MULTI-SECANT QUASI-NEWTON VARIANTS FOR PARALLEL FLUID-STRUCTURE SIMULATIONS – AND OTHER MULTI-PHYSICS APPLICATIONS

KLAUDIUS SCHEUFELE* AND MIRIAM MEHL†

Abstract. Multi-secant quasi-Newton methods based on secant information produced with no overhead throughout subsequent solver iterations have been shown to be particularly suited to solve non-linear fixed-point equations that arise from partitioned multi-physics simulations where the exact Jacobian is inaccessible. In all these methods, the multi-secant equation for the approximate (inverse) Jacobian is enhanced by a norm minimization condition. It is well-known, that fluid-structure simulations, e. g., typically require the use of secant-information from previous time steps. The number of these time-steps highly depends on the application, its parameters, the used solvers, and the mesh resolution. Using to few leads to a relatively high number of iterations, using too many not only to a computational overhead but also to an increase of the number of iterations, as well. Determining the optimal number requires a costly try-and-error process, which can be avoided in a modified method (presented in [13]) that considers the difference of the current (inverse) Jacobian and the one of the previous time step in the norm-minimization. Thus, previous time step information is taken into account in an implicit and automatized way without magic parameters. We show numerical results for fluid-structure interactions for both methods proving the robustness and numerical efficiency of the second variant. In addition, we propose a new algorithm eliminating the drawback of having to store full interface Jacobian approximations for the Jacobian difference norm minimization. This results in a highly efficient, parallelizable, and robust iterative solver applicable for surface coupling in many types of multi-physics simulations.

Key words. partitioned multi-physics, non-linear fixed-point solver, quasi-Newton, fluid-structure interactions

AMS subject classifications. 65B99, 65H10, 65M99, 68W10, 74B99, 76D05

1. Introduction.

Solving non-linear systems of equations that are given only implicitly or even as a black-box functionality has become a common issue in multi-physics simulations, where interface equations depend on the discretization details of the single-physics domains, or in coupled optimization problems, where the task function is evaluated by several software components together. In literature, two basic solution approaches are discussed: (accelerated) fixed-point iterations (also referred to as Anderson mixing [1, 8, 22, 14, 18]) and quasi-Newton methods [15, 9, 6, 12, 21, 16, 20]. Both approaches have been developed independently from each other but can be shown to be equivalent in large parts. Basically, they are generalizations of GMRES for non-linear problems. The common ingredient of all methods is the approximation of the (inverse) Jacobian matrix based on 1) a multi-secant approach using input and output data of the involved modules over several solver iterations and 2) a norm minimization condition. In particular for time-dependent problems requiring the solution of a non-linear equation in each time step, using information from previous time steps can substantially improve the convergence. This can be done in two fundamentally different ways: The first possibility is to explicitly include input and output data from previous time steps in the multi-secant equation. This induces the unknown parameter *number of reused time steps* which is highly problem-dependent and can only be determined in a costly try-and-error process. The other

*Institute for Parallel and Distributed Systems, Universität Stuttgart, Universitätsstraße 38, Stuttgart, Germany (klaudius.scheufele@ipvs.uni-stuttgart.de)

†Institute for Parallel and Distributed Systems, Universität Stuttgart, Universitätsstraße 38, Stuttgart, Germany (miriam.mehl@ipvs.uni-stuttgart.de)

possibility is to minimize the norm of the difference of subsequent (inverse) Jacobian approximations. This implicit reuse of old information results in very robust and fast converging solvers, but requires the explicit calculation and storage of the Jacobian matrix, which is not necessary otherwise. The costs for this are prohibitively large in most applications. Therefore, we present novel ideas to reduce these costs to linear complexity in terms of the number of unknowns by sophisticated Jacobian truncation and restart methods. In Sect. 2, we give an overview of the methods for a general fixed-point problem and present convergence results for partitioned fluid-structure interactions in Sect. 3. The efficient linear complexity algorithms are introduced in Sect. 4 followed by numerical results evaluating the suitability of the different Jacobian truncation and restart variants in Sect. 5.

2. Quasi-Newton and Anderson Acceleration.

When solving problems as described above, the problem can often be reduced to a general fixed-point equation

$$H(x) = x \quad \text{with the residual} \quad R(x) := H(x) - x = 0, \tag{2.1}$$

where $H : \mathbb{R}^N \to \mathbb{R}^N$ is an operator that we are able to apply, i.e., we assume that we have a simulation environment that allows to evaluate $H(x)$ for a given input x. No further details of H are known. We additionally assume that the evaluation of H is very expensive such that minimizing the number of iterations is crucial. In the last years, two different communities have developed methods for this problem: so-called quasi-Newton methods have been used in particular in the context of fluid-structure interactions [15, 6, 12, 16, 13], whereas variants of Anderson mixing [1] have been developed in a more general context [14, 22, 8, 18]. We introduce the general idea from a quasi-Newton point of view. A Newton step for (2.1) reads

$$x^{k+1} = x^k - J_R^{-1} R(x^k) = x^k - (J_{\tilde{R}}^{-1} - I) R(x^k) \tag{2.2}$$

with $\tilde{R} := I - H^{-1} = R \circ H^{-1}$ and the reasoning that $H - R = I$ and, thus, $R^{-1} = (H - R) \circ R^{-1} = \tilde{R}^{-1} - I$. All methods are based on matrices storing the response in output differences depending on input differences of the function $\widetilde{R}(x) := x - H^{-1}(x)$:

$$W_k = \left[\Delta\tilde{x}_0^k, \Delta\tilde{x}_1^k, \cdots, \Delta\tilde{x}_{k-1}^k\right], \text{with} \quad \Delta\tilde{x}_i^k = \tilde{x}^k - \tilde{x}^i \,,$$
$$V_k = \left[\Delta R_0^k, \Delta R_1^k, \cdots, \Delta R_{k-1}^k\right], \text{with} \quad \Delta R_i^k = R(x^k) - R(x^i)$$

with $\tilde{x}^k := H(x^k)$. k is a number that can be equal to the number of iterations done so far, include a certain number of iterations from previous time steps in time-dependent applications, or it can be fixed to a given value. For an approximation J^{-1} of the inverse Jacobian of $\widetilde{R}$, the multi-secant equation

$$J^{-1} V_k = W_k \tag{2.3}$$

has to be fulfilled. This under-determined system for the N^2 entries of J^{-1} needs suitable further restrictions which leads to a choice of methods that we shortly outline:

Least Squares. The least squares approach typically chooses k as the number of previous iterations (possibly with an additional filter ensuring the full rank of V_k by deleting columns if necessary). It enhances (2.3) by the norm minimization condition

$$\|J^{-1}\| \to \min, \tag{2.4}$$

where $\|\cdot\|$ denotes the Frobenius norm. This leads to the approximation

$$J^{-1} = W_k V_k^\dagger \quad \text{with} \quad V_k^\dagger := (V_k^T V_k)^{-1} V_k^T \tag{2.5}$$

of the inverse Jacobian with the pseudo-inverse $V_k^\dagger$ of V_k [21, 17, 13].

Broyden. Broyden's methods [4] always uses $k = 1$, i.e., only information from a single iteration is used, which leads, together with the norm minimization

$$\|J^{-1,i+1} - J^{-1,i}\| \to \min \tag{2.6}$$

with the currently best inverse Jacobian approximation $J^{-1,i}$ to an iterative improvement of the inverse Jacobian approximation according to the formula

$$J^{-1,i+1} = J^{-1,i} + \frac{(W_1 - J^{-1,i} V_1) V_1^T}{V_1^T V_1}. \tag{2.7}$$

This method is known as *bad* Broyden's method in contrast to the original *good* Broyden's method where both, the secant equation and the norm minimization are formulated in terms of the Jacobian instead of its inverse.

Multi-Vector-Update. The multi-vector-update method can be interpreted as a generalization of Broyden's method that replaces the rank-one update in (2.6) based on the previous iterate $J^{-1,i}$ of the Jacobian by a rank-k update of an 'older' Jacobian approximation. This is realized by the norm minimization

$$\|J^{-1} - J_{prev}^{-1}\| \to \min \tag{2.8}$$

Often, this method is used for time dependent problems with a non-linear fixed-point equation in every time steps which typically yields similar Jacobians in subsequent time steps. In the latter case, J_{prev}^{-1} is the inverse Jacobian approximation from the previous time step. The inverse Jacobian is computed by a rank-k update of J_{prev}^{-1}:

$$J^{-1} = J_{prev}^{-1} + (W_k - J_{prev}^{-1} V_k) V_k^\dagger. \tag{2.9}$$

Successive Rank-One Jacobian Updates. Haeltermann showed in [11] that the approximate inverse Jacobian in (2.5) can be written as a sequence of rank-one updates

$$\widehat{J}^{-1,k+1} - \widehat{J}^{-1,k} = A_M V_{k+1} V_{k+1}^\dagger - A_M V_k V_k^\dagger = A_M \bar{L}_{k+1} \bar{L}_{k+1}^T \tag{2.10}$$

for a linear mapping A_M with $A_M V_{k+1} = W_{k+1}$ and $\bar{L}_{k+1}$ denoting the last column of L_{k+1}, the matrix containing the orthonormalization of the columns of V_k. Without knowing the actual mapping A_M, one can compute

$$\bar{L}_{k+1} = \frac{(I - L_k L_k^T)\Delta R_k^{k+1}}{\|(I - L_k L_k^T)\Delta R_k^{k+1}\|} \quad \text{and} \quad A_M \bar{L}_{k+1} = \frac{(J^{-1,k} - I) R(x^{k+1})}{\|(I - L_k L_k^T)\Delta R_k^{k+1}\|}.$$

where L_k is the matrix with the orthonormalized columns of V_k. Generalizing (2.10) to the multi-vector-update formula yields

$$J^{-1,k+1} - J^{-1,k} = (A_M - J_{prev}^{-1}) \bar{L}_{k+1} \bar{L}_{k+1}^T. \tag{2.11}$$

As an initial guess for J^{-1}, $J^{-1,0} = J_{prev}^{-1}$ is used. Haeltermann [11] proposes an improved least squares method with old time step recovery using the least squares method in rank-one update formulation (2.5). It also uses $J^{-1,0} = J_{prev}^{-1}$ and is similar, but not identical with (2.11) as it lacks the term $J_{prev}^{-1}\bar{L}_{k+1}\bar{L}_{k+1}^{T}$in the update.

Anderson Mixing. Comparing [22, 8] with [6, 16, 13] shows that the quasi-Newton step executed with the Jacobian computed from the least squares condition (2.5) is equivalent to type II Anderson acceleration, whereas type I Anderson acceleration is equivalent to a quasi-Newton method using an approximation of the Jacobian instead of its inverse. [17, 13] show that for fluid-structure interactions, our application example described in the following section, the inverse Jacobian approximation is stable, which is not the case for the Jacobian approximation for non-trivial problems [17].

In the following, we restrict our attention to the least squares approach, denoted as **LS**, and the multi-vector-update method, denoted as **MVJ**. Before discussing algorithmical details in Sect. 4, we recapitulate convergence results for two transient and strongly coupled fluid-structure interaction scenarios in the following section.

3. Application to Fluid-Structure Interactions.

Partitioned fluid-structure interaction simulations use two separate, possibly black-box solvers for the fluid and the structure domain, respectively. For our experiments, we use the incompressible flow solver and the elastic structure solver from the open source simulation toolbox OpenFOAM[1]. The coupling conditions are realized by a Dirichlet-Neumann approach: The fluid solver takes displacements and velocities (x_d) at the wet surface and computes forces (x_f) exerted on the structure, whereas the structure solver takes these forces and computes new displacements and velocities at the wet surface. Depending on whether we execute the two solvers simultaneously or one after the other, this yields two different types of fixed-point equations that have to be fulfilled in each (implicit) time step:

$$\begin{pmatrix} 0 & F \\ S & 0 \end{pmatrix} \begin{pmatrix} x_f \\ x_d \end{pmatrix} = \begin{pmatrix} x_f \\ x_s \end{pmatrix} \text{ (Vec-system)} \quad \text{or} \quad S \circ F(x_d) = x_d \text{ (Seq-system)},$$

where S denotes the action of the structure solver at the wet surface, F the action of the flow solver. We consider two test-cases displayed in Fig. 3.1, the FSI3 benchmark from [19] and a flow through a flexible tube.

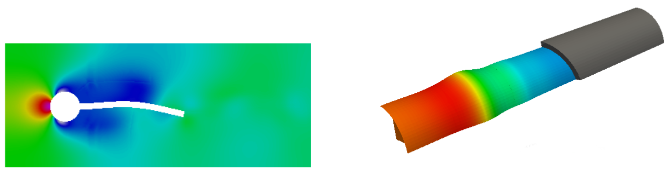

Fig. 3.1: Geometry and pressure visualization for the test-cases (a) FSI3 (at $t = 0.697\,s$) and (b) flexible tube (at $t = 3.0 \cdot 10^{-1}\,s$).

The FSI3 test-case models a flow around a cylinder with an attached flexible flap. The computational domain is $2.5\,\mathrm{m} \times 0.41\,\mathrm{m}$ in size and the incompressible flow is driven by a parabolic velocity profile with mean inflow velocity $\bar{v} = 0.2\,\mathrm{m\,s^{-1}}$ at the left boundary and free outflow at the right boundary. The flexible flap is modelled using a Saint-Venant-Kirchhoff material model. A period of $2\,s$ is with $dt = 1 \times 10^{-3}\,\mathrm{s}$. Both solvers use implicit Euler time steps. The flexible tube example [2, 6, 9] simulates a wave propagating in a three- dimensional elastic tube induced by an initial pressure

[1] http://www.openfoam.org/

pulse of peak 1333.2 Pa with a duration of $10^{-3}s$. The tube is 0.05 m long, has a wall thickness of 0.001 m, and an inner diameter of 0.01 m. It is considered an external elastic structure with neglected inertia. A hundred time steps with $dt = 1 \times 10^{-4}$ s are simulated. The physical parameters for both scenarios are given in Table 3.1.

FSI3 cylinder flap		**3d flexible tube**	
Fluid	Solid	Fluid	Solid
$\rho_f = 1 \times 10^3\,\mathrm{kg\,m^{-3}}$	$\rho_s = 1 \times 10^3\,\mathrm{kg\,m^{-3}}$	$\rho_f = 1 \times 10^3\,\mathrm{kg\,m^{-3}}$	$\rho_s = 1.2 \times 10^3\,\mathrm{kg\,m^{-3}}$
$\nu_f = 1.0 \times 10^{-3}\,\mathrm{Pa \cdot s}$	$E = 1.4 \times 10^6\,\mathrm{N\,m^{-2}}$	$\nu_f = 3.0 \times 10^{-3}\,\mathrm{Pa \cdot s}$	$E = 3.0 \times 10^5\,\mathrm{N\,m^{-2}}$
$Re = 200$	$\nu_s = 0.4$	$Re = 200$	$\nu_s = 0.3$

Table 3.1: Physical fluid and structure parameters for the two test-cases. ρ_f and ρ_s denote the fluid and structure density, ν_f the dynamic fluid viscosity, Re the Reynolds number, E the Young's modulus, ν_s the Poisson's ratio.

All experiments were conducted using preCICE[2] [10] for data transfer, data mapping, and the iterative solver. Table 3.2 shows the resulting numbers of iterations. They show that both solvers yield comparable convergence properties, but the MVJ solver does not require the costly optimization of the highly scenario-dependent number of reused time steps. For the flexible tube test-case, the MVJ method is superior even if we compare it with the LS method with the optimal number of re-used time steps. For more results, also on the parallel scalability of the solvers, see [17, 13, 5].

	3d flexible tube					**FSI3**						
reuse	0	4	8	12	16	0	2	4	6	8	16	32
Seq-LS	15.5	9.3	8.8	9.2	9.4	11.5	6.2	5.3	5.5	5.7	6.6	7.8
Seq-MVJ	8.5					5.5						
Vec-LS	28.9	14.6	13.4	13.2	13.3	20.0	8.0	6.2	5.6	5.3	6.2	11.9
Vec-MVJ	11.6					6.2						

Table 3.2: Averaged number of coupling iterations of the LS and MVJ method for the flexible tube and the FSI3 test-case for the Seq- and the Vec-system and varying numbers of reused time steps for the LS approach.

4. Efficient Algorithmic Realization.

We have shown the robustness and good convergence of the MVJ method for the fluid-structure interaction applications in Sect. 3. This, however, has to be seen alongside with the higher computational costs of the MVJ method compared to the LS approach. To illustrate this, we shortly summarize the algorithmic components of both methods. Further details and a description of the parallelized versions can be found in [5]. The quasi-Newton iteration for the least-squares method and the multi-vector-update, respectively, reads

$$x^{k+1} = H(x^k) - W_k V_k^{\dagger} r^k, \tag{4.1}$$

$$x^{k+1} = H(x^k) - (J_{prev}^{-1} + \widetilde{W}_k V_k^{\dagger}) r^k \quad \text{with} \quad \widetilde{W}_k := W_k - J_{prev}^{-1} V_k. \tag{4.2}$$

A common ingredient of both methods is the pseudo-inverse $V_k^{\dagger} = (V_k^T V_k)^{-1} V_k^T$. Finding $(V_k^T V_k)^{-1} V_k^T y$ for a given vector y is equivalent to solving the least-squares minimization $z = \mathrm{argmin}_{\bar{z} \in \mathbb{R}^N} \|V_k \bar{z} - y\|_2$. This can be done very efficiently using a **QR-decomposition** of V_k that exploits the fact that V_k only grows by one column in

[2]http://www5.in.tum.de/wiki/index.php/PreCICE Webpage

each iteration. From the decomposition $V_k = QR$, we get z from solving the quadratic system $\widehat{R}z = \widehat{Q}^T y$ via **backward substitution**. $\widehat{R} \in \mathbb{R}^{k\times k}$ denotes the first k rows of R and $\widehat{Q}$ contains the first k columns of Q (if $V_k \in \mathbb{R}^{N\times k}$). The costs for the QR-factorization including backward substitution are $O(N \times k^2) + O(k^3)$ (compare [5]). For both methods, we calculate $z = \alpha = (V_k^T V_k)^{-1} V_k^T(-r^k)$ using the right-hand side $y := r^k$. With this, we get the next quasi-Newton iterate from

$$x^{k+1} = \tilde{x}^k + W_k\,\alpha \quad \text{and} \quad x^{k+1} = \tilde{x}^k - J_{prev}^{-1} r^k + (W_k - J_{prev}^{-1} V_k)\,\alpha \tag{4.3}$$

for least-squares and the multi-vector-update, respectively. The multi-vector-update in addition requires the calculation of $\widetilde{W}_k$ and, at the end of a time step, the computation and storage of the resulting Jacobian to be used as J_{prev}^{-1} in the next time step. As $J^{-1} \in \mathbb{R}^{N\times N}$, we have to carefully design the algorithms for these two components in order to avoid any $O(N^2)$ costs in terms of storage and in terms of operations.

4.1. Efficient Representation of the Inverse Jacobian J^{-1}. J^{-1} has a rank that is substantially smaller than N. This can be exploited to reduce the storage requirements and the computational costs of the MVJ method. We avoid to build and store the entire matrix but rather re-compute and store the sub-matrices $\widetilde{W}_k \in \mathbb{R}^{N\times k}$ and $V_k^\dagger \in \mathbb{R}^{k\times N}$ over several time steps. If k_m is the number of iterations in time step m, the inverse Jacobian approximation after M time steps can be written as

$$J^{-1} = \widetilde{W}_{k_1}^T V_{k_1}^\dagger + \widetilde{W}_{k_2}^T V_{k_2}^\dagger + \ldots + \widetilde{W}_{k_M}^T V_{k_M}^\dagger. \tag{4.4}$$

Obviously, this approach is efficient only for M substantially smaller than N. Therefore, we investigate different restart alternatives. At each restart, the inverse Jacobian estimation is restarted with new data. In other words, we divide the overall simulation in chunks of time steps that represent one era of the Jacobian estimation. Whether or not information from previous chunks can be retained in newer chunks depends on the restart policy. In the following, M denotes an upper bound for the number of time steps per chunk and K an upper bound for the number of iterations per time step. We consider three different restart approaches:

RS-0. Clear all. This obviously results in $O(N \times K \times M)$ costs for both the storage of J^{-1} in the form (4.4) and for the M pairs of matrix-vector multiplications

$$y := V_{k_m}^\dagger x \;\text{ and }\; \widetilde{W}_{k_m} y \;\text{ in }\; J^{-1}x \;\text{ for any vector }\; x \in \mathbb{R}^N.$$

RS-LS. Clear J^{-1}, but keep columns in V and W from time steps within the current chunk, i.e., use the initial guess $J^{-1} := 0$, $\widetilde{W}_0 := W$, and $V_0^\dagger := (V^T V)^{-1} V^T$. If we reuse at most $\bar{K}$ columns, the total costs are $O(N \times \bar{K}) + O(N \times K \times M)$.

RS-SVD. Do a subspace tracking based on a singular value decomposition (SVD) of the matrix J^{-1}

$$J^{-1} = \Psi\Sigma\Phi^T \;\text{ with }\; \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N)$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_N \geq 0$ are real singular values, and $\Psi \in \mathbb{R}^{N\times N}$ and $\Phi \in \mathbb{R}^{N\times N}$ are orthogonal matrices. At restart, we truncate this decomposition by cutting off all singular values below a given threshold, i.e., we restart with

$$J^{-1} = \underbrace{(\Psi_{\cdot,j})_{j=1,\ldots,\bar{K}}}_{=:\,\overline{\Psi}} \underbrace{\begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_{\bar{K}} \end{pmatrix}}_{=:\,\overline{\Sigma}} \underbrace{(\Phi_{\cdot,j})_{j=1,\ldots,\bar{K}}^T}_{=:\,\overline{\Phi}^T}. \tag{4.5}$$

The costs strongly depend on the efficient realization of the underlying SVD decomposition. Apart from this step, the total costs are $O(N \times \bar{K}) + O(N \times K \times M)$ if we truncate the SVD decomposition such that only $\bar{K}$ values are left. For the efficient implementation of the SVD, we assume that we have a truncated singular value decomposition as in (4.5). At the end of the next chunk, our new estimate reads

$$\overline{\Psi}\,\overline{\Sigma}\,\overline{\Phi}^T + \sum_{m=1}^{M} \widetilde{W}_{k_m} V_{k_m}^{\dagger} \tag{4.6}$$

for which we have to compute an updated truncated SVD by performing M low-rank updates of the form

$$\overline{\Psi}\,\overline{\Sigma}\,\overline{\Phi}^T + AB^T = \left[\begin{array}{cc} \overline{\Psi} & A \end{array}\right] \left[\begin{array}{cc} \overline{\Sigma} & 0 \\ 0 & I \end{array}\right] \left[\begin{array}{cc} \overline{\Phi} & B \end{array}\right]^T \tag{4.7}$$

with $A, B \in \mathbb{R}^{N \times k_m}$. We use the algorithm proposed in [3], i.e., we compute the orthogonal components of A and B. With the matrices P and Q defining an orthonormal basis of the column space of $(I - \overline{\Psi}\,\overline{\Psi}^T)A$ and $(I - \overline{\Phi}\,\overline{\Phi}^T)B$, we define

$$R_A := P^T(I - \overline{\Psi}\,\overline{\Psi}^T)A, \quad \text{and} \quad R_B := Q^T(I - \overline{\Phi}\,\overline{\Phi}^T)B.$$

With this, the inverse Jacobian update (4.7) can be transformed to

$$\overline{\Psi}\,\overline{\Sigma}\,\overline{\Phi}^T + AB^T = \left[\overline{\Psi}\ P\right]\ K\ \left[\overline{\Phi}\ Q\right]^T \ \text{with}\ K = \left[\begin{array}{cc} \overline{\Sigma} & 0 \\ 0 & 0 \end{array}\right] + \underbrace{\left[\begin{array}{c} \overline{\Psi}^T A \\ R_A \end{array}\right]}_{:=\mathcal{A}} \underbrace{\left[\begin{array}{c} \overline{\Phi}^T B \\ R_B \end{array}\right]}_{:=\mathcal{B}}^T$$

Diagonalizing K as $\Psi'^T K \Phi' = \Sigma'$ finally yields

$$\overline{\Psi}\,\overline{\Sigma}\,\overline{\Phi}^T + AB^T = \left(\left[\overline{\Psi}\ P\right]\Psi'\right)\ \Sigma'\ \left(\left[\overline{\Phi}\ Q\right]\Phi'\right)^T. \tag{4.8}$$

If c is the dimension of the column space of $(I - \overline{\Psi}\,\overline{\Psi}^T)A$ and $(I - \overline{\Phi}\,\overline{\Phi}^T)B$, the costs for computing an orthonormal basis of these spaces are $O(c^2 N)$. The matrix $K \in \mathbb{R}^{(\bar{K}+c)\times(\bar{K}+c)}$ can be computed with $O(k_m \bar{K} N + c^2 N)$ (for the computation of $\mathcal{A}$ and $\mathcal{B}$) plus $O((\bar{K}+c)^2 k_m)$ (for the matrix-matrix multiplication $\mathcal{A}\mathcal{B}^T$) operations. The costs for the SVD of K depend only on the small number $\bar{K} + c$ and, thus, not on N. Summarizing, the costs for the update of the SVD are linear in N. After the update, the new SVD can be truncated again to keep a small but accurate representation.

4.2. Efficient Calculation of $\widetilde{W}_k$. Instead of recomputing $\widetilde{W}_k$ in every iteration, we update $\widetilde{W}_k := W_k - J_{prev}^{-1} V_k$ by adding the new column $\Delta\tilde{x}_{k-1}^k - J_{prev}^{-1}\Delta R_{k-1}^k$.

5. Numerical Results. The proposed MVJ restart variants influence the accuracy of the inverse Jacobian approximation. We study the effects of this influence for a one-dimensional flexible tube scenario [7] which allows for a detailed analysis and fast prototyping using MatLab due to it's simplicity while still offering characteristic challenges in coupled FSI simulation. We give a very brief scenario description, omitting formulas and details. The fluid flow is assumed to be incompressible and inviscid. The 1D model is obtained by averaging over the tube in radial direction, as shown in Figure 5.1 (left). The conservation of mass and momentum simplifies to

$$\partial_t(au) + \partial_x(au^2) + a\partial_x p = 0 \quad \text{and} \quad \partial_t + \partial_x(au) = 0$$

with the inflow velocity u, the kinematic pressure p and the cross sectional area of the tube a. We use a time varying, sine-shaped inlet velocity and non-reflecting outlet boundary conditions. The elastic wall is modelled by a Hookean constitutive law neglecting inertia. This enhances instabilities. The fluid exerts stresses on the structure only in radial direction leading to a purely radial motion of the tube wall. We simulate the scenario over one full period of the inlet velocity $[0; T]$ with 100 time steps. Following [7], we use the dimensionless structural stiffness κ, the dimensionless step size τ and the spatial resolution N as parameters characterizing the difficulty: The coupling becomes more challenging for decreasing structural stiffness κ, decreasing time step size τ and increasing spatial resolution N.

Figure 5.1 shows the averaged number of iterations numbers for the implicit MVJ method and the LS method (with reuse of eight time steps). The performance of the LS method suffers from its strong dependency on the *number of reused time steps* (R): We obtain in-acceptable numbers of iterations for zero-reuse, for $R = 8$, the difficult cases (small τ and κ) show excellent results, but the method diverges for the easier cases. A comparison of the restart variants is given in Figure 5.2 for different chunk sizes M of the MVJ estimation era and four parameter settings of (τ, κ). The clear-all method RS-0 yields the worst performance as expected unless M is large enough. As RS-0 yields a mixture of LS($R = 0$) and the MVJ method, it inherits the bad LS($R = 0$) convergence. In contrast, explicitly re-using multi-secant information at restart with the RS-LS method shows good results for small chunk sizes $M = 1, 2$ for all cases except the hardest setting ($\tau = 0.001, \kappa = 10$). Here, the amount of re-used information across MVJ-chunk borders is not sufficient for small M. Nonetheless, the good results for RS-LS($R_{RS} = 2, M = 2$) raise hopes for a cheap but efficient fixed-point acceleration method for scenarios with moderate instabilities. However, the mixture with the LS(R) method reintroduces the dependency on the reuse parameter $R = R_{RS}$ (only in a less sensitive way) as shown in Figure 5.3, where we vary the *number of reused time steps at restart* (R_{RS}) for different chunk sizes. Once again, $M = 2$ shows superior performance for arbitrary numbers of reused time steps at restart (except for the last case), but the optimum is obtained for $R_{RS} = 2$ (case 1,2) and $R_{RS} = 8$ (case 3), respectively, which is also the optimal parameter R for the LS method. The optimum of RS-LS($M = 2, R_{RS}$) yields lower numbers of iterations than the MVJ method as it benefits from both reuse approaches.

The RS-SVD restart approach yields the most robust and reliable results, even in the face of severe instabilities (case 4). The cut-off parameter ϵ which controls the

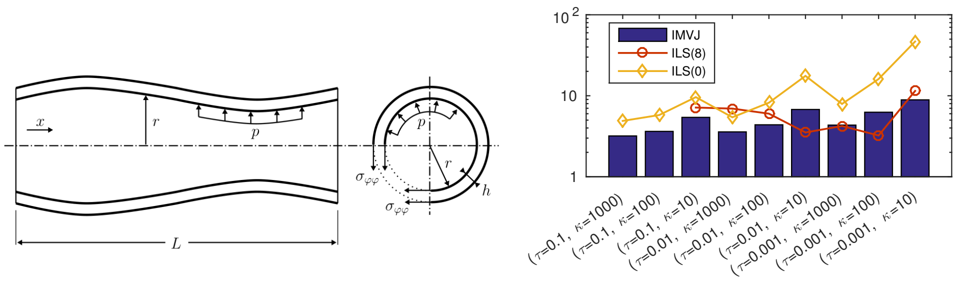

Fig. 5.1: *Left*: Schematic drawing of the deformed flexible tube. The fluid pressure acting on the inner tube walls leads to deformation in radial direction. *Right*: Comparison of the averaged number of coupling iterations of LS (with eight reused timesteps) and MVJ.

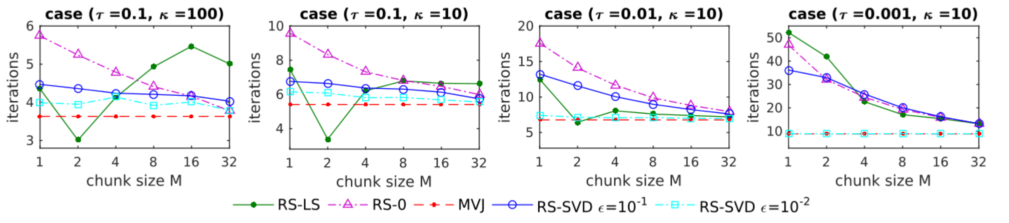

Fig. 5.2: Averaged number of iterations for the restart methods from Sect. 4, for different sizes of the MVJ estimation era M.

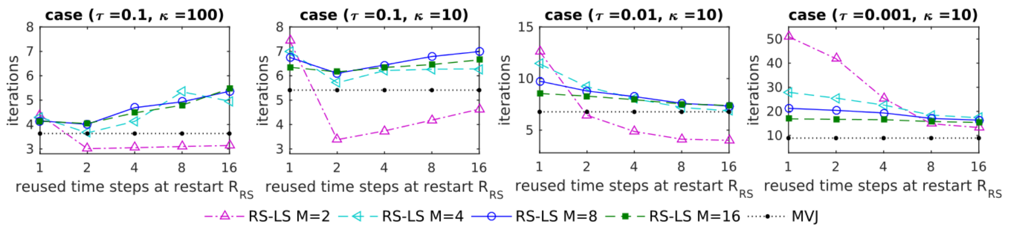

Fig. 5.3: Averaged number of iterations for RS-LS restart method for different numbers of reused time steps at restart R_RS, and chunk sizes M.

Jacobian truncation, however, has to be chosen properly as can be seen in Figure 5.2 where $\epsilon = 10^{-1}$ is too restrictive. For the efficiency of the RS-SVD method, it is crucial that the actual rank of the Jacobian is small compared to the number of interface elements N and independent from N. Figure 5.4 shows the rank of the Jacobian (grey bars) together with the rank of the truncated reduced order model, depending on the cut-off parameter ϵ for different N (corresponding to $2(N+1)$ unknowns). Depending on the difficulty of the test case, the rank grows, but it keeps (nearly) constant for increasing N. Thus, the RS-SVD restart method is expected to be well-suited and efficient for coupled FSI simulation – even in case of strong instabilities.

Conclusion. We have analysed two different quasi-Newton methods suited for partitioned multi-physics simulations using black-box solvers. The numerical results and the complexity analysis of the underlying algorithms showed that the novel combination of the MVJ method with a careful algorithm design based on efficient decompositions of the inverse Jacobian approximations and their truncation yields a very powerful and robust iterative solver for non-linear interface equations. In particular, it comes without the strong sensitivity on unknown solver parameters and has an optimal linear complexity.

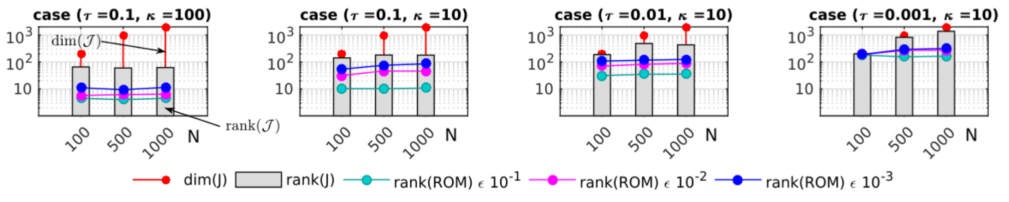

Fig. 5.4: Rank (grey bars) of the truncated Jacobian (ROM) for different values of truncation parameter ϵ and numbers of unknowns N (orange stems) at the interface.

REFERENCES

[1] D. G. Anderson, *Iterative Procedures for Nonlinear Integral Equations*, Journal of the ACM, 12 (1965), pp. 547–560.

[2] K.-J. Bathe and G. A. Ledezma, *Benchmark problems for incompressible fluid flows with structural interactions*, Computers & Structures, 85 (2007), pp. 628–644.

[3] M. Brand, *Fast low-rank modifications of the thin singular value decomposition*, Linear Algebra and its Applications, 415 (2006), pp. 20–30.

[4] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Mathematics of computation, (1965), pp. 577–593.

[5] H.-j. Bungartz, F. Lindner, M. Mehl, K. Scheufele, A. Shukaev, and B. Uekermann, *Partitioned Fluid-Structure-Acoustics Interaction on Distributed Data Coupling via preCICE*, Lecture Notes on Computational Science and Engineering (LNCSE), Springer, Berlin, Heidelberg, New York, 2016. submitted.

[6] J. Degroote, K. J. Bathe, and J. Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction*, Computers and Structures, 87 (2009), pp. 793–801.

[7] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels, *Stability of a coupling technique for partitioned solvers in FSI applications*, Computers & Structures, 86 (2008), pp. 2224–2234.

[8] H.-R. Fang and Y. Saad, *Two classes of multisecant methods for nonlinear acceleration*, Numerical Linear Algebra with Applications, 16 (2008), pp. 197–221.

[9] M. A. Fernández and M. Moubachir, *A Newton method using exact jacobians for solving fluid-structure coupling*, Computers and Structures, 83 (2005), pp. 127–142.

[10] B. Gatzhammer, *Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions*, PhD thesis, Technische Universität München, 2014.

[11] R. Haelterman, *Analytical study of the Least Squares Quasi-Newton method for interaction problems*, PhD thesis, Ghent University, 2009.

[12] R. Haelterman, J. Degroote, D. Van Heule, and J. Vierendeels, *The Quasi-Newton Least Squares Method: A New and Fast Secant Method Analyzed for Linear Systems*, SIAM Journal on Numerical Analysis, 47 (2009), pp. 2347–2368.

[13] F. Lindner, M. Mehl, K. Scheufele, and B. Uekermann, *A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction*, in Proceedings ECCOMAS Coupled Problems, Venice, 2015, pp. 1–12.

[14] P. Lott, H. Walker, C. Woodward, and U. Meier-Yang, *An accelerated Picard method for nonlinear systems related to variably saturated flow*, Advances in Water Resources, 38 (2012), pp. 92–101.

[15] C. Michler, *An interface NewtonKrylov solver for fluidstructure interaction*, International journal for numerical methods in fluids, 47 (2004), pp. 1189–1195.

[16] C. Michler, H. Van Brummelen, and R. De Borst, *An investigation of Interface-GMRES(R) for fluid-structure interaction problems with flutter and divergence*, Computational Mechanics, 47 (2011), pp. 17–29.

[17] K. Scheufele, *Robust Quasi-Newton Methods for Partitioned Fluid-Structure Simulations*, 2015. Master thesis, University of Stuttgart.

[18] A. Toth and C. T. Kelley, *Convergence analysis for Anderson acceleration*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 805–819.

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

[20] B. Uekermann, H.-J. Bungartz, B. Gatzhammer, and M. Mehl, *A parallel, black-box coupling for fluid-structure interaction*, in Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2013, S. Idelsohn, M. Papadrakakis, and B. Schrefler, eds., Stanta Eulalia, Ibiza, Spain, 2013.

[21] J. Vierendeels, J. Degroote, S. Annerel, and R. Haelterman, *Stability issues in partitioned FSI calculations*, in Fluid Structure Interaction II, H.-J. Bungartz, M. Mehl, and M. Schäfer, eds., Lecture Notes in Computational Science and Engineering, Heidelberg, Structure/Darupringer Berlin, 2010, pp. 83–102.

[22] H. F. Walker and P. Ni, *Anderson acceleration for fixed-point iterations*, SIAM J. Numer. Anal., 49 (2011), pp. 1715–1735.