boldsymbol{d})) - \boldsymbol{d}(\boldsymbol{f})\right) - \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\Delta \boldsymbol{f}. \tag{11}$$

It is important to note here, by recalling equations (5) and (6), that $\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}$.The main purpose of the quasi-Newton methods discussed in the latter part of this paper is thus to obtain approximations to the system Jacobians $\boldsymbol{J_s} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}}$ and $\boldsymbol{J}_F = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}}$.

One can alternately set-up a Newton linearised system for the residual equation defined by the difference in displacement between the current and previous iterate:

$$\boldsymbol{r} = \boldsymbol{d}_{k+1} - \boldsymbol{d}_k. \tag{12}$$

A Newton system based on the residual equation then becomes

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\Delta \boldsymbol{d} = -\boldsymbol{r}. \tag{13}$$

The benefit of the linearised system (13) is that the inverse approximation of $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}$ can be constructed directly, thereby negating the necessity for a linear system solve step. The quasi-Newton methods discussed in this paper are equally applicable to the residual system, and in general the performance in terms of number of coupling iterations are comparable, while being somewhat less robust.

4. Quasi-Newton Coupling Algorithms for FSI

Quasi-Newton methods is a class of root finding algorithms based on Newton's method to solve for a set of non-linear equations where the system Jacobian is approximated rather than fully computed.

4.1. Multi-vector update method

The requirements for a partitioned coupling algorithm is that it should be both efficient and robust across a wide range of FSI problems. In the context of black-box partitioning, we assume that the largest cost with respect to each of the coupling iterations in a given time step is the time required by each of the respective field solvers. In order to guarantee an efficient solution procedure it becomes important to minimise the number of solver calls and hence the number of coupling iterations. It would further be ideal if the chosen coupling algorithm can be applied to a wide range of problems without the need for tuning a set of problem specific heuristics. In this section, we propose a new, multi-vector update quasi-Newton method (MVQN) to approximate the system Jacobians, which we believe satisfies all the expected requirements.

In order to facilitate the discussion, let us assume that we have completed k FSI coupling iterations within time step $n+1$. We therefore have k interface displacements, which transferred to the fluid solver, subsequently provided k corresponding interface tractions. Using the displacement and traction observations, we can now construct two differencing observations matrices

$$\Delta \boldsymbol{D}_F^{n+1} = [\boldsymbol{d}_k - \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k-1} - \boldsymbol{d}_{k-2}, ..., \boldsymbol{d}_2 - \boldsymbol{d}_1] \quad (14)$$

$$\Delta \boldsymbol{F}_F^{n+1} = \left[\boldsymbol{f}_k - \boldsymbol{f}_{k-1}, \boldsymbol{f}_{k-1} - \boldsymbol{f}_{k-2}, ..., \boldsymbol{f}_2 - \boldsymbol{f}_1\right], \quad (15)$$

where the subscript $_F$ denotes information with regards to the fluid field solver. To simplify notation, the superscript $n+1$ was dropped from each of the respective observation vectors. It should be noted here that the order and form of the differencing in (14) and (15) is not important, and one can just as easily subtract the last observation k from all previous observations $1, 2, ..., k-1$.

Given the two differencing matrices, it is then possible to construct a generalised secant equation for the fluid solver

$$\boldsymbol{J}_F^{n+1} \Delta \boldsymbol{D}_F^{n+1} = \Delta \boldsymbol{F_F}^{n+1}, \quad (16)$$

where $\boldsymbol{J}_F^{n+1}$ is the fluid system Jacobian relating a change in displacement to a change in interface tractions. The system is however underdetermined, with only k available observations (with each observation of size m) to describe the full interface Jacobian of size $m \times m$, where m is the total number of interface degrees of freedom (DOFs), with $m \gg k$. As such, there are infinite possible solutions which will satisfy (16). One way in which to obtain a uniquely defined system is through the minimum norm solution of (16), i.e. solving for the minimum $||\boldsymbol{J}_F||$ subject to the constraint $\boldsymbol{J}_F^{n+1} \Delta \boldsymbol{D}_F^{n+1} = \Delta \boldsymbol{F}_F^{n+1}$. Solving for $\boldsymbol{J}_F^{n+1}$ via a minimum norm solution will add curvature to the approximation in only the k directions in which information is available, leaving all other directions within $\boldsymbol{J}_F^{n+1}$ unchanged. The minimum norm yields an approximate Jacobian of the following form

$$\boldsymbol{J}_F^{n+1} = \Delta \boldsymbol{F}_F \left([\Delta \boldsymbol{D}_F]^T \Delta \boldsymbol{D}_F \right)^{-1} [\Delta \boldsymbol{D}_F]^T, \quad (17)$$

where the superscript $n+1$ was once again dropped from the observation matrices in order simplify notation.

The approximate Jacobian given by (17) is now only based on observations from the current time step. The approximation can be greatly improved by reusing information from previous time steps. One manner in which to do so, would be to include interface information from previous time steps in the observation matrices (14) and (15). Doing so will in fact yield the exact same Jacobian approximation as provided by the popular least-squares, reduced order modelling (IBQN-LS) method of Degroote *et al.* [10] (discussed in Section 4.3). While the method has on several occasions been demonstrated to be a capable coupling method, it does suffer from a very problem dependent choice of how far in time histories should be retained. We therefore would like a Jacobian approximation method that can retain information from multiple time steps without appending additional information.

The MVQN method is therefore based on an iterative updating scheme. Starting with an initial Jacobian from the previous time step $\boldsymbol{J}^n$, we iteratively update the Jacobian based on an update rule of the form

$$\boldsymbol{J}_{F,k+1}^{n+1} = \tilde{\boldsymbol{J}}_{F,k+1}^{n+1} + \boldsymbol{J}_F^n, \quad (18)$$

where $\tilde{\boldsymbol{J}}_{F,k+1}^{n+1}$ denotes the Jacobian update approximation. We can then construct a secant equation for $\tilde{\boldsymbol{J}}_{F,k+1}^{n+1}$

$$\tilde{\boldsymbol{J}}_{F,k+1}^{n+1} \Delta \boldsymbol{D}_F^{n+1} = \left(\Delta \boldsymbol{F}_F^{n+1} - \boldsymbol{J}_F^n \Delta \boldsymbol{D}_F^{n+1} \right). \quad (19)$$

Solving for the minimum norm of $\tilde{\boldsymbol{J}}_{F,k+1}^{n+1}$ subject to the constraint of (19) we obtain a new Jacobian update rule:

$$\boldsymbol{J}_{F,k+1}^{n+1} = \boldsymbol{J}_F^n + \left(\Delta \boldsymbol{F}_F^{n+1} - \boldsymbol{J}_F^n \Delta \boldsymbol{D}_F^{n+1} \right) \left(\left(\Delta \boldsymbol{D}_F^{n+1} \right)^T \Delta \boldsymbol{D}_F^{n+1} \right)^{-1} \left(\Delta \boldsymbol{D}_F^{n+1} \right)^T. \quad (20)$$

The update equation (20) bears a marked resemblance to the classical rank-1 Broyden's update method (discussed briefly in Section 4.2). It is possible to reformulate (20) as the Jacobian update which minimises $\left|\left| \boldsymbol{J}^{n+1} - \boldsymbol{J}^n \right|\right|$, which closely relates to Broyden's method, where the update is based on minimising the difference between two successive iterates$\left|\left| \boldsymbol{J}_k^{n+1} - \boldsymbol{J}_{k-1}^{n+1} \right|\right|$.

The primary difference between our proposed multi-vector update and Broyden's update method is the ability of the multi-vector scheme to exactly reproduce all the information from the current time step, as opposed to the last observation of the current time step only. Information from preceding time steps is then matched in a minimum norm sense only, and is hence naturally less emphasized in the approximation. An interesting effect thereof is a certain inherent consistency with Newton's method, where the non-linear FSI equations are linearised about the current time step. Furthermore, if newer information from the current time step falls along the same directions as previous information, the older information is replaced, thereby entirely removing the possibility of contradictory information.

In a similar fashion, using interface tractions transferred to the solid solver, we obtain k interface displacements and can therefore construct a similar Jacobian for the solid solver

$$\boldsymbol{J}_{S,k+1}^{n+1} = \boldsymbol{J}_S^n + \left(\Delta \boldsymbol{D}_S^{n+1} - \boldsymbol{J}_S^n \Delta \boldsymbol{F}_S^{n+1}\right) \left(\left(\Delta \boldsymbol{F}_S^{n+1}\right)^T \Delta \boldsymbol{F}_S^{n+1}\right)^{-1} \left(\Delta \boldsymbol{F}_S^{n+1}\right)^T, \quad (21)$$

where the subscript $_S$ now denotes information with regards to the solid field solver.

A brief summary of the method is outlined in Algorithm 1, attached as an appendix. Because the MVQN method requires that at least two coupling iterations have been performed in order to construct a Jacobian approximation, for the first iteration of the first time step we make use of a fixed point iteration scheme with relaxation factor ω:

$$\boldsymbol{d}_1^{S,0} = \omega \tilde{\boldsymbol{d}}_1^{S,0} + (1-\omega)\, \boldsymbol{d}_0^{S,0}. \quad (22)$$

The Jacobian update rule (20) further requires the availability of $\boldsymbol{J}_F^n$. Since we do not have an available starting Jacobian, we set $\boldsymbol{J}_F^0 = [\boldsymbol{0}]$ and $\boldsymbol{J}_S^0 = [\boldsymbol{0}]$ for the first iteration in time step 1.

Remark 1: The convergence tolerance of each of the field solvers should be lower than the FSI coupling tolerance. If this is not true, it remains possible for interface information from the current time step to be contradictory within finite precision.

Remark 2: It is advised that the $()^{-1}$ required by the solution of the Jacobian update equations (20) and (21) not be solved using matrix inversion but rather through matrix factorisation. While the DOFs of the matrices $\left((\Delta \boldsymbol{D})^T \Delta \boldsymbol{D}\right)$ and $\left((\Delta \boldsymbol{F})^T \Delta \boldsymbol{F}\right)$ are sufficiently small (of size $[k \times k]$) to warrant being solved using matrix inversion it poses the risk of compounding matrix ill-conditioning if present.

Remark 3: The number of vectors in $\Delta \boldsymbol{D}$ and $\Delta \boldsymbol{F}$ should never exceed the number of interface DOFs. If this is the case, the secant equation becomes over-determined for which the minimum norm solution is no longer valid. The convergence behaviour of the multi-vector update method is sufficiently good that this should rarely be the case, except for systems with very small interface DOFs (for example a 1D FSI problem). Should the number of retained vectors exceed the number of interface DOFs, the Jacobians in (20) and (21) should be updated and the observation matrices (14) and (15) cleared. The convergence behaviour will be minimally affected if the problem DOFs are indeed sufficiently small, such that a premature update is necessary. An alternative (though not implemented for purposes of this paper) would be to use subspace methods like proper orthogonal decomposition, or singular value decomposition to remove the over-defined subspace information, or solving for the least squares Jacobian.

4.2. Broyden's Method

Broyden's method is a classical rank-1 update quasi-Newton method. The method was originally developed for the solutions of systems of non-linear equations for which the computation of the Jacobian is excessively expensive. Unlike other popular rank-1 update quasi-Newton methods (for example DFP, BFGS, SR1 etc.), Broyden's method places no restrictions on the update to the system Jacobian to be either symmetric or positive definite (in the case of approximating the system Hessian).

To the best of the authors' knowledge, Broyden's method has not yet been applied to realistic FSI problems. Considering the widespread application and large body of research focusing on black-box coupling techniques it is perhaps surprising that Broyden's method or other similar rank-1 quasi-Newton methods have received little to no attention. One noteworthy exception is the work of Haelterman [19] in which the numerical properties of several quasi-Newton and Krylov methods, including Broyden's method, were investigated for coupled problems. The numerical experiments were however limited to relatively simple 1D test cases.

Broyden's method requires that at least two iterations have been performed. A secant equation can then be constructed based on the difference between the two iteration vectors

$$\boldsymbol{J}_{F,k}^{n+1} \Delta \boldsymbol{d}_{F,k}^{n+1} = \Delta \boldsymbol{f}_{F,k}^{n+1}, \quad (23)$$

where $\Delta \boldsymbol{f}_{F,k}^{n+1} = \boldsymbol{f}_{F,k}^{n+1} - \boldsymbol{f}_{F,k-1}^{n+1}$ and $\Delta \boldsymbol{d}_{F,k}^{n+1} = \boldsymbol{d}_{F,k}^{n+1} - \boldsymbol{d}_{F,k-1}^{n+1}$. In Broyden's method [6] the underdetermined system with regards to $\boldsymbol{J}_{F,k}$ is solved by minimising $||\boldsymbol{J}_k - \boldsymbol{J}_{k-1}||$ which results in a Jacobian update formula

$$\boldsymbol{J}_{F,k}^{n+1} = \boldsymbol{J}_{F,k-1}^{n+1} + \frac{\Delta \boldsymbol{f}_{F,k}^{n+1} - \boldsymbol{J}_{F,k-1}^{n+1} \Delta \boldsymbol{d}_{F,k}^{n+1}}{\left|\left|\Delta \boldsymbol{d}_{F,k}^{n+1}\right|\right|^2} \left(\Delta \boldsymbol{d}_{F,k}^{n+1}\right)^T. \quad (24)$$

As with the MVQN method, because access to an initial Jacobian is not available, the initial Jacobian is set to zero $\left(\boldsymbol{J}_F^0 = [\boldsymbol{0}]\right)$ for the first iteration of the first time step. The solid field solver Jacobian may equivalently be approximated as

$$\boldsymbol{J}_{S,k}^{n+1} = \boldsymbol{J}_{S,k-1}^{n+1} + \frac{\Delta \boldsymbol{d}_{S,k}^{n+1} - \boldsymbol{J}_{S,k-1}^{n+1} \Delta \boldsymbol{f}_{S,k}^{n+1}}{\left|\left|\Delta \boldsymbol{f}_{S,k}^{n+1}\right|\right|^2} \left(\Delta \boldsymbol{f}_{S,k}^{n+1}\right)^T. \quad (25)$$

4.3. Least squares, reduced order modelling quasi-Newton method

One of the more popular black-box quasi-Newton coupling schemes is the 'Interface Block Quasi-Newton with an approximation for the Jacobians from Least-Squares models' (IBQN-LS) introduced by Vierendeels *et al.* [30]. The basic premise is the construction of approximate Jacobians for the block-Newton equations outlined in (10) and (11) from least squares reduced order models (ROM) of interface observations. A similar method, the 'Inverse approximation of the Jacobian from Least-Squares model' (IQN-LS) introduced by Degroote *et al.* [9] follows the same premise as the IBQN-LS model but only applied to the interface residual equation (13). There are several publications available on comparisons of the IBQN-LS and IQN-LS method [10, 11], including a comparison of the IQN-LS method to a full monolithic solver [9].

As in Section 4.1, let us consider that we have completed k coupling FSI iterations, where we can then construct in the same fashion as before a set of differencing observation matrices

$$\Delta \boldsymbol{D}_F^{n+1} = [\boldsymbol{d}_k - \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k-1} - \boldsymbol{d}_{k-2}, ..., \boldsymbol{d}_2 - \boldsymbol{d}_1] \tag{26}$$

$$\Delta \boldsymbol{F}_F^{n+1} = \left[\boldsymbol{f}_k - \boldsymbol{f}_{k-1}, \boldsymbol{f}_{k-1} - \boldsymbol{f}_{k-2}, ..., \boldsymbol{f}_2 - \boldsymbol{f}_1\right]. \tag{27}$$

Based on the work of Vierendeels *et al.* [30], an approximation of the interface change in displacement can be written as a linear combination of previous observations via a set of linear expansions coefficients $\boldsymbol{\alpha}$

$$\Delta \boldsymbol{d}_F = \Delta \boldsymbol{D}_F \boldsymbol{\alpha}. \tag{28}$$

Similarly, via the same linear expansion coefficients a change in interface tractions may be approximated as

$$\Delta \boldsymbol{f}_F = \Delta \boldsymbol{F}_F \boldsymbol{\alpha}. \tag{29}$$

Solving for $\boldsymbol{\alpha}$ via the least squares solution of (28) gives

$$\boldsymbol{\alpha} = \left(\Delta \boldsymbol{D}^T \Delta \boldsymbol{D}\right)^{-1} \Delta \boldsymbol{D}^T \Delta \boldsymbol{d} \tag{30}$$

which, when inserted into equation (29), results in a relation between the change in interface tractions to a change in interface displacements

$$\Delta \boldsymbol{f}_F = \Delta \boldsymbol{F} \left(\Delta \boldsymbol{D}^T \Delta \boldsymbol{D}\right)^{-1} \Delta \boldsymbol{D}^T \Delta \boldsymbol{d}_F, \tag{31}$$

where $\boldsymbol{J}_F = \Delta \boldsymbol{F} \left(\Delta \boldsymbol{D}^T \Delta \boldsymbol{D}\right)^{-1} \Delta \boldsymbol{D}^T$ can be viewed as the approximate fluid Jacobian.

The proficiency of the least squares quasi-Newton Jacobian can be greatly improved by including historical observations from previous time steps, as noted by Degroote *et al.* [10]. Degroote *et al.* therefore suggests appending the converged solution observation matrices from q previous time steps such that:

$$\Delta \boldsymbol{D} = \left[\Delta \boldsymbol{D}^{n+1}, \Delta \boldsymbol{D}^{n}, ..., \Delta \boldsymbol{D}^{n+1-q}\right] \tag{32}$$

$$\Delta \boldsymbol{F} = \left[\Delta \boldsymbol{F}^{n+1}, \Delta F^{n}, ..., \Delta \boldsymbol{F}^{n+1-q}\right]. \tag{33}$$

The convergence performance of the IBQN-LS method is however strongly dependent on the choice of how far in time, q, histories are retained. The choice of q is in fact a problem dependant heuristic which has to be chosen prior to runtime, with no *a priori* way of determining the optimal choice of q. Most problems become unstable unless q is chosen to be small, while other problems may benefit from a much larger choice. Degroote *et al.* [10] attribute this behaviour to the fact that histories from far back in time are no longer relevant for newer time steps. While there is certainly some truth to this, we will show in Section 5 via test problems that changing the time step size in general does not result in the optimal choice of q changing.

A far more complete explanation can be found by reformulating the IBQN-LS method as a quasi-Newton method where the Jacobian approximation is based on minimising $||\boldsymbol{J}||$ subject to the constraints of the secant equation $\boldsymbol{J}_F \Delta \boldsymbol{D}_F = \Delta \boldsymbol{F_F}$. Trying to match information from multiple time steps leads to the possibility of contradicting information. To put it simply, imagine the fluid solver is provided with two identical changes in displacement. It is now possible that for two different time steps, two different changes in interface tractions are observed. Solving for the minimum norm Jacobian with internally contradicting information leads to an inability to exactly match either of these observations, while at least one of these observations may be pertinent to the current solution space.

An equally relevant problem for minimum norm solutions is linear dependence. Linearly dependent vectors manifests itself in ill-conditioned matrices which affect the overall solution accuracy and the loss of precision. While linearly dependent observations remain possible for both the newly proposed MVQN and Broyden's method, the probability of obtaining linearly dependent vectors increase significantly with an increase in the number of appended (retained) vectors from previous time steps. There are several numerical techniques available to minimise the effects of ill-conditioning, see for example [5], where the IBQN-LS method was augmented through the use of proper orthogonal decomposition. While this is potentially beneficial for some problems by enabling the method to remain well conditioned and stable for larger choices of q, it significantly increases the computational complexity while providing no solution to the problem of contradictory observations.

It is important to note that the set of remarks highlighted in Section 4.1 is also in general applicable here for the IBQN-LS method.

4.4. Aitken's dynamic relaxation

Aitken's Δ^2 dynamic relaxation method has attained a large degree of popularity within the field of partitioned FSI coupling primarily due to its simplicity and relative efficiency while facilitating the use of black-box field solvers.

Fixed point iteration schemes iterate on the field operators until the interface displacement residual drops below some given tolerance ϵ where

$$\boldsymbol{r}_k^{n+1} = \boldsymbol{d}_k^{n+1} - \boldsymbol{d}_{k-1}^{n+1}. \tag{34}$$

A displacement update including a relaxation factor ω then becomes

$$\boldsymbol{d}_k^{n+1} = \boldsymbol{d}_k^{n+1} + \omega \boldsymbol{r}_k^{n+1}, \tag{35}$$

where as before, k indicates the current coupling iteration and $n+1$ the new time step. For complex FSI problems, for example the class of problems encountered in biomedical problems, fixed relaxation parameters is often insufficient to obtain convergence. Aitken's method augments the fixed point iterates using dynamic relaxation, where the relaxation parameter, ω, is modified at the start of each iteration using the displacement results from two preceding coupling iterates:

$$\omega_{k+1} = -\omega_k \frac{\left(\boldsymbol{r}_{k-1}\right)^T \left(\boldsymbol{r}_k - \boldsymbol{r}_{k-1}\right)}{\left(\boldsymbol{r}_k - \boldsymbol{r}_{k-1}\right)^T \left(\boldsymbol{r}_k - \boldsymbol{r}_{k-1}\right)}. \tag{36}$$

We include Aitken's method in our discussion on quasi-Newton methods by illustrating that Aitken's method is in fact the simplest form of a quasi-Newton method available, where the system Jacobian is approximated with the results from only two coupling iterations. To demonstrate this, let us return to quasi-Newton residual equation (13) solving for the updated displacement

$$\boldsymbol{d}_{\Gamma,k+1}^{n+1} = \boldsymbol{d}_{\Gamma,k}^{n+1} - \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} \boldsymbol{r}_{\Gamma,k+1}^{n+1}. \tag{37}$$

Given the results from two previous iterations, we can construct an approximation for the inverse Jacobian via a secant or FD approximation

$$\Delta \boldsymbol{d} = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} \Delta \boldsymbol{r}, \tag{38}$$

where $\Delta \boldsymbol{r} = \boldsymbol{r}_{\Gamma,k+1} - \boldsymbol{r}_{\Gamma,k}$ and $\Delta \boldsymbol{d} = \boldsymbol{d}_{\Gamma,k} - \boldsymbol{d}_{\Gamma,k-1}$. Given the differencing vectors we can then compute the minimum $\left|\left|\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1}\right|\right|$ satisfying the secant equation (38) providing a displacement update of

$$\begin{aligned}\boldsymbol{d}_{\Gamma,k+1}^{n+1} &= \boldsymbol{d}_{\Gamma,k}^{n+1} + \\ &\frac{\left(\boldsymbol{r}_k^{n+1}\right)^T \left(\boldsymbol{r}_{k+1}^{n+1} - \boldsymbol{r}_k^{n+1}\right)}{\left(\boldsymbol{r}_{k+1}^{n+1} - \boldsymbol{r}_k^{n+1}\right)^T \left(\boldsymbol{r}_{k+1}^{n+1} - \boldsymbol{r}_k^{n+1}\right)} \left(-\boldsymbol{r}_{\Gamma,k+1}^{n+1}\right).\end{aligned} \tag{39}$$

It is easy to show that this yields an identical formulation to Aitken's method seen by inserting the dynamic relaxation parameter from (36) into (35).

While Aitken's method is simple to implement, the limitation of the method stems primarily from its inability to re-use more iterate data. In contrast, the three other quasi-Newton methods already discussed allows for the re-use of some or all interface data acquired through the course of a simulation.

5. Test Problems

In this section we analyse the quasi-Newton methods applied to four incompressible flow benchmark FSI problems. The problems cover a wide spectrum of problem classes and are all considered to be strongly coupled FSI problems, with large added mass ratios requiring the use of implicit coupling strategies.

The FSI problems are solved using OpenFOAM [1] as the fluid field solver and Calculix [14] for the solid domain. While both field solvers are open source, and hence the source code is available, we treat both as black-box solvers. The focus of the analyses is on the comparison and computational efficiencies of the coupling strategies; issues surrounding solution accuracy and order of temporal convergence are not investigated. The time integration used for the fluid solver is an implicit backward-Euler and generalised alpha for the solid domain. It is therefore to be expected that the overall coupled FSI system's temporal solution order be limited by the fluid solver. The fluid solver is based on the *pimpleDyMFoam* solver, and can therefore comfortably handle Courant numbers as high as 2.

The interface load and motion transfer is performed using radial basis function interpolation. Interface tractions are transferred in a consistent fashion, thereby guaranteeing that a constant stress state (such as encountered in a constant stress patch test) can be exactly transferred regardless of the mesh mismatch along the interface.

The approximate Jacobians for both the fluid and solid field solvers are constructed along the solid interface. This is done since the number of DOFs along the solid interface is typically smaller; the coupling performance is however independent of this choice. It is also worth mentioning, while not documented in the results to follow, that the coupling behaviour of the quasi-Newton methods are largely independent of the mesh sizes chosen for the discretisation of each of the domains. The inverse $()^{-1}$ computations required for the MVQN and IBQN-LS methods are solved using QR factorization. Furthermore, we adopt the notation IBQN-LS(q) to indicate that information from q previous time steps is retained.

It is often suggested that the interface position be approximated at the start of each time step via the extrapolation of a higher order polynomial in time (see for example [9]). In this paper we are primarily interested in the ability of the quasi-Newton methods to approximate the system behaviour, including the validity of the Jacobians for reuse over multiple time steps. For this reason, the interface position at the start of each time step is not approximated but set to the converged solution from the previous time step, $\boldsymbol{d}_0^{n+1} = \boldsymbol{d}^n$.

The coupling convergence criterion used for all the benchmark problems to follow is

$$\frac{|\boldsymbol{d}_{\Gamma,k+1} - \boldsymbol{d}_{\Gamma,k}|}{\sqrt{m}} \leq \epsilon, \tag{40}$$

where m is the number of DOFs along the interface and is included to remove the dependency of the solution residual to the interface mesh size. The choice of ϵ is varied for the test problems to illustrate performance dependencies on convergence tolerance, if any are present. The convergence criterion is varied from suitable values at which reasonable results can be obtained to 10^{-9} across all problems to gain insights into the asymptotic convergence behaviour. In all cases the convergence tolerance for each of the separate field solvers is set to at least an order of magnitude lower. The initial relaxation factor required for the first iteration of the first time step is set to $\omega = 0.001$ for all test problems and quasi-Newton methods. Lastly, the time step sizes chosen for each of the problems are chosen to be similar to that encountered in literature, where the same problems were solved using monolithic and/or partitioned solution schemes with exact Jacobians. We therefore aim to demonstrate that the approximate Jacobians analysed in this paper are sufficiently accurate to provide stable solutions despite the comparatively large time steps and fine convergence tolerances.

5.1. Flow Induced Oscillating Flexible Beam

The first problem we analyse is flow around a flexible beam attached to a fixed rigid square. The problem considers large deformations of the beam induced by oscillating vortices formed by flow around a square bluff body. The problem was first introduced by Wall [31], and represents the simplest of the test problems we investigate in this paper; the solid density of the beam is approximately two orders of magnitude larger than the fluid density. The added mass effect, while present, is minimal. The oscillating flexible beam problem is an ideal example of where partitioned solution schemes can be competitive in comparison to monolithic procedures.

The problem set-up is depicted in Figure 1. The beam is 0.06cm thick and 4cm long, with linear elastic material properties of Young's modulus $E = 2.5 \times 10^6$g/(cm s^2), density of $\rho = 0.1$g/cm^3 and a Poisson's ratio $\nu = 0.35$. The material properties result in a fundamental frequency associated with an oscillation period of approximately 0.33 seconds. The square bluff body has side lengths of 1.0cm, with a domain inlet velocity of 51.3cm/s. Slip conditions are applied to the wall boundaries with zero-gradient velocity outlet conditions with the outlet pressure set to 0. The fluid flow density is set to $\rho_f = 1.18 \times 10^{-3}$g/cm^3 with a viscosity of $\mu = 1.82 \times 10^{-4}$g/(cm s) as per [31].

The mesh used for the analysis is shown in Figure 2. A total of 20 quadratic, full integration solid elements along with 3104 linear triangular flow elements are employed for the purposes of the analysis. Mesh movement is performed using radial basis function interpolation as per [8] where the entire fluid domain is deformed without any requirements for re-meshing. The tip displacement for three time steps sizes is shown in Figure 3.

A summary of the performance of each of the quasi-Newton coupling methods is given in Table 1 for different

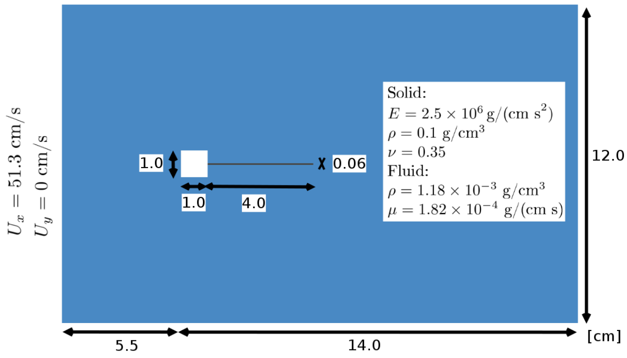
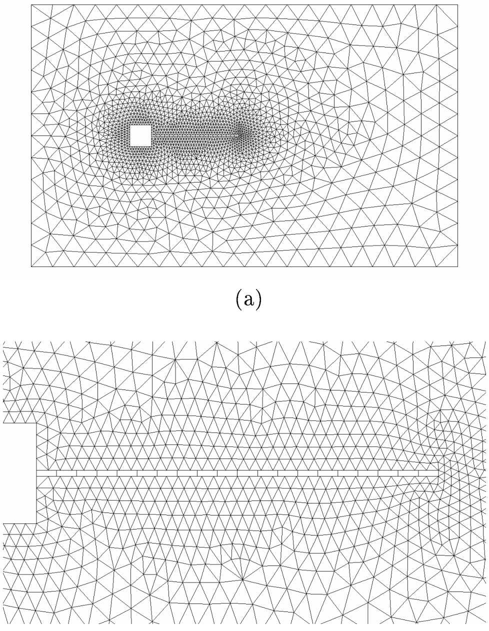

$U_x = 51.3$ cm/s
$U_y = 0$ cm/s
1.0
1.0
4.0
0.06
Solid:
$E = 2.5 \times 10^6$ g/(cm s^2)
$\rho = 0.1$ g/cm^3
$\nu = 0.35$
Fluid:
$\rho = 1.18 \times 10^{-3}$ g/cm^3
$\mu = 1.82 \times 10^{-4}$ g/(cm s)
12.0
5.5
14.0
[cm]

Figure 1: Flexible tail behind rigid square test problem set-up.



(a)

(b)

Figure 2: Domain discretisation for the oscillating flexible tail test problem. Total of 3104 linear, triangular fluid elements with 20 quadratic, full integration solid elements.

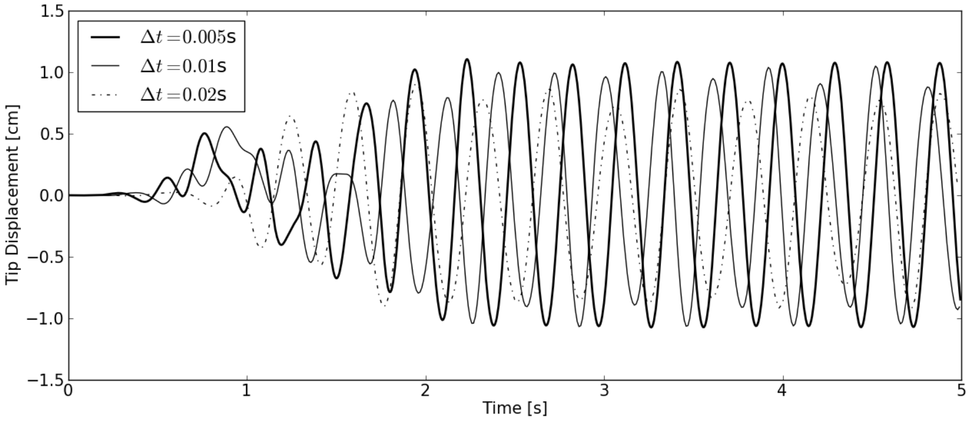
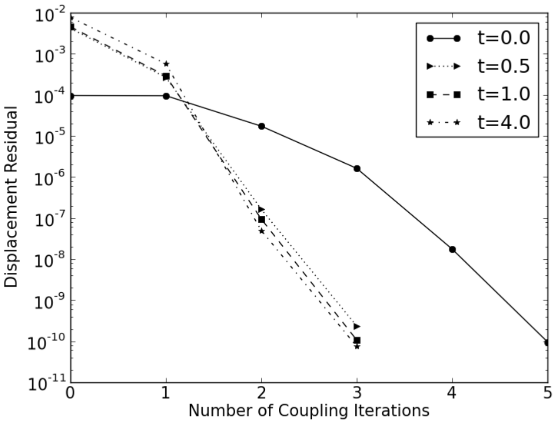


Figure 3: Beam tip displacement for the flexible tail problem shown for 3 different time step sizes.

time steps sizes and convergence tolerances. The average, minimum and maximum number of coupling iterations required for convergence are outlined along with the normalised relative CPU time. For this particular problem all the methods exhibit similar performance, with the MVQN being the most efficient of the four quasi-Newton methods. All four methods are convergent for large time steps and small convergence tolerances. Similarly, all the methods present favourable properties when comparing the mean number of iterations required to converge for the same step size for a change in ϵ from 10^{-6} to 10^{-9}, on average requiring between 1 to 2 additional iterations. Specifically, the MVQN requires an additional 1.14 iterations.

The typical convergence rates for the MVQN method at different times are shown in Figure 4. The benefit of additional information is apparent here when comparing the convergence behaviour of the first to later time steps. Aside from the very first time step, the displacement residual decreases by three orders of magnitude for each additional iteration (super-linear). Within the limit of reasonable convergence criteria, the quasi-Newton method is directly comparable (in terms of the required number of coupling iterations) to a Newton based coupling scheme.



(a)

Figure 4: Typical MVQN convergence rates for flexible tail problem at different times for $\Delta t = 0.02$s, $\epsilon = 10^{-9}$.

5.2. 1D dynamic piston-channel problem

We analyse here a dynamic 1D piston-channel problem. The problem layout is described in Figure 5, where a 10m long fluid domain is forced out of the domain by the acceleration of a unit by unit solid block. The coupling between the solid and fluid domains is strong, and requires the use of an implicit coupling scheme. The coupling strength is sufficiently strong that simple iterative schemes are insufficient to obtain convergence.

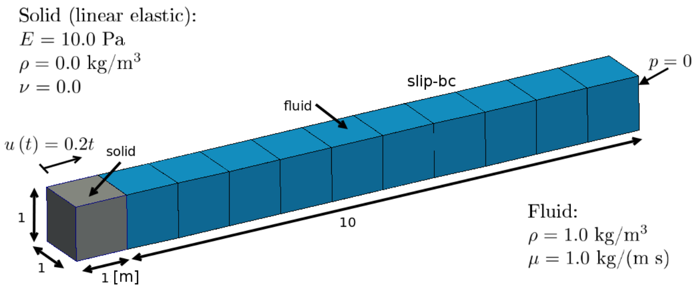


Figure 5: 1D piston-channel problem set-up.

The fluid domain consists of a fluid with density $\rho = 1.0$kg/m^3, with a viscosity of $\mu = 1.0$kg/(m s), all wall boundaries are no-slip conditions with an exit pressure of $p = 0$. In this paper, we make use of a linear elastic, small strain solid with $\rho = 0$kg/m^3, $E = 10$Pa and $\nu = 0$, with a prescribed velocity of $u(t) = 0.2t$ [m/s]. The use

Table 1: Comparison of the number of iterations and relative computational time for the flexible tail problem, for different time steps sizes and convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 0.02, \epsilon = 10^{-6}$				$\Delta t = 0.02, \epsilon = 10^{-9}$				$\Delta t = 0.005, \epsilon = 10^{-9}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	3.16	3	5	1.00	4.30	4	6	1.00	4.27	3	5	1.00
Aitken	6.12	5	7	1.89	8.76	7	10	1.96	9.91	9	11	2.25
Broyden	3.34	3	5	1.06	4.66	4	6	1.08	4.70	4	6	1.11
IBQN-LS(0)	3.90	3	5	1.25	5.52	5	6	1.28	5.70	4	6	1.33
IBQN-LS(1)	3.57	3	5	1.12	5.02	4	6	1.17	5.00	3	6	1.17
IBQN-LS(2)	3.84	3	6	1.24	5.48	4	10	1.27	5.27	3	8	1.26
IBQN-LS(3)	4.2	3	7	1.33	5.25	4	7	(5/250)	5.89	3	9	1.41
IBQN-LS(4)	4.50	3	8	1.43	5.25	4	7	(5/250)	6.21	3	16	1.51
IBQN-LS(5)	5.30	3	11	(10/250)	5.25	4	7	(5/250)	6.73	3	29	1.59

of small strain linear elasticity only slightly decreases the complexity of the problem as opposed to for example [3] where a hyper-elastic material model is employed, but does allow for the construction of a simplified expression for the 1D problem. While the simplified expression does not constitute an analytical expression it is sufficiently cheap to be computed to a high degree of accuracy using any numerical integration scheme with very small time steps.

Following the work of Suliman [27], let us consider the piston-channel problem as a 1D spring-mass system, where the piston acts as a linear spring, and the incompressible fluid as a variable mass system. Based on the balance of forces along the interface, it is possible to construct an expression for the interface displacement d_Γ and interface velocity u_Γ

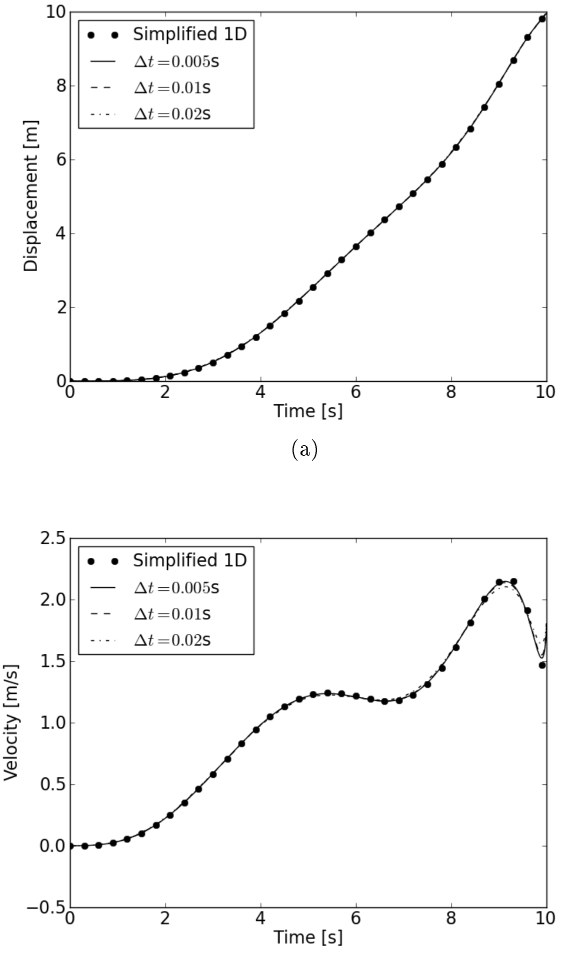
$$\frac{\partial d_\Gamma}{\partial t} = u_\Gamma, \tag{41}$$

$$\frac{\partial u_\Gamma}{\partial t} = \frac{10\left(d_\Gamma - 0.1t^2\right)}{\left(d_\Gamma - 10\right)}. \tag{42}$$

The force of the solid and fluid domains on the interface can be described by $F_{\Gamma,s} = 10\left(d_\Gamma - 0.1t^2\right)$ and $F_{\Gamma,f} = \left(d_\Gamma - 10\right)$ respectively.

The results for the FSI simulation for various time steps, along with the simplified 1D results are shown in Figure 6. It is important to note that the choice of interface coupling technique does not influence the results themselves. The only impact of the different coupling techniques are whether the problem in fact converges within a time step, and the number of coupling iterations required to do so.

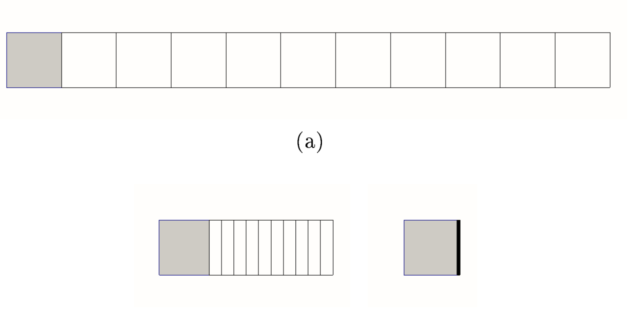
As a side note, the problem requires a capable fluid solver which can handle fairly large Courant numbers. Towards the end of the simulation, the fluid domain is compressed to such an extent that the fluid elements become very small. To illustrate this consider Figure 7 showing the fluid domain deformation at three different times.



(a)

(b)

Figure 6: Results for 1D piston-channel problem for various time steps compared to the simplified 1D expression for interface (a) displacement and (b) velocity.



(a)

(b) (c)

Figure 7: Domain deformation at simulation time (a) 0s, (b) 8.85s and (c) 10.0s.

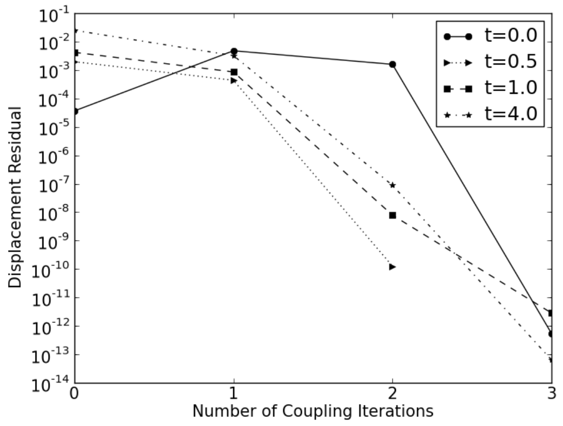
The typical convergence rates for the MVQN method at multiple time steps are shown in Figure 8. Even though the interface DOFs are small, the benefit in retaining information from multiple time steps can still be seen, with the progressive improvement in convergence rates compared to the first time step. Once again, aside from the first time step, super-linear asymptotic convergence rates are observed.

A summary of the results of the quasi-Newton methods is shown in Table 2. By comparing the results, it is apparent that the MVQN method provides superior results, especially in the presence of large time steps and a fine convergence tolerance.

Providing an explanation why the other 3 schemes perform so poorly is not straight forward. Firstly, the 1D piston-channel problem is a very strongly coupled problem. The 1D problem is well known to be more challenging than the 2D flexible beam problem analysed in Section 5.1 (comparing the results in Table 2 to those in Table 1 confirms this). Due to the complexity of the problem a very accurate Jacobian approximation is required. In this regard, the approximations provided by both Broyden and Aitken is simply not accurate enough to provide convergent results for large time steps and very fine convergence tolerances.

Explaining why the IBQN-LS method fails is however slightly more involved. While the method has been demonstrated in numerous publications to provide sufficiently accurate Jacobian approximations for most problems, it clearly struggles in this instance. The reason for this can be traced back to the minimum norm approximation. While the problem is 1D in nature, we solve the problem in three dimensions; along the interface we therefore have 3 DOFs. If more than 3 observation vectors are used to generate the Jacobian approximation, the system becomes over-defined, and the minimum norm solution to the secant conditions become a least squares solution. In other words, neither of the observations that were used to generate the approximation can be exactly matched, with the end result that the convergence behaviour deteriorates.

To rectify the over-defined problem for the IBQN-LS method would require removing vectors from the list of retained observations vectors. Unfortunately, which vectors to drop is not obvious. Intuition would stipulate dropping the oldest information. Often however this will result in dropping information which is critical to the current spatial and temporal location, and again cause a deteriorated convergence behaviour. On the other hand, the updating scheme for the MVQN method bypasses this problem. Whenever the number of observation vectors in the update approximations in (20) or (21) exceeds the number of interface DOFs, an update can be performed and all retained vectors be cleared. This effectively resets the Jacobian approximation without loosing any of the information which has been gained up to that point. For a true 1D problem (i.e. 1 DOF along Γ_{FSI}) this would render the MVQN method identical to Broyden's method.



(a)

Figure 8: Typical MVQN convergence rates for 1D piston-channel problem at different times, $\Delta t = 0.02$s, $\epsilon = 10^{-9}$.

5.3. Flexible Tube, $\rho_s = 1.2g/cm^3$ and $\rho_f = 1.0g/cm^3$

Let us now turn our attention to a 3D internal flow problem through a flexible tube proposed in [18], which is inspired by the type of flow problems encountered in haemodynamics. The density ratios of the fluid and solid are near unity, which in conjunction with internal incompressible flow results in a very strongly coupled FSI problem. Realistic results can however be obtained for relatively small discretisation DOFs and as a result has gained a lot of popularity as an FSI benchmark problem.

The flexible tube has a length of $l = 5$cm, with an inner and outer radius of $r_i = 0.5$cm and $r_0 = 0.6$cm respectively. A hyper-elastic, neo-Hookean material model with Young's modulus of $E = 3 \times 10^6$dynes/cm^2, density$\rho = 1.2$g/cm^3 and Poisson's ratio of 0.3 is modelled with 96 twenty noded quadratic brick elements with full integration. The fluid flow has a density of $\rho = 1.0$g/cm^3 and viscosity of $\mu = 0.03$P which is modelled using 1600 linear elements. In order to avoid spurious pressure waves which

Table 2: Comparison of the number of iterations and relative computational time for the 1D piston-channel problem, for different time steps and convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 0.02, \epsilon = 10^{-6}$				$\Delta t = 0.02, \epsilon = 10^{-9}$				$\Delta t = 0.01, \epsilon = 10^{-6}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	3.00	3	4	1.00	3.53	3	5	1.00	3.00	3	4	1.00
Aitken	16.12	9	67	3.95	–	–	–	(1/500)	–	–	–	(1/1000)
Broyden	3.53	3	4	1.20	4.47	3	24	(344/500)	3.01	3	4	1.01
IBQN-LS(0)	3.96	3	4	1.36	4.13	4	5	(170/500)	3.95	3	4	1.31
IBQN-LS(1)	4.33	3	11	(489/500)	6.29	3	32	(473/500)	3.93	3	7	1.29
IBQN-LS(2)	4.32	3	9	(449/500)	6.33	3	28	(498/500)	4.09	3	7	1.35

may occur if a high pressure is applied from the start of the simulation due to impacting on the solid tube, we apply a smooth input pressure wave $p(t)$ $\left[\text{dynes/cm}^2\right]$ at the inlet in the form

$$p(t) = \begin{cases} 1.3332 \times 10^4 \left(\left(\sin\left(\frac{2\pi t}{0.003} + \frac{3}{2}\pi\right)+1\right)/2 \right) & \text{if } t < 0.003 \\ 0 & \text{if } t \geq 0.003. \end{cases} \tag{43}$$

The time step size for the simulation is $\Delta t = 0.0001$s, where the convergence behaviour is studied for convergence tolerances ϵ of 10^{-7} and 10^{-9}. The pressure pulse propagation at time steps 0.003s and 0.005s is shown in Figure 9.

Of the problems analysed thus far, the flexible tube benchmark represents a problem where the benefit of retaining many observations from multiple time steps is significant. Consider the summarised results in Table 3. The IBQN-LS(q) method, while capable of producing good convergence behaviour, now requires the retention of interface observations from close to $q = 30$ time steps (in contrast to approximately only 1 required for the flexible beam and 1D piston-channel problems). While the performance of the IBQN-LS(q) method is relatively uniform from a choice of $q = 5$ onwards, the performance when compared to the MVQN method ranges all the way from 15 to 155 percent more expensive depending on the choice of q.

While retaining more information is beneficial, retaining too many vectors (as is the case for $q = 40$ for $\Delta t = 10^{-4}$ and $\epsilon = 10^{-9}$) results in an ill-conditioned system Jacobian due to linearly dependent vectors. The only reason the IBQN-LS(q) is capable of providing convergent results for $q \geq 30$ (where the condition number is already very high) is due to QR factorization. Solving the inverses in equation (31) using singular value decomposition (SVD) or proper orthogonal decomposition (POD) would enable the retention of all the time step information. For the flexible tube problem, retaining all the information does in fact provide noticeable benefit, see for example [5]. While SVD or POD solves issues surrounding ill-conditioning, it resolves little of the problems pertaining to an over-defined system or the presence of contradictory information from multiple time steps (as was encountered in the problems of Section 5.1 and 5.2).

The convergence rates for the MVQN method is shown in Figure 10. In the first time step, the convergence behaviour is not monotonically decreasing. However, as the Jacobian is trained through the course of the simulation the convergence behaviour improves considerably and from time step 2 onwards is comparable to an inexact-Newton scheme. The convergence behaviour of the MVQN method can in fact be likened to the typical convergence behaviours reported by Dettmer *et al.* [13] when selected chain rule terms were omitted from the computation of the exact cross-coupling terms for partitioned FSI problems. This is especially noteworthy considering the quasi-Newton method(s) investigated here allow for the re-use of partitioned black-box field solvers and do not require any exact gradient computations.

5.4. *Flexible Tube, $\rho_s = 0.12g/cm^3$ and $\rho_f = 1.0g/cm^3$*

Let us now reconsider the pressure wave propagation through the flexible tube described above, where the solid density is reduced to $\rho_s = 0.12\text{g/cm}^3$. The fluid density is now almost an order of magnitude higher than the solid density, with the result that the added mass effect is now substantially increased. A summary of the coupling results is shown in Table 4. Of the four coupling methods, only the MVQN and IBQN-LS methods were capable of producing convergent results, with the MVQN method again proving to be the most efficient. Despite the increase in problem complexity, the MVQN method only requires, on average, an additional 1.2 iterations for convergence when compared to the flexible tube with a solid density of $\rho_s = 1.2\text{g/cm}^3$. If we further compare the typical convergence behaviour for the MVQN method in Figure 11 (for $\rho_s = 0.12\text{g/cm}^3$) to Figure 10 (for $\rho = 1.2\text{g/cm}^3$), we observe that the MVQN approximate Jacobians do take longer to train. The resulting Jacobians again exhibit Newton-like convergence behaviour towards the end of the simulation.

While the solution of this problem does not constitute sufficient proof that the MVQN coupling method can

Table 3: Comparison of the number of iterations and relative computational time for pressure propagation through flexible tube where $\rho_s = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$, for different convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 10^{-4}, \epsilon = 10^{-7}$				$\Delta t = 10^{-4}, \epsilon = 10^{-9}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	4.13	3	7	1.00	5.34	4	10	1.00
Aitken	35.39	12	51	7.33	51.76	29	74	8.42
Broyden	4.83	4	12	1.15	6.58	5	14	1.22
IBQN-LS(0)	10.4	7	11	2.55	13.75	10	15	2.60
IBQN-LS(1)	8.52	3	10	2.11	11.58	4	13	2.15
IBQN-LS(5)	5.96	3	7	1.44	8.5	4	10	1.57
IBQN-LS(10)	5.08	3	7	1.23	7.2	4	10	1.34
IBQN-LS(20)	4.89	3	10	1.24	6.41	4	10	1.20
IBQN-LS(30)	4.71	3	10	1.15	6.40	4	10	1.26
IBQN-LS(40)	4.98	3	10	1.24	8.35	4	24	(51/100)

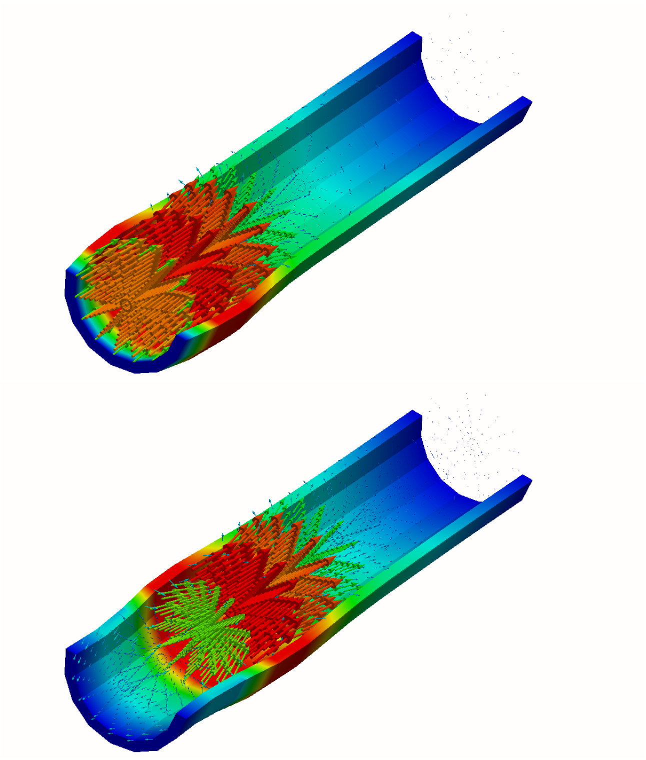
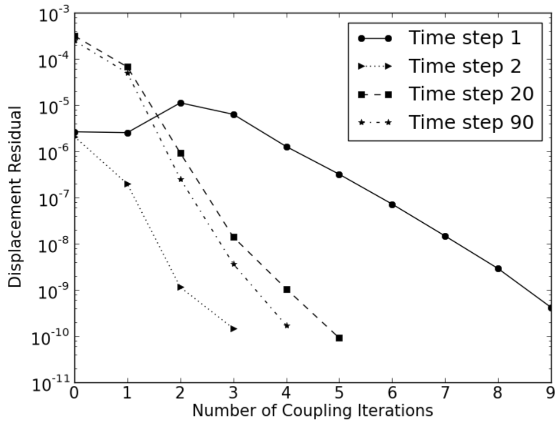
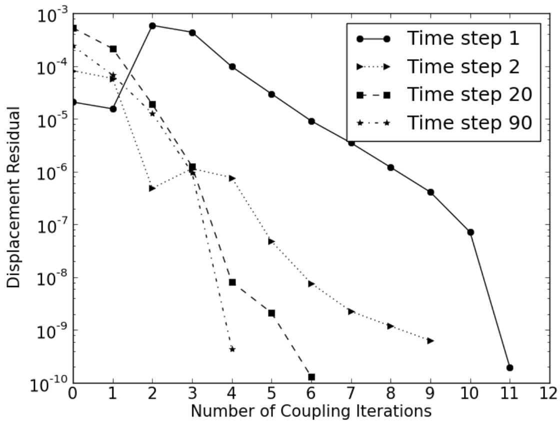


Figure 9: Pressure pulse propagation at 0.003 and 0.005 seconds. Wall displacement amplified 10 times.



(a)

Figure 10: Typical MVQN convergence rates for the flexible tube problem with $\rho_S = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ for $\Delta t = 10^{-4}$s and $\epsilon = 10^{-9}$ at different times.



(a)

Figure 11: Typical MVQN convergence rates for the flexible tube problem with $\rho_S = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ for $\Delta t = 10^{-4}\text{s}$ and $\epsilon = 10^{-9}$ at different times.

be applied to all haemodynamic problems, it does provide some confidence in the applicability of the method to solve strongly coupled FSI problems. It bears further merit when one considers the possibility of accelerating the training of the approximate Jacobian in the first couple of time steps using multi-grid like methods [12].

6. Conclusion

In this paper we introduced a new multi-vector quasi-Newton method for the implicit coupling of partitioned solvers, which facilitates the use of black-box field solvers. We analysed the method across a wide range of strongly coupled FSI benchmark problems for which the method was demonstrated to be both robust and efficient. The method outperformed other well known quasi-Newton methods and can be applied to incompressible FSI problems with high added mass ratios. We compared the method to the least squares, reduced order modelling quasi-Newton method of Vierendeels and Degroote *et al.* We demonstrated that the IBQN-LS method can be reformulated as finding the minimum norm Jacobian which satisfies the interface equations. This in turn, to some extent explains the method's problem dependent behaviour with regards to the choice of the number of time histories, q, to be retained. The new MVQN updating scheme outperformed the IBQN-LS method for all the problems analysed as well as Broyden's method, which in turn showed favourable performance when compared to the IBQN-LS method; Broyden's method was however unable to obtain a convergent solution when the solid density was chosen to be lower than the fluid density. We have further demonstrated, while fixed point iterations with dynamic relaxation is easy to implement, that it is predominantly ill-suited for the coupling of FSI problems with high added mass ratios; this confirms the findings of multiple researchers.

Appendix A: MVQN Algorithm

Algorithm 1 MVQN: Multi-vector update quasi-Newton method applied to the block-Newton equations (10) and (11).

```
$t = 0$; $n = 0$
$\boldsymbol{f}_1^{F,0} = \boldsymbol{F}\left(\boldsymbol{d}_0^0\right)$; $\boldsymbol{f}_1^{S,0} = \boldsymbol{f}_1^{F,0}$; $\boldsymbol{d}_1^{S,0} = \boldsymbol{S}\left(\boldsymbol{f}_1^{S,0}\right)$
$\boldsymbol{r}_1^0 = \boldsymbol{d}_1^{S,0} - \boldsymbol{d}_0^0$
while $t < t_{\text{end}}$ do:
    $k = 0$
    while $||\boldsymbol{r}_{k+1}^{n+1}|| > \epsilon$ do:
        if ($k = 0$ and $n = 0$) then:
            $\boldsymbol{J}_F^0 = [\boldsymbol{0}]$; $\boldsymbol{J}_S^0 = [\boldsymbol{0}]$
            $\boldsymbol{d}_1^{S,0} = \boldsymbol{d}_1^{S,0} + \omega_0 \boldsymbol{r}_1^0$
        else:
            $\boldsymbol{f}_{k+1}^{F,n+1} = \boldsymbol{F}\left(\boldsymbol{d}_k^{F,n+1}\right)$
            if ($k > 0$) then:
                compute $\boldsymbol{J}_F^{n+1}$ using (20)
            solve for $\Delta\boldsymbol{f}$: equation (10)
                i.e. $(\boldsymbol{J}_F\boldsymbol{J}_S - \boldsymbol{I})\Delta\boldsymbol{f} = -\left(\boldsymbol{f}_{k+1}^{F,n+1} - \boldsymbol{f}_k^{S,n+1}\right) - \boldsymbol{J}_F\left(\boldsymbol{d}_k^{F,n+1} - \boldsymbol{d}_k^{S,n+1}\right)$
            $\boldsymbol{f}_{k+1}^{S,n+1} = \boldsymbol{f}_k^{S,n+1} + \Delta\boldsymbol{f}$
            $\boldsymbol{d}_{k+1}^{S,n+1} = \boldsymbol{S}\left(\boldsymbol{f}_{k+1}^{S,n+1}\right)$
            if ($k > 0$) then:
                compute $\boldsymbol{J}_S^{n+1}$ using (21)
            solve for $\Delta\boldsymbol{d}$: equation (11)
                i.e. $(\boldsymbol{J}_S\boldsymbol{J}_F - \boldsymbol{I})\Delta\boldsymbol{d} = -\left(\boldsymbol{d}_{k+1}^{S,n+1} - \boldsymbol{d}_k^{F,n+1}\right) - \boldsymbol{J}_S\left(\boldsymbol{f}_{k+1}^{S,n+1} - \boldsymbol{f}_{k+1}^{F,n+1}\right)$
            $\boldsymbol{d}_{k+1}^{F,n+1} = \boldsymbol{d}_k^{F,n+1} + \Delta\boldsymbol{d}$
        $k = k + 1$
    $t = t + \Delta t$; $n = n + 1$
```

[1] Openfoam: The open source cfd toolbox user guide, version 2.1.0, 2010.

[2] Santiago Badia, Fabio Nobile, and Christian Vergara. Fluid–structure partitioned procedures based on robin transmission conditions. *Journal of Computational Physics*, 227(14):7027–7051, 2008.

[3] Klaus-Jürgen Bathe and Gustavo A Ledezma. Benchmark problems for incompressible fluid flows with structural interactions. *Computers & structures*, 85(11):628–644, 2007.

[4] Frederic J Blom. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Computer methods in applied mechanics and engineering*, 167(3):369–391, 1998.

[5] AEJ Bogaers, S Kok, and T Franz. Strongly coupled partitioned fsi using proper orthogonal decomposition. In *Eight South African Conference on Computational and Applied Mechanics, Johannesburg, South Africa*, September 2012.

[6] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965.

[7] Paola Causin, Jean-Frédéric Gerbeau, and Fabio Nobile. Added-mass effect in the design of partitioned algorithms for fluid–

Table 4: Comparison of the number of iterations and relative computational time for pressure propagation through flexible tube where $\rho_s = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$, for different convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 10^{-4}, \epsilon = 10^{-7}$				$\Delta t = 10^{-4}, \epsilon = 10^{-9}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	5.38	4	11	1.00	6.58	5	14	1.00
Aitken	71.54	49	90	(13/100)	227	227	227	(2/100)
Broyden	–	–	–	(1/100)	–	–	–	(1/100)
IBQN-LS(0)	17.63	11	22	(16/100)	25.60	12	29	(26/100)
IBQN-LS(1)	16.50	3	19	3.09	22.51	12	25	3.46
IBQN-LS(5)	10.66	3	16	2.01	14.35	9	25	2.22
IBQN-LS(10)	8.46	3	16	1.58	10.66	8	25	1.66
IBQN-LS(20)	6.37	3	16	1.20	7.90	5	25	1.23
IBQN-LS(30)	5.64	3	16	1.08	7.98	5	25	1.27
IBQN-LS(40)	5.83	3	16	1.13	16.56	6	49	(48/50)

structure problems. *Computer methods in applied mechanics and engineering*, 194(42):4506–4527, 2005.

[8] A De Boer, MS Van der Schoot, and H Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11):784–795, 2007.

[9] Joris Degroote, Klaus-Jürgen Bathe, and Jan Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. *Computers & Structures*, 87(11):793–801, 2009.

[10] Joris Degroote, Robby Haelterman, Sebastiaan Annerel, Peter Bruggeman, and Jan Vierendeels. Performance of partitioned procedures in fluid–structure interaction. *Computers & structures*, 88(7):446–457, 2010.

[11] Joris Degroote, Abigail Swillens, Peter Bruggeman, Robby Haelterman, Patrick Segers, and Jan Vierendeels. Simulation of fluid–structure interaction with the interface artificial compressibility method. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3-4):276–289, 2010.

[12] Joris Degroote and Jan Vierendeels. Multi-level quasi-newton coupling algorithms for the partitioned simulation of fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 225:14–27, 2012.

[13] Wulf G Dettmer and Djordje Perić. On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction. *Computational Mechanics*, 43(1):81–90, 2008.

[14] Guido Dhondt. Calculix crunchix user's manual version 2.5. 2007.

[15] Charbel Farhat, Kristoffer G Van der Zee, and Philippe Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering*, 195(17):1973–2001, 2006.

[16] Miguel Ángel Fernández and Marwan Moubachir. A newton method using exact jacobians for solving fluid–structure coupling. *Computers & Structures*, 83(2):127–142, 2005.

[17] Christiane Förster, Wolfgang A Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer methods in applied mechanics and engineering*, 196(7):1278–1293, 2007.

[18] Jean-Frédéric Gerbeau, Marina Vidrascu, et al. A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(4):631–647, 2003.

[19] Robby Haelterman. *Analytical study of the least squares quasi-Newton method for interaction problems.* PhD thesis, Ghent University, 2009.

[20] Matthias Heil, Andrew L Hazel, and Jonathan Boyle. Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.

[21] Björn Hübner, Elmar Walhorn, and Dieter Dinkler. A monolithic approach to fluid–structure interaction using space–time finite elements. *Computer methods in applied mechanics and engineering*, 193(23):2087–2104, 2004.

[22] Esko Järvinen, Peter Råback, Mikko Lyly, and Juha-Pekka Salenius. A method for partitioned fluid–structure interaction computation of flow in arteries. *Medical engineering & physics*, 30(7):917–923, 2008.

[23] U Küttler, M Gee, Ch Foerster, A Comerford, and WA Wall. Coupling strategies for biomedical fluid–structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3-4):305–321, 2010.

[24] Hermann G Matthies and Jan Steindorf. Partitioned strong coupling algorithms for fluid–structure interaction. *Computers & Structures*, 81(8):805–812, 2003.

[25] Steven M Rifai, Zdeněk Johan, Wen-Ping Wang, Jean-Pierre Grisval, Thomas JR Hughes, and Robert M Ferencz. Multiphysics simulation of flow-induced vibrations and aeroelasticity on parallel computing platforms. *Computer methods in applied mechanics and engineering*, 174(3):393–417, 1999.

[26] Keith R Stein, Richard J Benney, Vinay Kalro, Andrew A Johnson, and Tayfun E Tezduyar. Parallel computation of parachute fluid-structure interactions. *UMSI research report/University of Minnesota (Minneapolis, Mn). Supercomputer institute*, 97:54, 1997.

[27] R Suliman. Development of parallel strongly coupled hybrid fluid-structure interaction technology involving thin geometrically non-linear structures. Master's thesis, MEng Thesis, University of Pretoria, 2011.

[28] Tayfun E Tezduyar and Sunil Sathe. Modelling of fluid–structure interactions with the space–time finite elements: solution techniques. *International Journal for Numerical Methods in Fluids*, 54(6-8):855–900, 2007.

[29] Ryo Torii, Marie Oshima, Toshio Kobayashi, Kiyoshi Takagi, and Tayfun E Tezduyar. Computer modeling of cardiovascular fluid–structure interactions with the deforming-spatial-domain/stabilized space–time formulation. *Computer Methods in Applied Mechanics and Engineering*, 195(13):1885–1895, 2006.

[30] Jan Vierendeels, Lieve Lanoye, Joris Degroote, and Pascal Verdonck. Implicit coupling of partitioned fluid–structure interaction problems with reduced order models. *Computers & structures*, 85(11):970–976, 2007.

[31] Wolfgang A Wall. *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen.* PhD thesis, Institut für Baustatik, Univer-

sität Stuttgart, 1999.

[32] BJBM Wolters, MCM Rutten, GWH Schurink, Ursula Kose, J De Hart, and FN Van De Vosse. A patient-specific computational model of fluid–structure interaction in abdominal aortic aneurysms. *Medical engineering & physics*, 27(10):871–883, 2005.

[33] SZ Zhao, XY Xu, AD Hughes, SA Thom, AV Stanton, B Ariff, and Q Long. Blood flow and vessel mechanics in a physiologically realistic model of a human carotid arterial bifurcation. *Journal of biomechanics*, 33(8):975–984, 2000.