

THE PERFORMANCE IMPACT OF NEWTON ITERATIONS PER SOLVER CALL IN PARTITIONED FLUID-STRUCTURE INTERACTION

THOMAS SPENKE*, NORBERT HOSTERS* AND MAREK BEHR*

*Chair for Computational Analysis of Technical Systems (CATS)
Center for Simulation and Data Science (JARA-CSD)
RWTH Aachen University
Schinkelstraße 2, 52062 Aachen, Germany
e-mail: {spenke,hosters,behr}@cats.rwth-aachen.de,
web page: <http://www.cats.rwth-aachen.de>

Key words: Partitioned Algorithm, Fluid-Structure Interaction, Newton Iterations

Abstract. The cost of a partitioned fluid-structure interaction scheme is typically assessed by the number of coupling iterations required per time step, while ignoring the Newton loops within the nonlinear sub-solvers. In this work, we discuss why these single-field iterations deserve more attention when evaluating the coupling's efficiency and how to find the optimal number of Newton steps per coupling iteration.

1 INTRODUCTION

Partitioned algorithms enjoy great popularity in the fluid-structure interaction (FSI) community: Treated as black boxes, the fluid and structural solvers are coupled only via the exchange of interface data. Over the last decade, the inherent drawbacks regarding stability were mitigated significantly, e.g., by interface quasi-Newton methods [1, 2, 3].

Assuming the cost for data exchange to be negligible, the efficiency of a partitioned scheme is typically assessed by the number of coupling iterations, i.e., solver calls, required per time step. When coupling two nonlinear solvers, however, the cost of one solver call is not constant, but depends on various parameters. In particular, it is closely connected to the number of Newton iterations performed. On the one hand, this supports the conclusion that the Newton iterations required per time step provide a much better measure for the coupling's efficiency. On the other hand, it raises a central question: How many Newton iterations should best be run per coupling step? The answer is not trivial: While always iterating to full convergence produces unnecessary overhead, running too few steps brings the risk of impeding stability by feeding back defective data into the coupling loop.

This work discusses the impact of the Newton iterations per solver call and proposes an adaptive choice to improve the efficiency of partitioned fluid-structure interaction schemes.

2 PARTITIONED FLUID-STRUCTURE INTERACTION

Although the key aspects of this work are expected to hold true for other coupled multi-physics simulations as well, we restrict ourselves to partitioned fluid-structure interaction. More precisely, we consider an incompressible fluid in the domain Ω^f interacting with an elasto-dynamic structure Ω^s .

2.1 Fluid Subproblem

The fluid velocity $\mathbf{u}^f(\mathbf{x}, t)$ and its pressure $p^f(\mathbf{x}, t)$ are governed by the unsteady Navier-Stokes equations for an incompressible fluid:

$$\rho^f \left(\frac{\partial \mathbf{u}^f}{\partial t} + \mathbf{u}^f \cdot \nabla \mathbf{u}^f - \mathbf{b}^f \right) - \nabla \cdot \mathbf{T}^f = \mathbf{0} \quad \text{in } \Omega_t^f \quad \forall t \in [0, T], \quad (1a)$$

$$\nabla \cdot \mathbf{u}^f = 0 \quad \text{in } \Omega_t^f \quad \forall t \in [0, T], \quad (1b)$$

where ρ^f is the constant fluid density, while \mathbf{b}^f denotes the resultant of all external body forces per unit mass of fluid. Assuming a Newtonian fluid with dynamic viscosity μ^f , Stokes' law models the Cauchy stress tensor as $\mathbf{T}^f(\mathbf{u}^f, p^f) = -p^f \mathbf{I} + \mu^f (\nabla \mathbf{u}^f + (\nabla \mathbf{u}^f)^T)$. The problem is closed by a divergence-free initial velocity field as well as an appropriate set of boundary conditions on $\Gamma^f = \partial\Omega^f$.

The fluid problem is simulated by our in-house solver XNS, using stabilized P1P1 finite elements in space [4, 5] and a BDF1 scheme in time [6]. The ALE mesh is adapted to deforming domains via the linear elastic mesh-update method(EMUM) [7].

2.2 Structural Subproblem

The structural displacement $\mathbf{d}^s(\mathbf{x}, t)$ is given by the dynamic balance of stresses. From a Lagrangian viewpoint with respect to the undeformed configuration Ω_0^s , it reads

$$\rho^s \frac{d^2 \mathbf{d}^s}{dt^2} = \nabla_0 \cdot (\mathbf{S} \mathbf{F}^T) + \mathbf{b}^s \quad \text{in } \Omega_0^s \quad \forall t \in [0, T], \quad (2)$$

where ρ^s denotes the material density and \mathbf{b}^s the resultant of all body forces per unit volume. The 2nd Piola-Kirchhoff stresses \mathbf{S} are defined based on the Cauchy stress tensor \mathbf{T}^s and the deformation gradient \mathbf{F} as $\mathbf{S} := \det(\mathbf{F}) \mathbf{F}^{-1} \mathbf{T}^s \mathbf{F}^{-T}$.

As constitutive equation, the St. Venant-Kirchhoff material model provides the linear stress-strain law $\mathbf{S} = \mathbf{C}^s : \mathbf{E}$, where the constant matrix \mathbf{C}^s depends on two material parameters, e.g., Young's modulus E^s and Poisson's ratio ν^s . The definition of the Green-Lagrange strains $\mathbf{E} := \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I})$ introduces a geometrical nonlinearity into the structural model [8, 9].

A closed problem formulation requires both an initial displacement field (typically zero) and a set of Dirichlet and Neumann boundary conditions. The structural problem is then solved by the in-house finite-element code FEAFa using Lagrangian finite elements or isogeometric analysis (IGA) [10, 11] in space and a generalized- α scheme in time [12, 13].

2.3 Coupling Conditions

Naturally, in fluid-structure interaction the solution fields are not independent, but instead connected at the shared interface $\Gamma^{fs} = \partial\Omega^f \cap \partial\Omega^s$ [14]:

1. The *kinematic* coupling condition states the continuity of displacements, i.e., $\mathbf{d}^f = \mathbf{d}^s$ on Γ^{fs} , which directly implies the equality of velocities and accelerations, too.
2. Following Newton’s third law, the *dynamic* condition requires the equality of interface tractions: $\mathbf{T}^f \mathbf{n}^f = \mathbf{T}^s \mathbf{n}^s$ on Γ^{fs} , where \mathbf{n}^f and \mathbf{n}^s are the normal vectors.

Satisfying these coupling conditions for every point in time, i.e., in a continuous manner, ensures the conservation of mass, momentum, and energy over the FSI boundary [15].

2.4 Dirichlet-Neumann Scheme

This work relies on a partitioned FSI algorithm, meaning the two subproblems are addressed by two distinct solvers, that are coupled only via the exchange of interface data. While this strategy features a high flexibility regarding the solvers, their communication requires some additional considerations: (1) Since the meshes in general do not match at the interface, a conservative projection is needed, the *spatial coupling* [14]. (2) The interdependency of the two subproblems requires an iterative procedure to find a consistent solution of the coupled problem, referred to as *temporal coupling* [16, 17].

The most common temporal coupling algorithm for FSI problems is the *Dirichlet-Neumann scheme*: While the fluid tractions are passed as a Neumann boundary condition to the structure (dynamic continuity), the resulting interface deformation poses a Dirichlet condition for the fluid velocity (kinematic continuity). For each time step, the two solvers are successively called in a Gauss-Seidel iteration until convergence is reached [17].

As a partitioned FSI algorithm, the Dirichlet-Neumann scheme suffers from an inherent instability, caused by the *added-mass effect* [18, 19, 20]. Basically, it is characterized by overestimated deformations causing exaggerated fluid inertia terms and vice versa.

A common countermeasure is to augment the Dirichlet-Neumann scheme by an update step: Typically, the computed interface deformation $\tilde{\mathbf{x}}^k$ is modified before it is passed back to the fluid as \mathbf{x}^{k+1} . The simplest version is a *relaxation*, i.e., $\mathbf{x}^{k+1} = \omega \tilde{\mathbf{x}}^k + (1 - \omega) \mathbf{x}^k$, with the relaxation factor ω being either some constant $\omega < 1$ (“under-relaxation”) or updated dynamically, e.g., via *Aitken’s relaxation* [21, 22].

A more sophisticated approach are *interface quasi-Newton (IQN)* methods, which use a Newton-like update based on an approximated Jacobian of the coupled problem. The Jacobian approximation is successively improved by collecting information from the intermediate results of each coupling iteration. This way, IQN methods almost completely overcome the added-mass difficulty [1, 2, 3].

3 Newton Iterations per Solver Call

A central aspect of partitioned schemes for fluid-structure interaction is the concept of treating the single-field solvers as black boxes, in that only their in- and output, but not their interior setup and techniques, are known. Combined with the idea that the solver calls are the most expensive part of the simulation, increasing the algorithm's efficiency comes down to decreasing the number of coupling iterations required for convergence. This implicitly assumes the cost of one solver call to be constant - which is far from true for nonlinear solvers, but the best we can do without knowing any internal details.

3.1 Newton Iterations and Computational Cost

In practice, however, this full black-box case is rather uncommon, as the user or developer typically has access to either the solver's input configuration, such as the number of Newton iterations performed per call, or even to the source code. This naturally raises questions about the Newton iterations' effect on the computational cost.

Of course it is impossible to take all effects into account, but one Newton iteration basically corresponds to one repetition of the numerical solution procedure, i.e., assembling the matrix-vector system and solving it for the Newton increment. This supports the conclusion that assuming the cost of a solver call to scale with the number of Newton iterations performed is more accurate than assuming constant cost.

With that, the cost of the simulation depends not only on the total number of coupling steps $N_{Coupling}$, but also the Newton iterations N_{Newton}^i run for each subproblem i :

$$\mathbf{cost}(simulation) \approx N_{Coupling} \cdot \mathbf{cost}(data\ transfer) + \sum_{i=f,s} N_{Newton}^i \cdot \mathbf{cost}(Newton^i) .$$

As one Newton iteration is typically much more expensive than the data transfer within one coupling iteration, i.e., $\mathbf{cost}(Newton^i) \gg \mathbf{cost}(data\ transfer)$, this work uses the sum of Newton iterations $N_{Newton} = \sum_{i=f,s} N_{Newton}^i$ as the main efficiency measure for the partitioned scheme. Based on that, focus is put on how it is influenced by the number of Newton steps run per solver call. The relation is non-trivial as running too few iterations brings the risk of feeding inaccurate data back into the coupling loop, while with too many Newton iterations computational time is misspend on polishing up a solution that will be overwritten in the next coupling step anyway.

The remainder of this chapter is outlined as follows: Section 3.2 defines the convergence criteria used in this work, before Section 3.3 and 3.4 discuss the effect of the Newton iterations per solver call for two numerical examples. Drawing conclusions from this, Section 3.5 proposes new approaches for choosing the number of Newton iterations dynamically and investigates their effectiveness.

3.2 Convergence Criteria

Before looking into numerical examples, it is essential to define the convergence criteria:

1. A subproblem is considered converged when its residual vanishes, i.e., is lower than the bound $\varepsilon_{Problem}$. In that case, the problem's Newton loop terminates independently from the iteration number. We will refer to this as *single-field convergence*.
2. In line with the fixed-point character of the partition scheme [1, 21], *Coupling convergence* is reached if the solutions of all subproblems stay virtually unchanged within one coupling iteration, i.e., all relative changes are lower than $\varepsilon_{Coupling}$.

If and only if both conditions are satisfied, the coupled simulation has converged and proceeds with the next time step.

3.3 Example: Tank with Elastic Bottom

The first numerical example of this work is depicted in Figure 1: A square tank with rigid side walls is filled by an incompressible fluid. For simplicity, the fluid domain has a natural Neumann boundary on the top rather than a free surface, allowing for a free in- and outflow. Its elastic bottom is clamped on both ends and deforms due to the fluid's gravity. The material and geometrical properties are listed in Figure 1a.

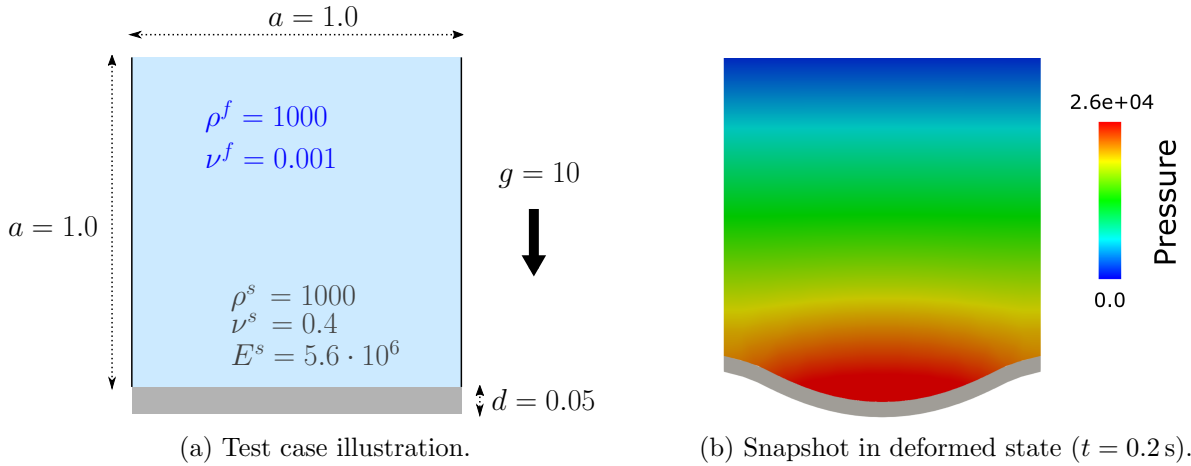


Figure 1: Elastic bottom test case.

To simplify the parameter study, the discretization is rather coarse: While the fluid mesh has 400 triangular finite elements, the structural problem is solved by 60 isogeometric elements of spline degree 2. The simulations are run for 50 time steps of size $\Delta t = 0.01$.

To handle the added-mass instability, which is expected to be strong because of the high density ratio $\rho^f/\rho^s = 1.0$, the structural deformation is updated via the IQN-IMVLS method [3]. The convergence criteria were chosen as $\varepsilon_{Coupling} = 10^{-5}$ for the coupling and $\varepsilon_{Problem} = 10^{-10}$ for both fluid and structure.

This test case was run for different combinations of the Newton iterations per solver call for the fluid and the structural subproblem, from now on referred to as N^f and N^s .

Table 1: Iterations required for different choices of N^f and N^s . The bold numbers denote the **coupling iterations**, the underlined ones the total Newton iterations. The single-field Newton iterations are colored **blue for the fluid** and **brown for the structural solver**.

		Structural Newton Iterations N^s											
		1		2		3		4		5		∞	
Fluid Newton Iterations N^f	1	617	<u>1166</u>	567	<u>1316</u>	583	<u>1499</u>	605	<u>1622</u>	573	<u>1546</u>	573	<u>1546</u>
		617	567	567	549	583	549	605	549	573	549	573	549
	2	497	<u>1488</u>	460	<u>1613</u>	438	<u>1667</u>	460	<u>1836</u>	461	<u>1837</u>	461	<u>1837</u>
		994	494	919	694	875	792	919	917	922	915	922	915
	3	468	<u>1872</u>	366	<u>1708</u>	366	<u>1843</u>	374	<u>1974</u>	381	<u>2014</u>	381	<u>2016</u>
		1404	468	1097	611	1098	745	1122	852	1143	871	1143	873
	4	407	<u>2011</u>	342	<u>1939</u>	335	<u>2043</u>	352	<u>2225</u>	352	<u>2240</u>	352	<u>2240</u>
		1604	407	1342	597	1315	728	1386	839	1386	854	1386	854
∞	5	414	<u>2300</u>	335	<u>2108</u>	336	<u>2264</u>	352	<u>2441</u>	355	<u>2460</u>	355	<u>2460</u>
		1886	414	1523	585	1528	736	1602	839	1604	856	1604	856
∞	∞	412	<u>2410</u>	343	<u>2237</u>	333	<u>2333</u>	355	<u>2565</u>	354	<u>2547</u>	354	<u>2547</u>
		1998	412	1639	589	1609	724	1713	852	1700	847	1700	847

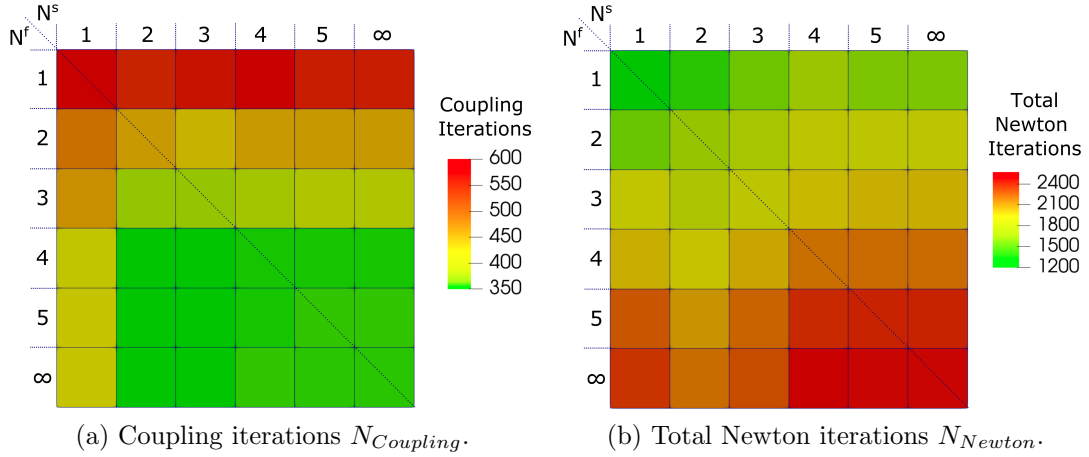


Figure 2: Influence of different choices for N^f and N^s on the required number of coupling $N_{Coupling}$ and Newton iterations N_{Newton} , illustrated by color scale.

Based on that, Table 1 lists the number of coupling and Newton iterations - for each subproblem as well as in total. Figure 2 visualizes the data in two color plots.

As expected, the results show that increasing the number of Newton iterations per call leads to fewer coupling iterations and vice versa. Interestingly, however, the total number of Newton iterations N_{Newton} shows the opposite trend: Running fewer or even just one Newton iteration per solver call requires fewer Newton steps. Following the arguments from Section 3.1, the computational cost are therefore expected to be lower too.

An explanation is that exchanging data after every Newton iteration makes sure to always use the most recent solution of the other subproblem, keeping the boundary conditions up to date. Moreover, in case an IQN method is applied, every Newton iteration adds a new data pair and hence improves the Jacobian approximation.

Another interesting observation is that if the Newton iterations per solver call are kept fixed for one problem, increasing it for the other one reduces $N_{Coupling}$. This effect can be very useful if the cost of the two subproblems are very different. A common example would be a very complex fluid problem coupled to a rather coarse structural simulation: In that case setting $N^f = 1$ and $N^s > 1$ is expected to yield the best performance.

3.4 Example: Elastic Beam in Channel Flow

While the observations made for the first example are typical for FSI problems with a strong interdependency, the characteristics of the second test case, illustrated in Figure 3, are different: Since the elastic beam positioned in the channel flow is rather heavy, the added-mass effect is less emphasized. Instead, the coupled system is mainly driven by the flow problem, so that an under-relaxation of the interface deformation with $\omega = 0.8$ is sufficient to stabilize it.

The fluid problem is discretized by 771 triangular finite elements, the structure by 60 isogeometric elements of spline degree 2. The simulations run for 50 time steps of width $\Delta t = 0.005$. Convergence is triggered by $\varepsilon_{Coupling} = 10^{-5}$ and $\varepsilon_{Problem} = 10^{-10}$.

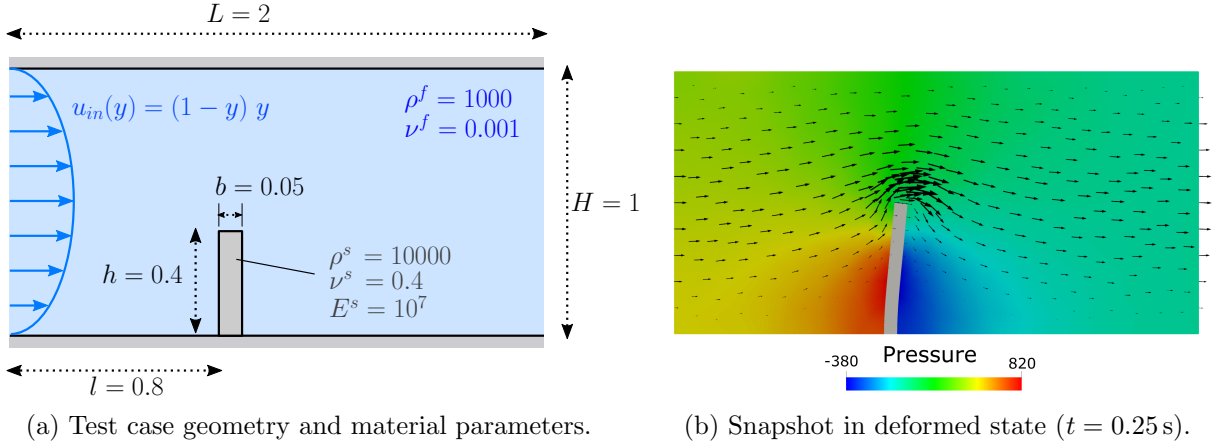


Figure 3: Elastic beam in channel flow test case.

Just as in the previous example, Table 2 lists the required coupling iterations $N_{Coupling}$, the single-field Newton iterations, as well as the total Newton iterations N_{Newton} , depending on the choices for N^f and N^s , i.e., the Newton steps per solver call. Figure 4 visualizes the data in two heat maps. The columns for $N^s > 3$ are omitted, as none of the simulations required more than three Newton iterations in any structural solver call.

Table 2: Newton and coupling iterations required for different choices of N^f and N^s . The formatting is equivalent to that of Table 1: **coupling iterations**, total Newton iterations, fluid's Newton iterations, structural Newton iterations.

		Structural Newton Iterations N^s							
		1		2		3		∞	
Fluid Newton Iterations N^f	1	1083	<u>2109</u>	1083	<u>2548</u>	1083	<u>2611</u>	1083	<u>2611</u>
		<u>1083</u>	<u>1026</u>	<u>1083</u>	<u>1465</u>	<u>1083</u>	<u>1528</u>	<u>1083</u>	<u>1528</u>
	2	901	<u>2702</u>	901	<u>3142</u>	901	<u>3205</u>	901	<u>3205</u>
		<u>1802</u>	<u>900</u>	<u>1802</u>	<u>1340</u>	<u>1802</u>	<u>1403</u>	<u>1802</u>	<u>1403</u>
	3	721	<u>2884</u>	721	<u>3324</u>	721	<u>3386</u>	721	<u>3386</u>
		<u>2163</u>	<u>721</u>	<u>2163</u>	<u>1161</u>	<u>2163</u>	<u>1223</u>	<u>2163</u>	<u>1223</u>
4		718	<u>3485</u>	718	<u>3925</u>	718	<u>3988</u>	718	<u>3988</u>
		<u>2767</u>	<u>718</u>	<u>2767</u>	<u>1158</u>	<u>2767</u>	<u>1221</u>	<u>2767</u>	<u>1221</u>
5		718	<u>3867</u>	718	<u>4307</u>	718	<u>4370</u>	718	<u>4370</u>
		<u>3149</u>	<u>718</u>	<u>3149</u>	<u>1158</u>	<u>3149</u>	<u>1221</u>	<u>3149</u>	<u>1221</u>
∞		718	<u>4014</u>	718	<u>4454</u>	718	<u>4517</u>	718	<u>4517</u>
		<u>3296</u>	<u>718</u>	<u>3296</u>	<u>1158</u>	<u>3296</u>	<u>1221</u>	<u>3296</u>	<u>1221</u>

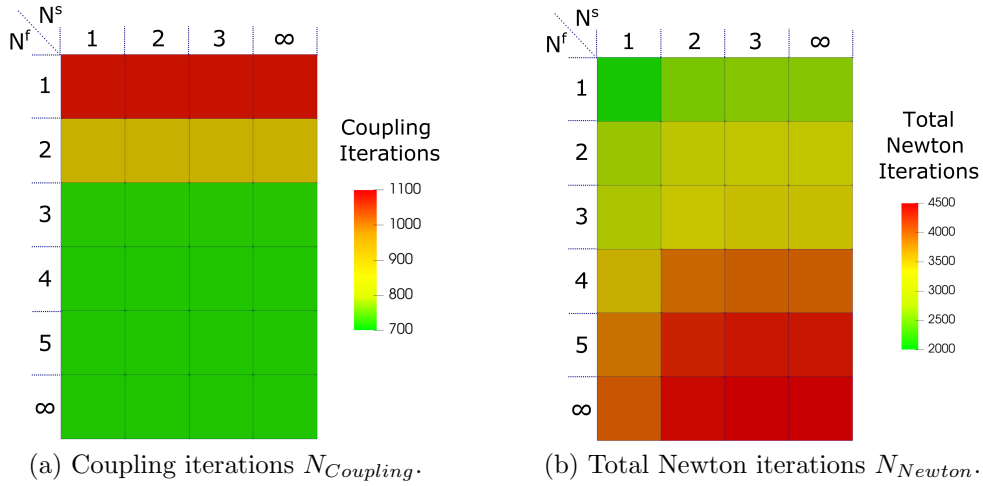


Figure 4: Influence of different choices for N^f and N^s on the required number of coupling $N_{Coupling}$ and Newton iterations N_{Newton} , illustrated by color scale.

This already points out the biggest difference to Table 1, the negligible influence of N^s on the coupling iterations $N_{Coupling}$. It is a result of the structure's lower sensitivity to the flow solution, i.a., due to the decreased density ratio of $\rho^f/\rho^s = 0.1$.

For the number of Newton iterations performed per fluid solver call N^f , in contrast, the same relation to $N_{Coupling}$ as in the previous test case can be observed: While increasing N^f results in fewer coupling steps, choosing $N^f = N^s = 1$ again yields the smallest number of Newton iterations and therefore supposedly also the lowest computational cost.

3.5 Adapt Newton Iterations Dynamically

The results discussed so far identified running just one Newton iteration per solver call as a good and efficient choice. Nevertheless, it is not without flaws in that it typically requires many coupling iterations, which might produce unwanted coupling overhead. This motivated the idea of adapting N^f and N^s dynamically. We propose two approaches:

1. The frequent data exchange is of particular importance in the first Newton iterations, in which the solutions still change a lot. In later steps, however, the increments are decreasing until they might no longer justify the effort of coupling after every iteration. As a remedy, we suggest to run just one Newton step per solver call, $N^f = N^s = 1$, as long as coupling convergence (see Section 3.2) is not reached. Once it is, the two solvers switch to full single-field convergence, $N^f = N^s = \infty$. We will refer to this technique as *N1-CC approach* (“one Newton iteration until coupling convergence”). Naturally, further variants are obtained by running more than one iteration for the unconverged coupling, i.e., N2-CC, N3-CC, etc. Note that in case after setting $N^f = N^s = \infty$ the coupling convergence criterion is not satisfied anymore, N^f and N^s are switched back. In practice, however, this issue never occurred in any simulation of this work. If it would, using a stricter criterion, e.g., $0.1 \cdot \varepsilon_{Coupling}$, for triggering full single-field convergence could fix it.
2. The goal of the second suggestion is to avoid feeding back inaccurate data into the coupling loop. Therefore, the *converged interface data* approach runs the Newton loop until the relative change in the coupling data, e.g., the fluid tractions, is less than some bound ε_{CID} . In contrast to the N1-CC method, this approach typically results in additional Newton steps only in the first few coupling iterations.

Revisiting the two test cases from Sections 3.3 and 3.4, Table 3 investigates the effectiveness of these adaptive techniques. It clearly shows that the suggested N1-CC variant significantly reduces the number of coupling iterations $N_{Coupling}$ compared to the choice $N^f = N^s = 1$, overcoming its major drawback. At the same time, its main strength, i.e., the very low total number of Newton iterations N_{Newton} , is retained. In fact, the results indicate that N_{Newton} is even further decreased for the given examples.

These observations can be explained by the reduced data exchange of the N1-CC approach once coupling convergence is reached: Aside from the evident effect on $N_{Coupling}$, the Newton iterations are no longer synchronized. Therefore, an already converged problem will not be called for each iteration of the other, non-converged problem, avoiding unnecessary overhead. Moreover, depending on $\varepsilon_{Coupling}$ and $\varepsilon_{Problem}$, the remaining increments might satisfy the coupling convergence, yet still slow down the other problem’s Newton loop by slightly updating its boundary conditions in every iteration.

Of course these investigations are mainly based on examples, but nevertheless they indicate a superiority of the dynamic N1-CC method over using fixed numbers of iterations. Beyond that, the results of the N3-CC variant are interesting for cases in which more

Table 3: Iterations required for the adaptive techniques (formatting as in previous tables). The results in gray are repeated for simpler comparison.

Setting	Elastic Bottom		Beam in Channel Flow	
N1-CC	417	945	768	1637
	528	417	869	768
N3-CC	355	1799	720	3384
	1071	728	2161	1223
Converged Interface Data, $\varepsilon_{CID} = 10^{-4}$	572	1720	1083	3182
	928	792	1880	1302
$N^f = N^s = 1$	617	1166	1083	2109
	617	567	1083	1026
$N^f = N^s = \infty$	354	2547	718	4517
	1700	847	3296	1221

focus is put on the coupling iterations, e.g., if the data exchange is expensive, because it reaches the same number of coupling iterations as $N^f = N^s = \infty$ in less Newton steps.

The converged interface data approach, on the other hand, cannot keep up with neither $N^f = N^s = 1$ nor the N1-CC approach. This indicates that the risk of passing back inaccurate data into the coupling loop is not very severe. Instead, the additional Newton steps in the first coupling iterations, in which the interface data has not converged yet, prove to be rather ineffective, which is in line with the discussion on the N1-CC approach.

4 Conclusion

This work revolves around one central question: In partitioned fluid-structure interaction with two non-linear subproblems, how many Newton iterations per solver call result in the most efficient coupling scheme?

Its motivation arises from the argument that the required Newton iterations are a better measure for a partitioned algorithm’s computational efficiency than the number of coupling iterations; because rather than assuming constant cost per solver call, they reflect the reiterative nature of the numerical solution procedure, i.e., the Newton loop.

Based on typical examples, the discussion shows that iterating to full convergence for every solver call does in fact require the fewest coupling steps. The total number of Newton iterations and therefore the computational cost, however, can be reduced significantly by a more frequent communication, in particular by running just one Newton step per call.

Against this backdrop, this work discusses two adaptive choices for the number of Newton iterations per solver call. In particular, the N1-CC approach exchanges data after every Newton iteration only until coupling convergence is reached; after that, every solver call iterates to full convergence. The numerical examples confirm that this technique maintains (and even further decreases) the low number of Newton iterations, while reducing the number of coupling iterations substantially.

Although the discussion is purely based on numerical experiments and logical arguments rather than a firm mathematical foundation, its findings provide valuable assistance and guidelines on how to properly set the number of Newton iterations per solver call in a partitioned fluid-structure interaction scheme.

In the end, the optimal choice of course depends on a variety of factors, as for example the time step size, the specific implementation, or in general the problem at hand. Therefore, future works will extend the investigation to a wider set of parameters.

REFERENCES

- [1] Lindner, F. and Mehl, M. and Scheufele, K. and Uekermann, B. A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction. *COUPLED VI: proceedings of the VI International Conference on Computational Methods for Coupled Problems in Science and Engineering* (2015) 477–485.
- [2] Degroote, J. and Bathe, K.J. and Vierendeels, J. Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. *Computers & Structures* (2009) **87**:793–801.
- [3] Spenke, T. and Hosters, N. and Behr, M. A multi-vector interface quasi-Newton method with linear complexity for partitioned fluid–structure interaction. *Comput. Methods Appl. Mech. Eng.* (2020) **361**:112810.
- [4] Pauli, L. and Behr, M. On stabilized space-time FEM for anisotropic meshes: Incompressible Navier–Stokes equations and applications to blood flow in medical devices. *Int. J. Numer. Methods Fluids* (2017) **85**:189–209.
- [5] Donea, J. and Huerta, A. *Finite element methods for flow problems*. John Wiley & Sons, (2003).
- [6] Forti, D. and Dedè, L. Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a high performance computing framework. *Computers & Fluids* (2015) **117**:168–182.
- [7] Behr, M. and Abraham, F. Free-surface flow simulation in the presence of inclined walls, *Comput. Methods Appl. Mech. Eng.* (2002) **191**:5467–5483.
- [8] Bathe, K.J. *Finite element procedures*. TBS, (1996).
- [9] Yibin, F.U. and Ogden, R.W. *Nonlinear elasticity: Theory and applications*, London Mathematical Society Lecture Note Series, Vol. 283, (2001).
- [10] Cottrell, J.A. and Hughes, T.J.R. and Bazilevs, Y. *Isogeometric analysis: Toward integration of CAD and FEA*. John Wiley & Sons, (2009).

- [11] Hughes, T.J.R. and Cottrell, J.A. and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.* (2005) **194**:4135–4195.
- [12] Chung, J. and Hulbert, G.M. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- α method. *Journal of Applied Mechanics* (1993) **60**:371–375.
- [13] Erlicher, S. and Bonaventura, L. and Bursi, O.S. The analysis of the generalized- α method for non-linear dynamic problems. *Computational mechanics* (2002) **28**:83–104.
- [14] Hosters, N. and Helmig, J. and Stavrev, A. and Behr, M. and Elgeti, S. Fluid-structure interaction with NURBS-based coupling. *Comput. Methods Appl. Mech. Eng.* (2018) **332**:520–539.
- [15] Küttler, U. and Förster, C. and Wall, W.A. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains. *Computational Mechanics* (2006) **38**:417–429.
- [16] Degroote, J. and Vierendeels, J. Multi-solver algorithms for the partitioned simulation of fluid–structure interaction. *Comput. Methods Appl. Mech. Eng.* (2011) **200**:2195–2210.
- [17] Hosters, N. *Spline-based methods for fluid-structure interaction*. Dissertation, RWTH Aachen University, (2018).
- [18] Förster, C. *Robust methods for fluid-structure interaction with stabilised finite elements*. Dissertation, University of Stuttgart, (2007).
- [19] Förster, C. and Wall, W.A. and Ramm, E. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Comput. Methods Appl. Mech. Eng.* (2007) **196**:1278–1293.
- [20] Causin, P. and Gerbeau, J.-F. and Nobile, F. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput. Methods Appl. Mech. Eng.* (2005) **194**:4506–4527.
- [21] Küttler, U. and Wall, W.A. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational mechanics* (2008) **43**:61–72.
- [22] Irons, B.M. and Tuck, R.C. A version of the Aitken accelerator for computer iteration. *Int. J. Num. Meth. Engng.* (1969) **1**:275–277.