An evaluation of quasi-Newton methods for application to FSI problems involving free surface flow and solid body contact

AEJ Bogaers[a]*, S Kok[b], BD Reddy[c,d], T Franz[e,f]

[a]Advanced Mathematical Modelling, Modelling and Digital Sciences, CSIR, South Africa

[b]Department of Mechanical and Aeronautical Engineering, University of Pretoria, South Africa

[c]Department of Mathematics and Applied Mathematics, University of Cape Town, South Africa

[d]Centre for Research in Computational and Applied Mechanics, University of Cape Town, South Africa

[e]Division of Biomedical Engineering, Department of Human Biology, University of Cape Town, South Africa

[f]Research Office, University of Cape Town, South Africa

Abstract

The design of coupling algorithms for partitioned fluid-structure interaction (FSI) simulations are typically validated on FSI problems involving large deformations of thin elastic structures with large added mass ratios. A large number of FSI problems may however feature additional internal non-linearities, examples of which include problems with free surface flow or FSI problems involving contact between two or more solid bodies. In this paper we aim to demonstrate the applicability of quasi-Newton methods when applied to these classes of problems. The analyses will focus on a comparison between two promising families of quasi-Newton methods, namely the 'quasi-Newton least squares' (QN-LS) family of methods and the 'multi-vector iteratively updated quasi-Newton' (MVQN) method. Both of these families of quasi-Newton methods construct approximations of the FSI system Jacobians using only iteratively obtained interface information, and can therefore be applied to black-box subdomain solvers. We will further attempt to quantify the ability of these QN methods to adequately approximate these additional non-linearities based on the form of the chosen interface equations.

1 Introduction

Fluid-structure interactions (FSI) is of importance in a number of engineering and life science applications, ranging from flutter prediction in aeroelasticity [23, 37], wind excited vibration of buildings and bridges [48] and biomedical applications of blood flow through the vascular system [39, 47, 50]. In general, there are two main approaches for the design of FSI modelling software, namely the monolithic approach [8, 30, 29] or the partitioned approach [17, 32, 35, 43]. The monolithic approach is to solve the fluid, solid and interface equations simultaneously in a unified solver, where the partitioned approach uses separate field solvers, which are solved in an iterative or successive fashion.

There are a wide array of arguments in favour of each of the two primary approaches. It is however commonly accepted that monolithic schemes are generally more robust and often more efficient than

[*] Corresponding author: abogaers@csir.co.za

partitioned schemes [16, 32]. In fact, the types of problems analysed in this paper, and many far more complex, are readily solvable by most state of the art commercial multi-physics packages, including for example Adina [3] (based on many of the FSI procedures outlined in [4, 5, 6, 7]). In general however, monolithic solution schemes preclude the coupling of two or more existing field solvers. Research pertaining to partitioned solution schemes therefore remains an important avenue of study, critical in cases where the user either has a direct need or particular preference in using a specific field solver. The remainder of this paper will be limited to the discussion of partitioned FSI schemes.

Strongly coupled FSI, for example problems involving high added mass ratios [12], require (semi-) implicit coupling of the two sub-domains (i.e. multiple iterations within a given time step). In the presence of strong non-linearities, simple backward and forward transfer of information is often insufficient to guarantee convergence. To improve the stability of partitioned FSI, several methodologies have been proposed ranging from fixed point iteration with dynamic relaxation [45], Robin interface conditions [24] and computing or approximating the system interface Jacobians [16, 20, 21, 25, 32, 35].

In this regard, Quasi-Newton (QN) methods have been demonstrated to be particularly useful in accelerating partitioned coupling schemes. Currently, two promising family of QN schemes are the 'quasi-Newton least squares method' (QN-LS) [43] and the 'multi-vector iteratively updated quasi-Newton' (MVQN) method [9]. Both of these QN methods allow for black-box sub-domain solvers, and construct approximations of the system Jacobians using iteratively obtained information from the sub-domain field solvers. While these two families of methods are closely related, they differ in how they incorporate information from multiple time steps. The QN-LS family of methods are well established and have been investigated in numerous publications [16, 17]. The QN-LS methods have been demonstrated to be capable of providing favourable performance properties, but are sensitive to a problem specific parameter of how many time steps to retain, with no *a priori* means to determine the optimal choice for this parameter. The MVQN method, by using a least change iterative updating scheme, removes the need to choose any parameters, and has furthermore been demonstrated to be capable of providing superior convergence behaviour [9, 38, 34].

In general, there are two alternative sets of interface equations to which the QN methods can be applied. The first relates to approximating the Jacobian of a residual equation relating the successive differences between one of the primary interface variables, (i.e. the differences between interface displacements or interface pressures). An alternative form would be to construct two block-Newton (BN) equations relating to the iterative changes in both interface pressures and displacements, and in turn construct approximations of the sensitivities of interface pressures with respect to displacements and *vice versa*. Both sets of interface equations offer their own merits. In general, the BN form of the interface Jacobians provide slightly improved coupling performance, on average requiring fewer coupling iterations (see for example [17]). The BN Jacobians further provide an approximation of the sensitivities between interface forces and displacements, which can be reused. See for example [10], where BN approximations were used to accelerate and stabilise an artificial compressibility based FSI solver to deal with the 'incompressibility dilemma'. The incompressibility dilemma refers to the mathematically ill-posed nature of FSI problems involving fully enclosed, incompressible fluid domains, when solved using partitioned Dirichlet-Neumann interface conditions. An example often provided in literature is that of a balloon inflation problem (see for example [33]). In such problems, the stationary Dirichlet fluid boundary condition and the incompressible nature of the fluid means flow cannot be forced into the fully enclosed domain.

The limitation of the BN system is that it requires solving two linear systems within a given coupling iteration. By using the interface residual equations, one can directly construct an approximation of the inverse of the Jacobian, thereby negating the need to solve any additional linear systems. The size of these linear systems are equal to the number of interface degrees of freedom (DOF) which are typically much smaller than the total combined DOFs of both the fluid and solid domains. Unfortunately, these interface Jacobians are often fully-populated (non-sparse), and the resulting linear systems can be computationally expensive to solve. When treating very large interfaces the computational cost of these linear systems may even exceed the combined computational time of both the fluid and solid sub-domains.

For affine problems, with exact arithmetic, Haelterman *et al.* [28] proved that the two forms (BN and interface residual) are algebraically identical. Therefore, for typical applications there would be little justification in using the block-Newton approach, which would provide the same result but increased computational effort. Unfortunately, because of computational limitations, numerical methods are limited to finite precision where FSI problems are furthermore highly non-linear. In this paper we therefore aim to compare the approximations provided by the MVQN and QN-LS methods to the two forms of interface equations. In order to better judge the comparative merits, the numerical experiments are chosen to include additional internal non-linearities. These problems include a steady state problem, a free-surface flow problem with an advancing wave front and a solid-body contact problem.

2 Partitioned FSI

Fluid-structure interactions can be defined as a two-field coupled problem, involving a fluid domain Ω_f and a solid domain Ω_s sharing a common interface Γ_{FSI}. In a partitioned setting, the fluid and solid subdomain solvers can be viewed as interface operators. The structural domain solver, $\boldsymbol{S}$, can be viewed as an interface operator which maps a given interface force (integrated pressure and shear stress fields), $\boldsymbol{f}$, to provide an interface displacement, $\boldsymbol{d}$, such that

$$\boldsymbol{d} = \boldsymbol{S}\left(\boldsymbol{f}\right), \tag{1}$$

where the fluid field solver can be described as

$$\boldsymbol{f} = \boldsymbol{F}\left(\boldsymbol{d}\right). \tag{2}$$

The fluid and solid interface mapping operators $\boldsymbol{S}$ and $\boldsymbol{F}$ represent the spatially discretised fluid and solid equations which are typically solved via numerical methods such as the finite element or finite volume methods. While the mapping operators represent the full field equations, the transfer quantities $\boldsymbol{f}$ and $\boldsymbol{d}$ typically only involve the forces and displacements along the shared interface Γ_{FSI}.

In order to satisfy the dynamic and kinematic conservation, the interface fields are integrated and transferred such that the interface stress states and displacement/velocity are conserved,

$$\boldsymbol{t}_s = \boldsymbol{t}_f, \quad \frac{\partial \boldsymbol{d}_s}{\partial t} = \boldsymbol{u}_f \quad \text{along}\, \Gamma_{\text{FSI}}, \tag{3}$$

where $\boldsymbol{u}_f$ denotes the fluid velocity along the interface and $\boldsymbol{t}_f = p_f \boldsymbol{n}_f - \boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f$ and $\boldsymbol{t}_s = \boldsymbol{\sigma}_s \cdot \boldsymbol{n}_s$. p_f denotes the fluid pressure along the interface, $\boldsymbol{\sigma}_f$ the fluid viscous stress tensor and $\boldsymbol{\sigma}_s$ the solid stress tensor. $\boldsymbol{n}_s$ and $\boldsymbol{n}_f$ denote the respective outward pointing normals along Γ_s and Γ_f if the solid and fluid interfaces are non-matching, $\Gamma_s \neq \Gamma_f$.

In this paper, the fluid operator $\boldsymbol{F}$ is solved using OpenFOAM where Calculix is used for the structural analysis. The interface load and motion transfer is performed using radial basis function (RBF) interpolation, using a consistent formulation. RBF interpolation requires no grid connectivity information and therefore is an elegant means to transfer information along non-matching interfaces, between a linear finite volume mesh to a higher-order finite element mesh. Interface tractions are transferred using a consistent formulation, thereby guaranteeing that a constant stress state (such as encountered in a constant stress patch test) can be exactly transferred regardless of mesh mismatch along the interface [11, 13, 14]. While a mismatched interface does imply that $\boldsymbol{t}_s \neq \boldsymbol{t}_f$, the errors introduced by the consistent RBF interface transfer falls below the errors already present due to the finite volume based fluid field solver, and disappears in the limit of mesh refinement [11].

For implicit-partitioned solvers, the interface operators in (1) and (2) are typically solved in an iterative fashion, with the interface quantities transferred backwards and forwards until some or other convergence tolerance is satisfied. Depending on the strength of coupling or non-linearity of the problem in question, simple backwards and forwards transfer of information is often insufficient. While there are several coupling acceleration schemes that have been proposed, in this paper the discussion is limited to the MVQN and QN-LS family of quasi-Newton methods.

2.1 Block-Newton Equations

The block-Newton equations are obtained by rewriting the interface operators in (1) and (2) as a root-finding problem, such that

$$\boldsymbol{r}_F = \boldsymbol{F}(\boldsymbol{d}) - \boldsymbol{f} = \boldsymbol{0}, \tag{4}$$

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

The coupled system defined by equations (4) and (5) can then be solved by computing the system Jacobian and solving for an update in the Newton direction

$$\begin{bmatrix} \dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\dfrac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I} & \dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} \\ \dfrac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} & \dfrac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} - \boldsymbol{I} \end{bmatrix} \left\{ \begin{array}{c} \Delta \boldsymbol{f} \\ \Delta \boldsymbol{d} \end{array} \right\} = \begin{bmatrix} -\boldsymbol{F}(\boldsymbol{d}) + \boldsymbol{f} \\ -\boldsymbol{S}(\boldsymbol{f}) + \boldsymbol{d} \end{bmatrix} \tag{6}$$

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

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k + \Delta \boldsymbol{f} \tag{7}$$

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

Because the fluid and solid partitioned solvers are executed in a staggered fashion, the system of linear equations are solved in a block-Newton fashion. Following a call to the fluid solver (that returns an interface force $\boldsymbol{f}$), the interface forces are updated:

$$\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I}\right)\Delta \boldsymbol{f} = -\left(\boldsymbol{F}(\boldsymbol{d}) - \boldsymbol{f}\right) - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\Delta \boldsymbol{d}. \tag{9}$$

Similarly, following a call to the solid solver (that returns a displacement $\boldsymbol{d}$), the displacement is updated

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

The aim of the quasi-Newton methods are then to construct approximations to the system Jacobians $\boldsymbol{J}_S = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}}$ and $\boldsymbol{J}_F = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}}$.

2.2 Interface Residual Equations

The block-Newton equations to be solved using (9) and (10) require constructing approximations for both $\boldsymbol{J}_S$ and $\boldsymbol{J}_F$ as well as solving two linear systems. The linear systems to be solved are of size $n_\Gamma \times n_\Gamma$, where n_Γ is the number of DOFs along the FSI interface and typically form a small fraction of the total DOFs of the complete system. These linear systems are however non-sparse and therefore not well suited to iterative solution procedures. For very large systems, the additional costs of these linear systems may in fact exceed the cost of the sub-domain field solvers. By constructing interface equations based on successive differences in one of the primary variables requires that no such linear systems need to be solved, and has been demonstrated to provide comparable coupling performance [17].

A root-finding equation using successive displacements (or equally interface forces) can be written as

$$\boldsymbol{R}_k = \boldsymbol{S}(\boldsymbol{F}(\boldsymbol{d}_k)) - \boldsymbol{d}_k, \tag{11}$$

where $\boldsymbol{S}\left(\boldsymbol{F}\left(\boldsymbol{d}_k\right)\right)$ is meant to indicate the sequential calling of $\boldsymbol{F}$ and $\boldsymbol{S}$. A Newton system based on the residual equation then becomes

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}} \Delta \boldsymbol{d} = -\boldsymbol{R}. \tag{12}$$

The benefit of (12) is that the inverse of $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}$ can be approximated directly such that no linear system solution steps are necessary, i.e.

$$\Delta \boldsymbol{d} = -\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} \boldsymbol{R}. \tag{13}$$

3 Quasi-Newton Methods

3.1 QN-LS

The Quasi-Newton Least Squares (QN-LS) method was first proposed in [43] and has subsequently been widely adopted. There are several associated methods within the family of QN-LS methods, with the primary differences meant to indicate which form of the interface Jacobian is approximated. The BQN-LS (block quasi-Newton least squares) indicates approximations applied to the block-Newton equations (9) and (10), the QN-LS refers to an approximation of the Jacobian in (12) and QN-ILS (quasi-Newton inverse least squares) to the inverse Jacobian in (13) [17].

3.1.1 QN-ILS

The QN-ILS has been the most widely adopted of the three variations. Aside from being somewhat simpler to implement, the QN-ILS form of the QN-LS method does offer the advantage that no linear system solutions are necessary, and only requires matrix-vector multiplications. To outline the QN-ILS algorithm, consider that k FSI iterations in time step $n+1$ have been performed. Therefore k observations are available in the current time step, which can be used to construct two observation matrices such that

$$\boldsymbol{V}^k = \left[\Delta \boldsymbol{R}^{k-1} \Delta \boldsymbol{R}^{k-2} \cdots \Delta \boldsymbol{R}^{1} \Delta \boldsymbol{R}^{0}\right], \tag{14}$$

$$\boldsymbol{W}^k = \left[\Delta \tilde{\boldsymbol{d}}^{k-1} \Delta \tilde{\boldsymbol{d}}^{k-2} \cdots \Delta \tilde{\boldsymbol{d}}^{1} \Delta \tilde{\boldsymbol{d}}^{0}\right], \tag{15}$$

where the observation vectors in $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ are constructed as

$$\Delta \boldsymbol{R}^k = \boldsymbol{R}^k - \boldsymbol{R}^{k-1}, \tag{16}$$

$$\Delta \tilde{\boldsymbol{d}}^k = \tilde{\boldsymbol{d}}^k - \tilde{\boldsymbol{d}}^{k-1}. \tag{17}$$

The approximate Jacobian is then computed as

$$\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} = \boldsymbol{W}\left(\boldsymbol{V}^T \boldsymbol{V}\right)^{-1} \boldsymbol{V}^T - \boldsymbol{I}, \tag{18}$$

with a quasi-Newton displacement update

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k - \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} \boldsymbol{R}^k. \tag{19}$$

Notationally it is important to note that $\tilde{\boldsymbol{d}}$ in (17) refers to the results obtained from the solid solver, $\tilde{\boldsymbol{d}}^k = \boldsymbol{S}\left(\boldsymbol{F}\left(\boldsymbol{d}^k\right)\right)$, where $\boldsymbol{d}^k$ refers to the displacement approximation from the QN update in (19).

The QN-ILS approximation requires that at least two iterations have been performed. To avoid large initial divergence, relaxation is performed in the first iteration of the first time step, in the form

$$\boldsymbol{d}^1 = (1-\omega)\,\boldsymbol{d}^0 + \omega\tilde{\boldsymbol{d}}^0, \tag{20}$$

where ω indicates the chosen relaxation factor.

Because information from previous time steps are often relevant to the current estimate, the performance of the QN-ILS can be significantly improved by including information from previous time steps [17]. To include information from multiple time steps, the QN-LS family of methods append observation matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ from q preceding time steps such that

$$\boldsymbol{V}^k = \left[\begin{array}{cccc} \boldsymbol{V}^n & \boldsymbol{V}^{n-1} & \cdots & \boldsymbol{V}^{n-q} \end{array}\right] \tag{21}$$

$$\boldsymbol{V}^k = \left[\begin{array}{cccc} \boldsymbol{W}^n & \boldsymbol{W}^{n-1} & \cdots & \boldsymbol{W}^{n-q} \end{array}\right]. \tag{22}$$

The optimal choice of q is unfortunately problem dependent, with no *a priori* way of determining an appropriate choice at run time.

One of the limitations of solving for an approximation in the form of (18) is that if one or more observation vectors are (nearly) linearly dependent then $\left(\boldsymbol{V}^T\boldsymbol{V}\right)^{-1}$ would be ill-conditioned. Degroote *et al.* [16] proposed that the least squares problem be solved using economy size QR decomposition of $\boldsymbol{V}^k$ such that

$$\boldsymbol{V}^k = \boldsymbol{Q}^k U^k \tag{23}$$

where $\boldsymbol{Q}^k$ represents the orthogonal and U^k the upper triangular matrices. The least squares coefficients $\boldsymbol{\alpha}^k$ can then be obtained solving the triangular system

$$U^k\boldsymbol{\alpha}^k = \boldsymbol{Q}^k\Delta\boldsymbol{R}. \tag{24}$$

The displacement update can now be solved in a matrix free fashion via the matrix-vector product

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \boldsymbol{W}^k\boldsymbol{\alpha}^k + \boldsymbol{R}^k. \tag{25}$$

In addition to providing a matrix free-solution, QR decomposition provides a useful way to indicate (near) linear dependence. If two vectors in $\boldsymbol{V}^k$ are (nearly) linearly dependent, the corresponding diagonal entry in U^k would be very small. This allows for the construction of a cutoff criterion where the i-th column of matrices $\boldsymbol{V}^k$ and $\boldsymbol{W}^k$ are omitted if

$$|U_{ii}| < \varepsilon, \tag{26}$$

where ε is the threshold criteria and is typically chosen to be very small. An alternative cutoff criterion, also used in this paper is

$$|U_{ii}| \leq \varepsilon \left|\left|U^k\right|\right|_2, \tag{27}$$

where $||\cdot||_2$ represent the Frobenius norm of the matrix.

It is important that information be appended to $\boldsymbol{W}^k$ and $\boldsymbol{V}^k$ such that it corresponds to a QR filtering scheme where older information is omitted first, and thereby preferentially retaining newer information.

3.1.2 BQN-LS

The approximations for the BN set of equations (9) and (10) can be obtained by constructing observation matrices relating differences in interface forces and displacements

$$\Delta \boldsymbol{D}_F = \left[\begin{array}{cccc} \Delta \boldsymbol{d}_F^k & \Delta \boldsymbol{d}_F^{k-1} & \cdots & \Delta \boldsymbol{d}_F^0 \end{array}\right] \tag{28}$$

$$\Delta \boldsymbol{F}_F = \left[\begin{array}{cccc} \Delta \boldsymbol{f}_F^k & \Delta \boldsymbol{f}_F^{k-1} & \cdots & \Delta \boldsymbol{f}_F^0 \end{array}\right], \tag{29}$$

where the approximate Jacobians are then constructed as

$$\boldsymbol{J}_F = \Delta \boldsymbol{F}_F \left(\Delta \boldsymbol{D}_F^T \Delta \boldsymbol{D}_F\right)^{-1} \Delta \boldsymbol{D}_F^T, \tag{30}$$

where subscripts F indicate that the observation matrices are constructed using displacement information provided to the fluid solver, and forces obtained from the fluid solver, i.e. $\boldsymbol{f}_F = \boldsymbol{F}\left(\boldsymbol{d}_F\right)$. Similarly, an approximation for $\boldsymbol{J}_S$ can be constructed to enable solving force and displacement updates using (9) and (10).

As with the QN-ILS, it is possible to introduce QR-filtering when computing the displacement and force updates using $\boldsymbol{J}_F$ and $\boldsymbol{J}_S$.

3.2 MVQN

The multi-vector quasi-Newton method was first proposed in [9]. While there are many similarities between the MVQN and QN-LS method, the MVQN method makes use of an iterative, least-change updating procedure. It therefore enables retaining information from previous time steps without the need to choose up front, how many time steps' information should be retained and has further been demonstrated to be capable of providing superior convergence performance [9, 38, 34] .

To remain consistent with the naming convention of the QN-LS family of methods, we will refer to the MVQN method applied to the block-Newton set of equations as MV-BQN and MV-IQN to indicate an approximation of the inverse interface Jacobian.

3.2.1 MV-IQN

As with the QN-ILS method, the MV-IQN method approximates the inverse Jacobian based on a series of observation vectors. Once again, assuming k FSI iterations have been performed two observation matrices can be constructed in the form

$$\boldsymbol{V}^k = \left[\Delta \boldsymbol{R}^k \Delta \boldsymbol{R}^{k-1} \cdots \Delta \boldsymbol{R}^1 \Delta \boldsymbol{R}^0\right], \tag{31}$$

$$\boldsymbol{W}_{\mathrm{MV}}^k = \left[\Delta \boldsymbol{d}^k \Delta \boldsymbol{d}^{k-1} \cdots \Delta \boldsymbol{d}^1 \Delta \boldsymbol{d}^0\right]. \tag{32}$$

It is important to note that there is a slight difference in the observation matrix $\boldsymbol{W}_{\mathrm{MV}}^k$ when compared to the QN-ILS method. For the MV-IQN method, $\boldsymbol{W}_{\mathrm{MV}}$ is constructed using $\Delta \boldsymbol{d}$, where QN-ILS requires $\Delta \tilde{\boldsymbol{d}}$. To distinctly clarify the notational difference we highlight here again that the residual for iteration k is given by

$$\boldsymbol{R}^k = \boldsymbol{S}\left(\boldsymbol{F}\left(\boldsymbol{d}^k\right)\right) - \boldsymbol{d}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k. \tag{33}$$

The primary difference between the MV-IQN method and QN-ILS method stems from how information from previous time steps are reused. Rather than appending information from previous time steps to $\boldsymbol{V}$ and $\boldsymbol{W}$, the MV-IQN method relies on an iterative updating procedure. By simply appending information to $\boldsymbol{V}$ and $\boldsymbol{W}$ the possibility of obtaining contradictory information and linearly dependent vectors increases. Furthermore, information from far back may no longer be relevant to the current time step, but should such information be retained by appending it to $\boldsymbol{V}$ and $\boldsymbol{W}$, it is given equal importance as information closer to the current time step solution. While many of these potential issues are somewhat mitigated by using a suitable filtering scheme (such as QR-filtering), the performance of the QN-LS family of methods are subject to the choice in number of retained histories q, and to a lesser extent the choice in cutoff criterion ε.

The MVQN family of methods make use of an iteratively updated Jacobian in an attempt to remove the need to choose any problem specific parameters. Given an approximate inverse Jacobian from the previous time step, $\boldsymbol{J}^n = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1}$, the update for the current time step is computed using the following update formula

$$\boldsymbol{J}^{n+1} = \boldsymbol{J}^n + \left(\boldsymbol{W}_{\mathrm{MV}} - \boldsymbol{J}^n\boldsymbol{V}\right)\left(\boldsymbol{V}^T\boldsymbol{V}\right)^{-1}\boldsymbol{V}^T. \tag{34}$$

The update formula (34) is equivalent to finding the updated Jacobian which minimises $\left|\left|\boldsymbol{J}^{n+1} - \boldsymbol{J}^n\right|\right|$ subject to the constraint of the interface secant equation. The Jacobian update is performed for each iteration, where $\boldsymbol{J}^n$ remains unchanged for the current time step, and $\boldsymbol{V}$ and $\boldsymbol{W}_{\mathrm{MV}}$ contains observation vectors from the current time step only. For a more detailed discussion on the MVQN method, and how it relates to other quasi-Newton methods we refer the reader to the work presented in [9].

As with the QN-LS family of methods, at least two iterations need to have been performed to construct the approximate inverse Jacobians. Therefore, for the first iteration in time step 1, relaxation is advised to avoid initial divergence. Furthermore, an initial approximation for $\boldsymbol{J}^n$ is required. In order to guarantee a non-singular starting matrix $\boldsymbol{J}^0 = -\boldsymbol{I}$. By setting $\boldsymbol{J}^0 = -\boldsymbol{I}$ results in the MV-IQN and QN-ILS to be identically equivalent for all iterations within the first time step, where the differences between the two methods are only present from time step 2 onwards.

One of the limitations of the MV-IQN method when compared the QN-ILS method is that the full Jacobian matrix $\boldsymbol{J}^n$ needs to be stored. This is in comparison to only having to store the currently retained observation vectors when using the QN-ILS method. Therefore, while the MV-IQN method also only requires matrix-vector products to solve for a displacement update, it may require a lot more memory when solving for systems with very large interface DOFs.

3.2.2 MV-BQN

The MV-BQN method uses the same observation matrices as used for the BQN-LS method with the primary difference being $\boldsymbol{J}_S^{n+1}$ and $\boldsymbol{J}_F^{n+1}$ are based on the update formula as described by (34). As with the MV-IQN method, initial approximations for the solid and fluid Jacobians are needed. For the initial Jacobian updates to be non-singular the Jacobians are initialised with zero matrices, $\boldsymbol{J}_S^0 = [\boldsymbol{0}]$ and $\boldsymbol{J}_F^0 = [\boldsymbol{0}]$. Doing so, again results in the MV-BQN and BQN-LS methods to be identically equivalent for all iterations within the first time step. The two families differ only on the basis of how information from multiple time steps are retained.

4 Test Problems

The test problems in the sections to follow have been performed using OpenFOAM [1] for the fluid domain and Calculix [22] for the structural domain. In the problems to follow we assume that the largest cost per iteration is accounted for by the fluid and structural domain solvers. The performance of the coupling schemes are therefore evaluated based on how many coupling iterations are required to reach the specified convergence. The convergence criteria across all problems are set as

$$\boldsymbol{r} = \frac{\left|\left|\boldsymbol{d}_\Gamma^k - \boldsymbol{d}_\Gamma^{k-1}\right|\right|}{\sqrt{n_\Gamma}} \leq 10^{-7}, \tag{35}$$

where n_Γ is the number of interface DOFs. It is perhaps important to note that the results cited in the sections to follow are not influenced by the chosen coupling schemes (within convergence tolerance).

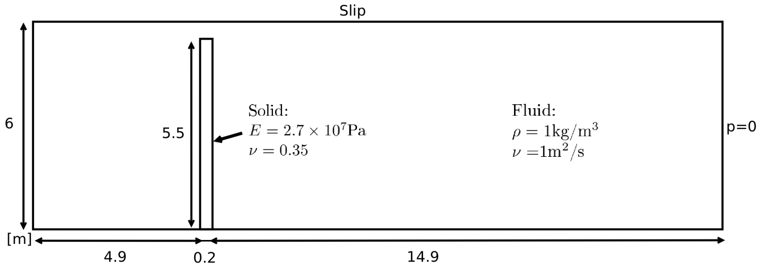


Figure 1: Steady state domain description.

The coupling schemes only affect the number of coupling iterations required to reach convergence, if convergence is obtained at all. The solution accuracy of the FSI problems are governed by the numerical schemes used within each of the field solvers, the spatial and temporal discretisation and interface information transfer.

4.1 Steady State Flow Past Vertical Beam

The first problem analysed here is a steady state problem of flow past a vertical beam, with the problem layout described in Figure 1. Steady state FSI problems are, generally speaking, far less prevalent than transient problems. Researchers and designers are usually more interested in the dynamic interaction of flow induced structural deformations or *vice versa*. Steady state problems are however of potential interest as a benchmarking tool (see for example [40]), and have found application in topology and shape optimisation (see for example [49] and the citations therein). While most FSI publications focus on the complexities of strongly coupled problems, specifically with high added mass instabilities, steady state problems offer their own set of challenges. While steady state problems are not density driven, they lack time steps which typically impose physical limits on the extent of possible structural deformations (from one time step to the next). It is of course possible to solve steady state problems using several steps (or pseudo-iterations), perhaps while ramping up the inlet velocity to simplify the numerical coupling. In this test problem we solve the steady state FSI problem using a steady state fluid solver, coupled to a steady state solid solver. The steady state solution is therefore equivalent to solving a transient problem with only 1 time step.

Because we solve the steady state problem without any time steps or pseudo-time steps, the MVQN and QN-LS family of methods are identical (since the primary difference between the two QN families is based on how information from multiple time steps are incorporated). The steady state problem therefore affords an opportunity to compare the difference between the BN and interface residual forms of the QN approximations. Based on the proof presented by Haelterman *et al.* [28], given exact arithmetic, and small structural deformations, the two QN approximations should yield identical convergence behaviour.

To test this hypothesis, two inlet flow velocities are used, namely $U_{\text{in}} = 0.1$m/s and $U_{\text{in}} = 1.0$m/s. The fluid velocity profile along with the beam deformation is shown in Figure 2, with the beam deflections plotted in Figure 3. For an additional comparison, we include the tip deflections for transient simulations, with a solid beam density of $\rho_s = 1000$kg/m^3 and time step sizes of $\Delta t = 0.1$s. As expected, given sufficient time, the transient tip deflection converges to the steady state results.

The steady state convergence rates for the two inlet velocities are shown in Figure 4. The results for the small, linear beam deformation confirms the findings of Haelterman *et al*, where the BQN and IQN convergence rates are nearly identical. It equally however highlights the potential differences between the two interface formulations when applied to non-linear problems with finite precision. The source of the differences is difficult to ascribe, but in general, for this particular problem, the IQN formulation appears to be marginally better behaved, where the residual decreases monotonically.

Finally in Table 1, for completeness, we include the mean number of coupling iterations required

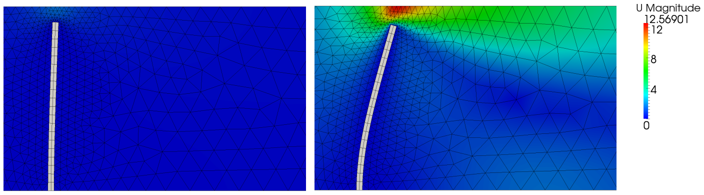


Figure 2: Steady state simulation results of a flexible membrane for inlet velocities of $U_{\mathrm{in}} = 0.1\mathrm{m/s}$ (left) and $U_{\mathrm{in}} = 1.0\mathrm{m/s}$ (right).

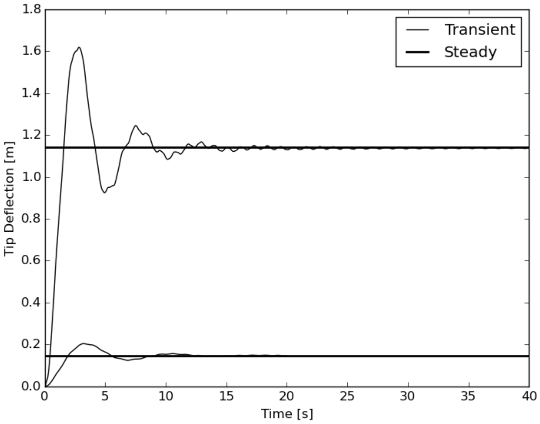
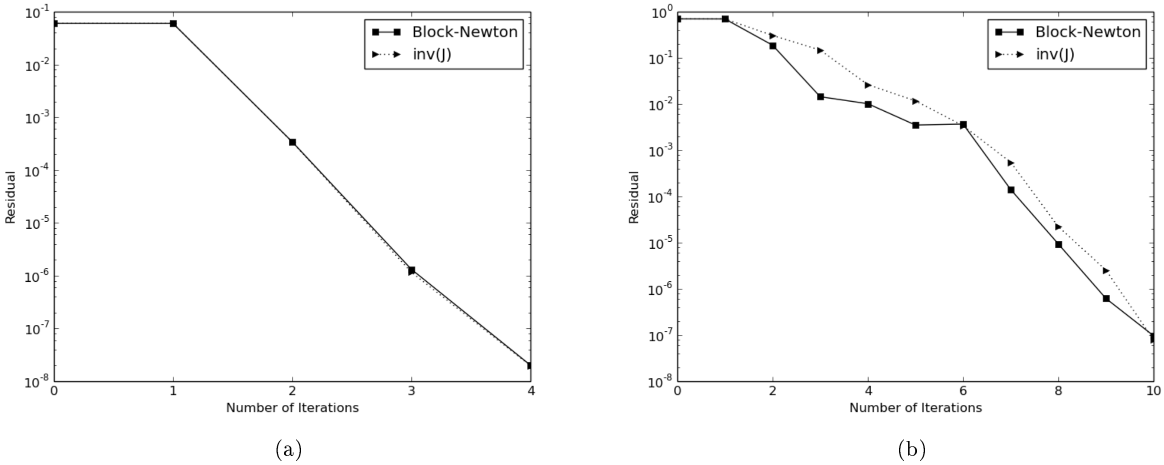


Figure 3: Beam tip deflections for inlet flow velocities of $U_{\mathrm{in}} = 0.1\mathrm{m/s}$ and $U_{\mathrm{in}} = 1\mathrm{m/s}$ past the vertical beam for both steady and transient simulations.



(a)

(b)

Figure 4: Convergence rates for the steady state simulation for inlet velocities of (a) $U_{\mathrm{in}} = 0.1\mathrm{m/s}$ and (b) $U_{\mathrm{in}} = 1.0\mathrm{m/s}$.

Table 1: Mean number of coupling iterations required for the transient vertical beam test problem with $U_{\text{in}} = 1\text{m/s}$ and time step size of $\Delta t = 0.1\text{s}$. QN-LS(q) is meant to indicate that q time step histories are retained, and the QR filtering is performed using a cutoff criterion of $|U_{ii}| \leq 10^{-8}$.

	No filtering		QR filtering	
	BN	inv($\boldsymbol{J}$)	BN	inv($\boldsymbol{J}$)
MVQN	3.15	3.39	–	–
QN-LS(0)	3.54	3.57	3.56	3.57
QN-LS(1)	3.12	3.42	3.12	3.34
QN-LS(2)	3.08	3.29	3.08	3.28
QN-LS(3)	3.11	3.28	3.10	3.23
QN-LS(5)	3.10	3.28	3.10	3.23
QN-LS(10)	3.22	4.08	3.19	3.28

for the transient simulation with an inlet velocity of $U_{\text{in}} = 1\text{m/s}$ and $\Delta t = 0.1\text{s}$. With a solid to fluid density ratio of 1000, this problem would typically be classified as weakly coupled, and therefore relatively simple to solve. The performance of QN-LS(q), with QR filtering, improves as q is increased until some threshold value of q, after which the mean number of coupling iterations tend to remain fairly constant. This observation of the QN-LS family of methods hold true for many FSI problems, including strongly coupled problems (see for example [19], where BQN-LS is applied to blood flow problems for multiple choices of q). This problem further highlights that should q be appropriately chosen, the QN-LS family of methods can outperform the MVQN family of methods.

4.2 Dam Break with Elastic Obstruction

FSI with free surfaces, involving totally or partially submerged bodies, are of interest in a number of fields, including civil and offshore engineering. Multiphase flow requires accurately capturing the interface between two or more fluids, and can be relatively unsteady due to large density differences within the flow domain. Due to these, potentially large, density differences the FSI interface can now be subjected to abruptly changing interface loading conditions. FSI with free surfaces have been the focus of several studies, and have been solved both using partitioned methods and monolithic schemes [18, 31, 36, 44, 46].

To investigate the performance of the QN methods, we analyse a collapsing column of water, striking an elastic wall, previously analysed in [31, 44]. The problem setup is shown in Figure 5, and consists of a 29.2cm column of water which collapses under gravity, striking an 8cm tall, 1.2cm wide, elastic obstacle. The tank is open at the top, and surface tension effects are ignored due to the large length scales.

The dam break problem (and other similar free surface flow problems) conceptually poses difficulties for the QN methods. Prior to the striking of the advancing front, the coupling algorithms spend several time steps training the interface Jacobians given a very low loading condition, followed by a very sharp and abrupt increase in interface forces. This additional source of non-linearities is especially interesting considering the difference between the BN and inverse Jacobian approximations. The inverse Jacobian, based only on sequential displacements, has no formal way of accounting for large changes in fluid forces. The BN approximation on the other hand constructs an approximation accounting for the relationship between interface forces and displacements, and therefore conceptually, should be better equipped to deal with these additional non-linearities.

In Figure 6, the advancing front along with the elastic deformation is shown for various time steps. The problem is solved here using both a coarse and fine mesh, with the tip deflection shown in Figure 7. The coarse mesh consists of 3672 linear FVM elements and 14 quadratic, full integration solid elements, with time step sizes of $\Delta t = 0.001\text{s}$, where the fine mesh consists of 30340 linear fluid elements and 112 quadratic, full integration solid elements, using a time step size of $\Delta t = 0.0005\text{s}$. The smaller

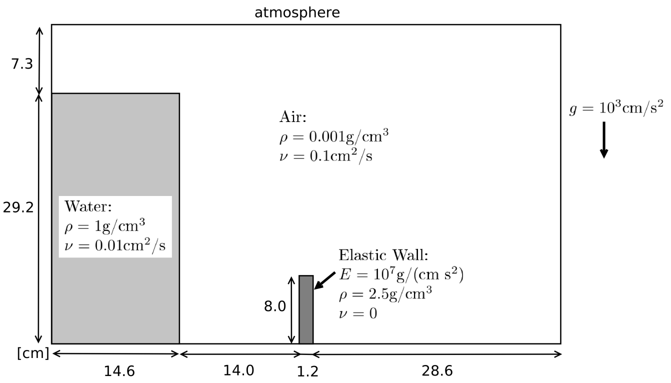
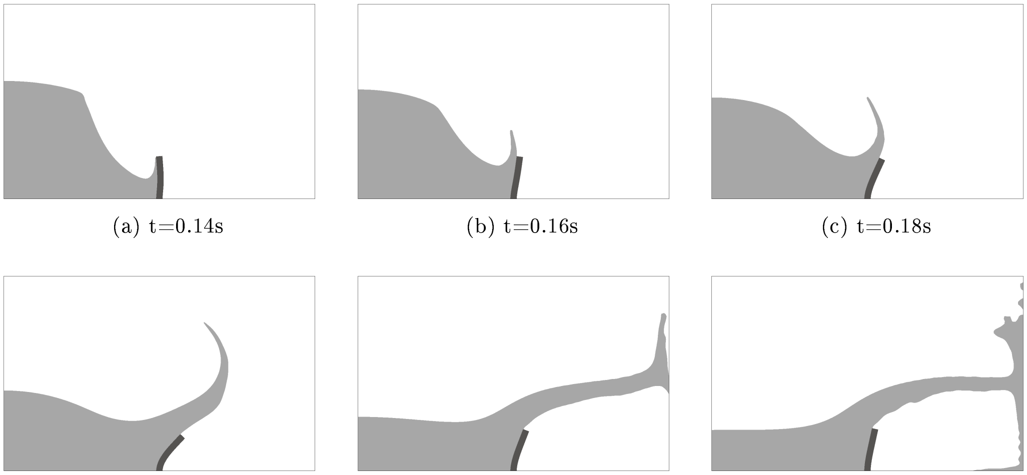

atmosphere
7.3
29.2
Air:
$\rho = 0.001\text{g/cm}^3$
$\nu = 0.1\text{cm}^2/\text{s}$
$g = 10^3\text{cm/s}^2$
Water:
$\rho = 1\text{g/cm}^3$
$\nu = 0.01\text{cm}^2/\text{s}$
Elastic Wall:
$E = 10^7\text{g/(cm s}^2)$
$\rho = 2.5\text{g/cm}^3$
$\nu = 0$
8.0
[cm]
14.6
14.0
1.2
28.6

Figure 5: Dam break with elastic structure problem description.


(a) t=0.14s

(b) t=0.16s

(c) t=0.18s

(d) t=0.24s

(e) t=0.34s

(f) t=0.42s

Figure 6: Wave interaction with elastic obstruction at various time steps.

time step size used for the finer discretisation is not to facilitate the FSI coupling, but rather for the stability of the free-surface flow fluid computations, which fails to solve for larger time step sizes.

In Table 2 the mean number of iterations required for a convergence tolerance of $\epsilon \leq 10^{-7}$ is outlined and the typical convergence behaviours, for time step 130 (directly following the initial wave strike on the elastic structure), is shown in Figure 8, along with the number of iterations for each of the time steps. As expected, the BN form of the Jacobian approximation provides an overall improved performance over the IQN Jacobians. The improvement is however slight, and considering that the IQN formulations have no linear systems to solve, remains an attractive option. The dam break results further highlight the sensitivity of the QN-LS family of methods to the choice of q. Should they be chosen appropriately, the QN-LS methods compare well to the MVQN approximations. However choosing them too large or too small, results in either divergence or a significantly deteriorated performance. The use of QR-filtering does however improve the stability and performance of the QN-LS(q) family of methods. In Table 3 the mean number of coupling iterations, for a full sweep of QR filtering parameters is shown. The overall performance of the QN-LS family of methods remain fairly constant amongst the various cutoff criteria.

In general, the QN methods analysed in this paper appear to be well suited to dealing with problems where there are abrupt changes in the magnitude of fluid forces.

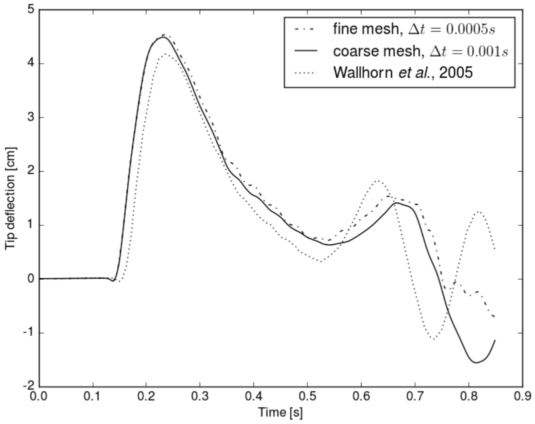
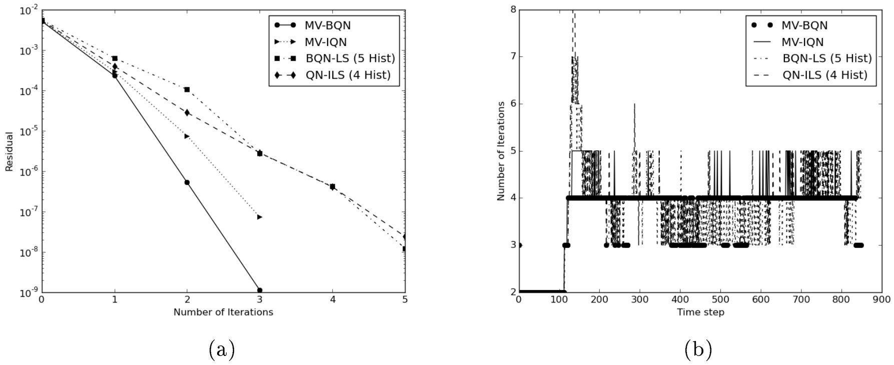


Figure 7: Plot of beam tip deflection for the dam break benchmark problem.

Table 2: Comparison of the mean number of iterations required for convergence for the dam break with elastic obstruction problem. QN-LS(q) indicates q time histories are retained and if a scheme is non-convergent the time step at which divergence occurred is indicated in brackets. QR filtering is applied here using a cutoff of criterion of $|U_{ii}| \leq 10^{-8}$. A comprehensive comparison of QN-LS family for different QR filtering parameters is given in Table 3.

	Coarse Mesh, $\Delta t = 0.001$s				Fine Mesh, $\Delta t = 0.0005$s			
	No filtering		QR filtering		No filtering		QR filtering	
	BN	inv($\boldsymbol{J}$)	BN	inv($\boldsymbol{J}$)	BN	inv($\boldsymbol{J}$)	BN	inv($\boldsymbol{J}$)
MVQN	3.60	3.83	–	–	3.40	3.83	–	–
QN-LS(0)	5.07	5.33	5.02	5.32	5.77	(343/1700)	(343/1700)	5.94
QN-LS(1)	4.28	4.24	4.23	4.20	4.93	(343/1700)	4.91	(311/1700)
QN-LS(2)	3.94	(132/850)	3.93	(581/850)	4.42	(343/1700)	4.47	4.08
QN-LS(3)	(130/850)	(129/850)	3.77	(581/850)	(1543/1700)	(343/1700)	(1530/1700)	3.89
QN-LS(4)	(130/850)	(129/850)	(584/850)	3.872	(250/1700)	(343/1700)	3.98	3.89
QN-LS(5)	(129/850)	(129/850)	3.65	3.905	(249/1700)	(343/1700)	(343/1700)	3.83



(a)

(b)

Figure 8: Performance plots for the dam break benchmark problem for the coarse mesh, illustrating (a) the typical convergence behaviour, shown here for time step 130, directly following the initial contact of the wave with the elastic structure and (b) a summary of the number of iterations per time step.

Table 3: Comparison of mean number of iterations required by BQN-LS and QN-ILS for different number of histories with different QR filtering parameters for the dam break with elastic obstruction benchmark problem using the coarse mesh. If a scheme is non-convergent, the time step at which divergence occurs is shown in brackets.

		No	QR filtering: $\lvert U_{ii}\rvert < \epsilon$			QR filtering: $\lvert U_{ii}\rvert \leq \epsilon \left\|U^k\right\|_2$		
	# Hist	filtering	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-10}$
BQN-LS	0	5.07	15.42	5.06	5.02	5.00	5.04	5.04
	1	4.28	7.67	4.25	4.23	4.27	4.28	4.26
	2	3.94	7.21	3.95	3.93	3.96	3.72	(583/850)
	3	(130/850)	(560/850)	3.82	3.77	3.85	3.77	3.76
	4	(130/850)	5.69	3.72	(584/850)	3.77	3.71	3.79
	5	(129/850)	5.64	3.69	3.65	3.74	3.71	(136/850)
QN-ILS	0	5.33	7.44	5.51	5.32	5.35	5.34	5.34
	1	4.24	7.91	4.30	4.20	4.23	4.18	4.19
	2	(132/850)	5.14	4.00	(581/850)	3.95	(584/850)	3.92
	3	(129/850)	4.99	3.93	(581/850)	3.88	3.86	(148/850)
	4	(129/850)	(284/850)	(583/850)	3.872	(558/850)	4.41	(134/850)
	5	(129/850)	4.94	(582/850)	3.905	(581/850)	(581/850)	(136/850)

4.3 Flexible Beam with Solid Body Contact

This section aims to provide a preliminary analysis of the QN methods applied to FSI problems with solid body contact. Examples of FSI with solid body contact include opening and closing heart valves [2, 41], interaction between blood cells, or blood cells flowing through an occlusion or capillary. Most mature structural analysis packages, including for example Abaqus, and in our case Calculix, support solid body contact. Solid body contact is commonly done through the inclusion of springs, to impose forces to resist movement or penetration between two surfaces [22, 42]. The contact force magnitude is then based on the penetration distance and a chosen spring stiffness function.

Solid body contact via the insertion of spring forces pose a serious challenge to FSI solvers. Preventing large inter-body penetration may involve arbitrarily large contact forces, and these occur sporadically only when contact is active. As opposed to free surface flow, these additional, potentially non-linear, forces come from the structural solver rather than the fluid solver. As such, approximation methods using the BN-QN formulation, do not have access to these additional forces when constructing an approximation of the fluid force-solid displacement relationship.

While Calculix natively supports contact, we implement a simple contact algorithm externally to the solid solver. This is done in order to have access to the contact forces, which in turn can be made available when constructing the BN-QN approximations. While implementing contact externally to the solid solver is no longer strictly speaking a 'black-box' operation, the next section will attempt to show the effect on the BN-QN performance, when including and excluding the contact forces. Implementing contact externally to the solid solver is not strictly necessary if the user's specific choice of solid solver allows for contact stresses (or forces) to be extracted from the results.

It should be mentioned that flow solvers based on an arbitrary Lagrangian-Eularian (ALE) description of the fluid domain (as used in this study) are not ideally suited to contact problems. The event of contact implies that the fluid region between two bodies in contact should disappear, which cannot be done naturally using solvers such as OpenFOAM. In order to fully close off a domain requires topological changes in terms of both localised re-meshing (when the mesh quality deteriorates) as well as redefining boundary definitions. This is both time consuming, problem specific, and may introduce errors when mapping solutions between different domain or topological definitions. For these classes of problems, methods such as the immersed boundary method [27], fixed grid methods [26] or fictitious domain methods [15, 41] are better suited. Since the primary interest in our analysis is the behaviour of the QN methods, we avoid this issue by forcing a small gap to remain present at all times.

While we retain the gap to avoid topological changes, the gap is symbolically closed via the fluid equations. The fluid velocity in the cells between the two surfaces in contact is forced to be equal to the adjoining interface velocities. This allows for the small gap to remain, while the effect on the flow profile will be resemblant of flow when the fluid cells are removed, barring the geometrical error introduced by the small gap. This geometrical error is directly proportional to the size of the enforced gap, and is a function of the solver Courant number requirements, the smaller the enforced gap, the smaller the maximum time step size. Interestingly, closing the gap via the equations does not induce the “incompressibility dilemma” observed for fully enclosed problems [33], which is likely due to the iterative (SIMPLE-like) velocity-pressure solution algorithm used in OpenFOAM. One solution to the incompressibility dilemma is to use artificial compressibility. Since the problem analysed in this section is intrinsically one which would ordinarily require such methods, we include in our analysis the performance results of using interface artificial compressibility (IAC) in conjunction with the MVQN method. The IAC+MVQN method is beyond the scope of this paper, but for more information, we refer the reader to [10].

4.3.1 Contact Formulation via Springs

In this paper a simple spring analogy contact formulation is implemented based on the penetration distance of slave surface coordinates into a master surface. For the purposes of this analysis, frictional forces are ignored and only normal contact forces are considered; this is illustrated in Figure 9. The contact algorithm implemented here can be summarised as follows:

1. Pair up slave nodes to master surface elements.
2. Compute the distance between slave nodes and master surface elements.
3. If slave node penetrates the master surface, connect a contact spring, with direction normal to master surface.

The spring force, when active, is given by

$$f_{\text{contact}} = (K d_p)\, \boldsymbol{n}_m, \tag{36}$$

where K is the spring stiffness coefficient, d_p the penetration distance and $\boldsymbol{n}_m$ the master surface normal vector. Choosing K as a large constant will result in hard contact, or conversely a small K will result in soft contact which leads to larger penetrations. It is important to note, the numerical example to follow is a 2D model. OpenFOAM however only supports 3-dimensional systems, where the 3-axis is ignored in the simulation. To remain fully consistent, the solid domain is also modelled in 3D, with a single element through the thickness and appropriate boundary conditions to model plane stress. Therefore, whenever contact is present, two springs are included as opposed to the single spring as suggested by Figure 9. As a result, whenever spring stiffness characteristics are provided, should replication of the results be attempted in true 2-dimensions, the spring stiffnesses should be doubled. This in no way alters the behaviour of the contact system, but is mentioned for completeness.

4.3.2 2D Valve Contact Problem

The contact problem analysed here is a simple valve like problem first analysed in [41]. The problem is representative of a mitral valve where the papillary muscle has been omitted, allowing for the valve to fully snap through, given the inlet velocity. The domain geometry is shown in Figure 10, where the contact plane is chosen in such a way that the valve is in contact from the start of the simulation, releasing from contact following the snap-through instance. The fluid and valve properties are chosen to be representative of biomechanic systems, with a density of $\rho_f = 1.0\text{g/cm}^3$, and viscosity of 0.03P. The valve is described by a geometrically non-linear FEM formulation with $E = 3 \times 10^6\text{dynes/cm}^2$, $\rho_s = 1.2\text{g/cm}^3$ and $\nu = 0.3$. The problem presents two complexities. Firstly, in the form of high added mass with large deformations, and secondly, solid to rigid body contact.

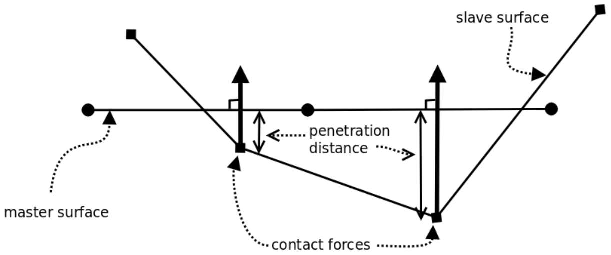


Figure 9: Simple depiction of slave nodes to master surface contact.

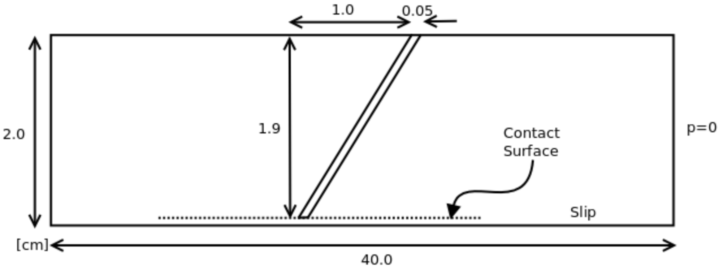


Figure 10: Valve-contact problem layout.

The fluid domain is discretised using 1861 linear fluid elements, and 40 linear FEM elements are employed for the structural domain. Linear elements are used for the solid domain due to the choice of using a slave *node*-master surface contact formulation. For higher order elements, a more sophisticated surface to surface contact scheme would be required. While linear solid elements are not ideal, they are sufficient for the purposes of illustrating the behaviour of the QN methods with solid body contact.

The inlet flow velocity is given by

$$U_{\mathrm{in}} = 20.0 \sin\left(2\pi t\right) \text{ cm/s}. \tag{37}$$

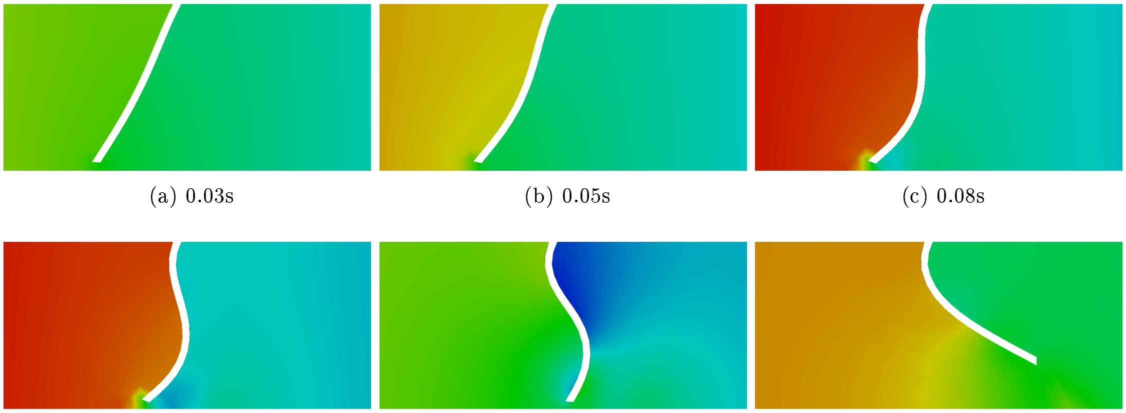
The pressure contours along with the valve displacement at various time steps are shown in Figure 11. Despite the small gap, a large pressure drop across the valve remains evident.

4.3.3 Effect on MV-BQN when contact forces are included or excluded

Prior to comparing the performance of the QN methods, applied to the different interface equations, we investigate the ability of the MV-BQN to cope with contact when the forces are included or excluded when constructing the approximate Jacobians. This is done by comparing the performance for both 'hard' and 'soft' contact. Soft contact is achieved by lowering the contact stiffness, which reduces the contact force gradients and thereby simplifying the FSI coupling, but in turn also increases the contact inter-body penetration. To illustrate the difference in penetration distance, the valve displacement is shown in Figure 12 for a spring stiffness of $K = 1 \times 10^4 \mathrm{g/cm}^2$ and $K = 2 \times 10^3 \mathrm{g/cm}^2$, with a plot of the valve tip displacement shown in Figure 13.

To include the contact forces, the fluid forces $\boldsymbol{f}$, used in the construction of the observation matrices $\boldsymbol{\Delta F}_S$ and $\boldsymbol{\Delta F}_F$ in (29) should be modified to $\{\boldsymbol{f}\} = \{\boldsymbol{f}_{\text{fluid}} + \boldsymbol{f}_{\text{contact}}\}$. The included contact forces are therefore treated as an additional fluid force. In the current implementation these contact forces are available because contact is implemented externally to the solid solver. An equivalent black-box implementation would be possible if the choice of solid solver allows the user to extract contact stresses or forces from the set of results.

The mean number of coupling iterations are outlined in Table 4. Using soft contact, the MV-BQN approximation is capable of producing convergent results. However for hard contact, the MV-BQN


(a) 0.03s

(b) 0.05s

(c) 0.08s

(d) 0.1s

(e) 0.12s

(f) 0.18s

Figure 11: Pressure and valve displacement plots at different time steps, for $\Delta t = 0.0001$s.

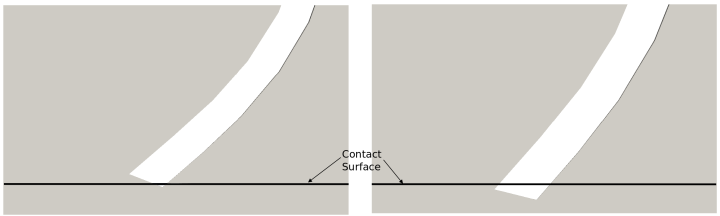


Figure 12: Penetration distance for choice of spring contact stiffness of $K = 1 \times 10^4 \text{g/cm}^2$ (left) and $K = 2 \times 10^3 \text{g/cm}^2$ (right).

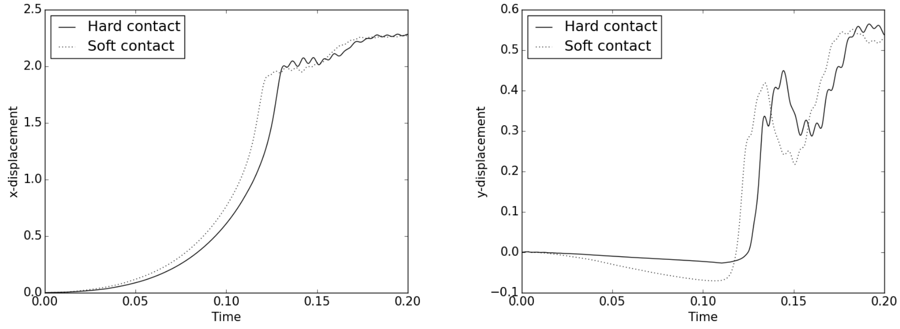
method is unable to converge unless the contact forces are made visible. Despite the choice of using a linear force-displacement relationship for the contact forces (which could equally be formulated to be a non-linear relationship) the effective contact forces have the potential of being highly non-linear. To illustrate this, the contact forces for both hard and soft contact are shown in Figure 14. The MV-BQN approximation, using hard contact, diverges at 0.0047s, which corresponds to the initial activation of the contact force, as seen in Figure 14(b).

4.3.4 Comparison of the BN-QN and IQN formulations

For the remainder of the analysis, only hard contact ($K = 1 \times 10^4 \text{g/cm}^2$) is used, where the contact forces are made visible to the BN-QN approximations. The mean number of iterations required for convergence is shown in Table 5 with the typical convergence behaviour shown in Figure 15. Since this is a problem which would typically require a solution method that can deal with fully-enclosed fluid

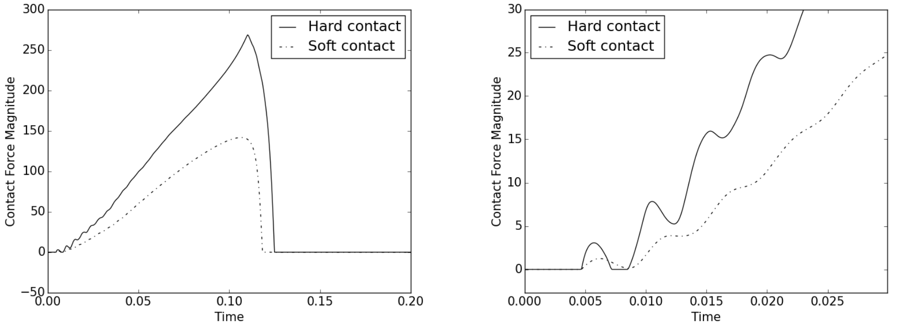
Table 4: A comparison of the mean number of iterations required for both soft and hard contact using the MVQN formulation. The comparison focuses on the performance of the BN approximation when the external contact forces are included or excluded in the QN approximation.

	contact force	soft contact	hard contact
	included	$(K = 2 \times 10^3)$	$(K = 1.0 \times 10^4)$
MV-BQN	yes	4.41	5.02
MV-BQN	no	4.65	non-convergence (47/2000)



(a) (b)

Figure 13: The (a) x and (y) y-displacement of the bottom right tip of the valve with contact benchmark problem.



(a) (b)

Figure 14: (a) A plot of the contact forces for both 'hard' and 'soft' contact with (b) a close up on the first 0.03 seconds of the simulation.

Table 5: Comparison of the mean number of iterations required for convergence the valve with solid-body contact problem. QN-LS(q) indicates q time histories are retained and if a scheme is non-convergent the time step at which divergence occurred is indicated in brackets. These results are for contact forces included within the block-Newton QN approximations. QR filtering performed here using a cutoff criteria of $|U_{ii}| \leq 10^{-8} \left|\left|U^k\right|\right|_2$ with a comparison for a sweep of QR filtering parameters provided in Table 6.

	No Filtering		QR Filtering	
Method	BN	inv $(\boldsymbol{J})$	BN	inv$(\boldsymbol{J})$
MVQN	5.02	4.92	–	–
IAC+MVQN	5.58	N/A	–	–
QN-LS(0)	(932/2000)	(1212/2000)	(914/2000)	(130/2000)
QN-LS(1)	(830/2000)	7.49	(905/2000)	7.51
QN-LS(2)	(623/2000)	6.36	(692/2000)	6.41
QN-LS(5)	(395/2000)	5.29	(11/2000)	5.33
QN-LS(10)	(404/2000)	(1104/2000)	(13/2000)	(1111/2000)

Table 6: Comparison of the mean number of iterations required by BQN-LS and QN-ILS for different number of time histories and QR filtering parameters for the valve with solid body contact. If a scheme is non-convergent the time step at which divergence occurs is shown in brackets.

		QR filtering: $|U_{ii}| < \epsilon$			QR filtering: $|U_{ii}| \leq \epsilon \left|\left|U^k\right|\right|_2$		
	# Histories	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-10}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-10}$	$\varepsilon = 10^{-12}$
BQN-LS	0	(1)	(670)	(1478)	(914)	10.40	10.40
	1	(1)	(877)	(892)	(905)	(892)	(892)
	2	(1)	(690)	(686)	(692)	(680)	(686)
	3	(1)	(603)	(579)	(578)	(579)	(579)
	5	(1)	(24)	(504)	(11)	(505)	(504)
	10	(1)	(14)	(424)	(13)	(403)	(429)
QN-ILS	0	(1)	(1236)	(1253)	(130)	(1253)	(1253)
	1	(1)	(1005)	7.50	7.51	7.50	7.50
	2	(1)	6.40	6.40	6.41	6.40	6.40
	3	(1)	5.86	5.84	5.86	5.84	5.84
	5	(1)	5.32	(1104)	5.33	(1104)	(1104)
	10	(1)	(1109)	(1104)	(1111)	(1105)	(1106)

domains, the results using interface artificial compressibility in conjunction with MVQN (IAC+MVQN) is included as an additional reference. It is important to note that the IAC+MVQN method does not have a residual equivalent, and therefore only the results for the block-Newton approximations are shown.

Overall the MVQN approximations outperform the QN-LS family of methods. With the exception of 0 retained histories with a QR filtering criteria of $|U_{ii}| \leq 10^{-10} \left|\left|U^k\right|\right|_2$, the BQN-LS fails to converge regardless of the number of retained histories, and while QN-ILS provides convergent results, the method remains sensitive to the choice of q.

5 Conclusion

In this paper, a numerical comparison of the QN-LS and MVQN methods was performed for FSI problems involving free surface flow, solid body contact and a steady state analysis. The analyses focused on a comparison of the convergence behaviour of the QN methods when used to approximate the inverse interface Jacobian and the block-Newton interface Jacobians. Despite the additional non-

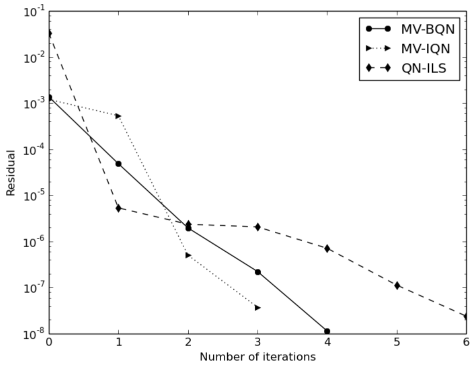


Figure 15: Typical convergence behaviour shown here for $t = 0.1$s for the solid body contact valve problem. The BQN-LS is not included as it failed to provide convergent results.

linearities present within the chosen problems, both sets of interface equations resulted in comparable performance, with predominantly super-linear convergence rates.

Because the block-Newton Jacobian approximations attempt to construct a relationship between interface displacements and forces, it is important that all the interface forces and displacements are properly accounted for. This naturally occurs for free surface flow, where the additional non-linear interface forces, resulting from the large fluid density differences are included within the fluid interface stresses. On the other hand, for solid body contact, these additional non-linear forces are a result of the solid domain, and are therefore not natively included. Excluding these contact forces when constructing the BN Jacobian approximations results in an unstable system. The inverse Jacobian approximation deals with these non-linearities naturally, without any special treatment of the information. Therefore, in many regards the inverse Jacobian can be considered an approximation of the total derivative of the interface equations. They do in some sense include the partial derivatives of the additional contact forces with respect to the change in displacement, where for the block-Newton systems these approximations have to be constructed independently.

Unless there is a specific requirement that the sensitivities of interface forces with respect to interface displacements (or *vice versa*) be made available, approximating the inverse interface Jacobian appears to be the preferred method. It provides comparable coupling performance and does not require solving any linear systems, which can result in significant computational savings for large problems.

References

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

[2] Matteo Astorino, Jean-Frédéric Gerbeau, Olivier Pantz, and Karim-Frédéric Traore. Fluid–structure interaction and multi-body contact: application to aortic valves. *Computer Methods in Applied Mechanics and Engineering*, 198(45):3603–3612, 2009.

[3] KJ Bathe. Adina theory and modeling guide, 2002.

[4] K.J. Bathe and H. Zhang. Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions. *Computers & Structures*, 56:193–213, 1995.

[5] K.J. Bathe and H. Zhang. Finite element developments for general fluid flows with structural interactions. *Int. Journal for Numerical Methods in Engineering*, 60:213–232, 2004.

[6] K.J. Bathe and H. Zhang. Mesh adaptivity procedure for cfd and fluid-structure interactions. *Computers & Structures*, 87:604–617, 2009.

[7] K.J. Bathe and H. Zhang. The solution of maxwell's equations in multiphysics. *Computers & Structures*, 132:99–112, 2014.

[8] 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.

[9] AEJ Bogaers, S Kok, BD Reddy, and T Franz. Quasi-Newton methods for implicit black-box FSI coupling. *Computer Methods in Applied Mechanics and Engineering*, 279:113–132, 2014.

[10] AEJ Bogaers, S Kok, BD Reddy, and T Franz. Extending the robustness and efficiency of artificial compressibility for partitioned fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 283:1278–1295, 2015.

[11] Alfred EJ Bogaers. *Efficient and robust partitioned solution schemes for fluid-structure interactions*. PhD thesis, University of Cape Town, 2015.

[12] Paola Causin, Jean-Frédéric Gerbeau, and Fabio Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer methods in applied mechanics and engineering*, 194(42):4506–4527, 2005.

[13] A De Boer, AH Van Zuijlen, and H Bijl. Review of coupling methods for non-matching meshes. *Computer methods in applied mechanics and engineering*, 196(8):1515–1525, 2007.

[14] Aukje de Boer, Alexander H van Zuijlen, and Hester Bijl. Comparison of conservative and consistent approaches for the coupling of non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 197(49):4284–4297, 2008.

[15] J De Hart, FPT Baaijens, GWM Peters, and PJG Schreurs. A computational fluid-structure interaction analysis of a fiber-reinforced stentless aortic valve. *Journal of biomechanics*, 36(5):699–712, 2003.

[16] 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.

[17] 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.

[18] Joris Degroote, Antonio Souto-Iglesias, Wim Van Paepegem, Sebastiaan Annerel, Peter Bruggeman, and Jan Vierendeels. Partitioned simulation of the interaction between an elastic structure and free surface flow. *Computer methods in applied mechanics and engineering*, 199(33):2085–2098, 2010.

[19] 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.

[20] Simone Deparis, Marco Discacciati, Gilles Fourestey, and Alfio Quarteroni. Fluid–structure algorithms based on steklov–poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5797–5812, 2006.

[21] 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.

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

[23] 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.

[24] Luca Gerardo-Giorda, Fabio Nobile, and Christian Vergara. Analysis and optimization of robin-robin partitioned procedures in fluid-structure interaction problems. *SIAM Journal on Numerical Analysis*, 48(6):2091–2116, 2010.

[25] 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.

[26] Axel Gerstenberger and Wolfgang A Wall. An extended finite element method/lagrange multiplier based approach for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1699–1714, 2008.

[27] Antonio J Gil, A Arranz Carreño, Javier Bonet, and O Hassan. The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics*, 229(22):8613–8641, 2010.

[28] R Haelterman, B Lauwens, H Bruyninckx, and J Petit. Equivalence of qn–ls and bqn–ls for affine problems. *Journal of Computational and Applied Mathematics*, 278:48–51, 2015.

[29] 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.

[30] 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.

[31] Christophe Kassiotis. *Nonlinear fluid-structure interaction: a partitioned approach and its application through component technology*. PhD thesis, Université Paris-Est, 2009.

[32] 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.

[33] Ulrich Küttler, Christiane Förster, and Wolfgang A Wall. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure dirichlet fluid domains. *Computational Mechanics*, 38(4-5):417–429, 2006.

[34] Florian Lindner, Miriam Mehl, Klaudius Scheufele, and Benjamin Uekermann. A comparison of various quasi-newton schemes for partitioned fluid-structure interaction. In *VI International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2015*, 2015.

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

[36] Eugenio Oñate and J Garcıa. A finite element method for fluid–structure interaction with surface waves using a finite calculus formulation. *Computer methods in applied mechanics and engineering*, 191(6):635–660, 2001.

[37] 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.

[38] Klaudius Scheufele. Robust quasi-newton methods for partitioned fluid-structure simulations. Master's thesis, MEng Thesis, University of Stuttgart, 2015.

[39] 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.

[40] Stefan Turek and Jaroslav Hron. *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow.* Springer, 2006.

[41] Raoul van Loon, Patrick D Anderson, and Frans N van de Vosse. A fluid–structure interaction method with solid-rigid contact for heart valve dynamics. *Journal of computational physics*, 217(2):806–823, 2006.

[42] ABAQUS Version. 6.7. users manual. *Inc. and Dassault systemes*, 2007.

[43] 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.

[44] Elmar Walhorn, Andreas Kölke, Björn Hübner, and Dienter Dinkler. Fluid–structure coupling within a monolithic model involving free surface flows. *Computers & structures*, 83(25):2100–2111, 2005.

[45] Wolfgang A Wall. *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen.* PhD thesis, Institut für Baustatik, Universität Stuttgart, 1999.

[46] Wolfgang A Wall, Steffen Genkinger, and Ekkehard Ramm. A strong coupling partitioned approach for fluid–structure interaction with free surfaces. *Computers & Fluids*, 36(1):169–183, 2007.

[47] 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.

[48] Jann N Yang, Anil K Agrawal, Bijan Samali, and Jong-Cheng Wu. Benchmark problem for response control of wind-excited tall buildings. *Journal of Engineering Mechanics*, 130(4):437–446, 2004.

[49] Gil Ho Yoon. Stress-based topology optimization method for steady-state fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 278:499–523, 2014.

[50] 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.