X International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2023
M. Papadrakakis, B. Schrefler and E. Oñate (Eds)

THE EFFECT OF THE NUMBER OF SUBPROBLEM ITERATIONS IN PARTITIONED FLUID-STRUCTURE INTERACTION SIMULATION

Nicolas Delaissé[*,†], Thomas Spenke[‡], Norbert Hosters[‡] and Joris Degroote[†,§]

[†] Department of Electromechanical, Systems and Metal Engineering,
Ghent University, Sint-Pietersnieuwstraat 41, Ghent, Belgium
e-mail: Nicolas.Delaisse@UGent.be

[‡] Chair for Computational Analysis of Technical Systems (CATS)
Center for Simulation and Data Science (JARA-CSD),
RWTH Aachen University, Schinkelstraße 2, Aachen, Germany

[§] Flanders Make @ UGent – Core Lab MIRO

Key words: Partitioned Algorithm, Solver Iterations, Fluid-Structure Interaction, Coupled Problems

Abstract. In a partitioned fluid-structure interaction simulation separate flow and structure solvers, each with their own spatial domain, are coupled by exchanging data on the common interface. Its computational cost is dominated by the execution of these solvers, and the cost associated with the coupling algorithm and communication are often deemed negligible. From this point of view, the computational cost is in literature typically expressed by the number of required coupling iterations per time step or equivalently the number of solver executions. However, this reasoning implicitly assumes a constant solver cost and ignores the varying number of internal *subproblem iterations*, i.e., solver iterations in the nonlinear solvers.

This work addresses this shortcoming and shows that the computational cost of a partitioned fluid-structure interaction simulation is significantly impacted by the number of subproblem iterations performed in each solver call. Specifically, it is demonstrated that performing subproblem iterations until the solver is fully converged in each call does typically minimize the number of coupling iterations, but does not lead to minimal computational time. Instead, under the assumption of constant subproblem iteration cost, the optimum is found by minimizing a weighted sum of both coupling and subproblem iterations. The weighting factors are determined by the problem itself as well as the computer architecture.

1 INTRODUCTION

In fluid-structure interaction (FSI) simulation, the partitioned approach is often adopted, because it allows to reuse optimized and validated codes for the flow and structural subproblem, in contrast to the monolithic approach where the equations of both subproblems are solved in one system. In the partitioned strategy, the solvers are regarded as black boxes and communication is restricted to their shared fluid-structure interface. Its drawback is the need for

consecutive fluid and structure solver executions with the intermediate exchange of interface data. These consecutive executions within every time step, called coupling iterations, assure that the position of the common interface and the forces thereon are identical up to a certain tolerance in both subproblems, but they also give rise to stability issues due to the added mass effect [1], [2].

A series of techniques has been proposed to counter these instabilities and they range from simply relaxing the change in interface data to applying quasi-Newton techniques, which use an approximated Jacobian. This last approach forms the basis for the current state-of-the-art methods. By calculating the differences between the interface data of consecutive iterations, typically displacement and residual, they construct an approximated low-rank Jacobian, which is used to determine a new interface value for the next iteration. For an extensive overview of implementation and variants of these techniques, the reader is referred to [3].

The central question in this work stems from two observations. First, it is common practice to keep the tolerance of a subproblem constant in every coupling iteration or, in other words, converge every solver execution up to full tolerance. Nevertheless, in the first few coupling iterations the intermediate solution is still far from the final one, which means that computational resources are spent on polishing provisional data. From this perspective it seems beneficial to gradually increase the computational accuracy of the subproblems in consecutive coupling steps. However, this also means that less accurate data is fed back into the coupling loop. Especially when using quasi-Newton techniques, care is warranted since these techniques construct an approximate Jacobian from differences between interface data of consecutive iterations. The accuracy of these differences directly depends on the tolerance of the subproblems, such that loosening these tolerances will also negatively affect the accuracy of the quasi-Newton approximated Jacobian.

Second, in literature, the cost of a partitioned FSI simulation is often expressed by the number of coupling iterations, or equivalently the number of solver executions. While it is true that the cost of the solver calls are the dominant contributor to the calculation time, this reasoning also implicitly assumes a constant cost for every solver execution. In practice, even when the solvers calculate up to full convergence in each call, this is not the case. For example, the cost will decrease from one coupling iteration to the next if the previous solver call is used as starting point. Furthermore, also the time evolution of the subproblems affects this cost, e.g., a sudden change in boundary condition will automatically lead to a higher number of internal iterations and hence a more expensive solver call.

Based on these observations, this work investigates the effect of not fully converging in every solver execution on the total calculation time, by imposing a maximal number of *subproblem iterations* per solver call. Towards this goal, a measure for the computational efficiency is used that takes the weighted sum of not only the number of coupling iterations but also the number of subproblem iterations [4].

The next section discusses the subsolvers and how a consistent level of accuracy is reached when the number of subproblem iterations is limited. Section 3 focuses on the new measure for computational time and describes how the weighting factors can be obtained through regression. Section 4 presents a test case for which the results are discussed in Section 5. Finally, the most important findings are summarized in Section 6.

2 PARTITIONED FSI: COUPLING OF SUBSOLVERS

In a partitioned FSI simulation, separate flow and structure solvers are coupled: by performing multiple coupling iterations in each time step, the kinematic and dynamic equilibrium conditions on the interface are fulfilled up to a certain tolerance. These conditions entail corresponding displacement and forces on the interface in both domains.

In most cases, internal nonlinear iterations are performed in both the flow and structure solver, whether these are solved using finite volumes (FV), finite elements (FE) or another technique. These iterations will be referred to with the general term *subproblem iterations*. These iterations are used to find the solution $\boldsymbol{u}$ of the discretized system

$$\boldsymbol{A}(\boldsymbol{u})\,\boldsymbol{u} = \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ contains the dependent variables of interest, e.g., pressure and velocity for the flow solver, and n is the number of degrees of freedom (DOF) of the system. Subproblem iterations are required since the coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ typically depends on $\boldsymbol{u}$, rendering the system nonlinear.[1] When solving a subproblem separately, the right-hand side $\boldsymbol{b} \in \mathbb{R}^n$ containing the boundary conditions and source terms, on the other hand, is independent of $\boldsymbol{u}$.[2]

However, when two subproblems are coupled together as in FSI, the RHS does depend on $\boldsymbol{u}$. Indeed, the pressure forces applied on the interface in the structure solver depend on the result of the flow simulation, which in turn depends on the displacement, the variable of interest in the structure simulation. Therefore, Eq. (1) is reformulated as

$$\boldsymbol{A}(\boldsymbol{u})\,\boldsymbol{u} = \boldsymbol{b}(\boldsymbol{u}). \tag{2}$$

As a consequence, only the first subproblem iteration in each subsolver call has a RHS corresponding to the current value of $\boldsymbol{u}$. For the subsequent subproblem iterations the RHS will be defective, since it is not updated with the newly obtained $\boldsymbol{u}$. This observation has lead to the introduction of a new convergence criterion in Spenke and Delaissé et al. [4]. This new criterion considers a coupled simulation converged when both subproblems converge in the first subproblem iteration. That way, the need of specifying a residual tolerance for the coupling iterations naturally disappears.

For the research question in this work, it has the additional benefit that the number of subproblem iterations in each solver call can be varied without affecting the final accuracy. With a conventional coupling convergence criterion this would not necessarily be the case: imagine that the maximum number of subproblem iterations in a solver execution is limited, then the coupling residual could drop below the prescribed threshold, before both of the solvers have reached their prescribed tolerance.

3 EQUIVALENT TIME MEASURE

This works investigates the effect of limiting the number of subproblem iterations in each solver call on the total calculation effort. It will become clear in Section 5 that this limitation will increase the number of coupling iterations, but has the potential to significantly shorten

[1] Another possibility is an iterative linear solver.

[2] More often than not, part of the dependence on $\boldsymbol{u}$ is in fact treated explicitly resulting in a lagging contribution to the RHS $\boldsymbol{b}$. For simplicity, it is assumed that this is not the case here.

the computational time. This shows that the number of coupling iterations is not a suitable measure for assessing the computational efficiency. Instead, a new measure is introduced in Spenke and Delaissé et al. [4], which computes a weighted sum of the total number of coupling, flow solver and structure solver iterations.

Instead of assuming a constant cost associated to each subsolver execution, the new measure assumes a fixed cost for each internal solver iteration:

$$\text{Cost of a flow solver call } C^f \approx C^f_{\text{fix}} + n^f \cdot C^f_{\text{iter}} \tag{3}$$

$$\text{Cost of a structure solver call } C^s \approx C^s_{\text{fix}} + n^s \cdot C^s_{\text{iter}} \ , \tag{4}$$

where $C^{f,s}$ are the cost of a solver call in seconds, $C^{f,s}_{\text{fix}}$ the constant cost associated to a solver call, e.g., communication or mesh motion for the flow solver, and $C^{f,s}_{\text{iter}}$ the constant cost of one subproblem iteration. These cost factors are considered fixed for a particular run, but the cost of a single solver call can vary, since the number of subproblem iterations $n^{f,s}$ can differ from one solver execution to another. Therefore, the cost associated to coupling iteration $\bar{k}$ equals

$$C^p(\bar{k}) \approx C^p_{\text{fix}} + n^p(\bar{k}) \cdot C^p_{\text{iter}} \quad \text{for} \quad p = f, s \ . \tag{5}$$

$\bar{k}$ counts the coupling iterations over all time steps, going from 1 to N^c, the total number of coupling iterations.

By expressing the time associated to each coupling iteration as C^c, which includes for example data exchange, the total simulation time $t_{\text{simulation}}$ can be approximated by $C_{\text{simulation}}$

$$t_{\text{simulation}} \approx C_{\text{simulation}} = \sum_{\bar{k}=1}^{N^c} \left[C^c + \sum_{p=f,s} C^p(\bar{k}) \right] = N^c \cdot C^c + \sum_{\bar{k}=1}^{N^c} \sum_{p=f,s} C^p(\bar{k}) \ . \tag{6}$$

After inserting Eq. (5) in Eq. (6) and rearranging the terms, the new cost measure is obtained:

$$C_{\text{simulation}} = N^c \cdot \overline{C}^c + N^f \cdot C^f_{\text{iter}} + N^s \cdot C^s_{\text{iter}} \tag{7}$$

Here, $\overline{C}^c$ equals the sum of the fixed contribution of the coupling, flow and structure part, i.e., $\overline{C}^c = C^c + C^f_{\text{fix}} + C^s_{\text{fix}}$, and $N^{f,s}$ are the total number of flow and structure iterations, respectively.

So in short, the new equivalent time measure $C_{\text{simulation}}$ represents the cost as weighted sum of the total number of coupling, flow and structure iterations. The weighting factors are cost factors and are generally not known. Their value depends on the simulated problem, the solution techniques of the subsolvers and the coupling algorithm, but also on for example computer architecture. In this work, the cost factors are determined by performing regression over all simulations of the parameter study, which is discussed in detail in Section 5. This approach will allow to gain insights in the trends that occur when limiting the number of subproblem iterations per solver call. Additionally, it helps to avoid the effects of the *random* variation in run time on the results due to, e.g., load balancing or loading and clock speeds of the processors. For this reason the equivalent time will be used in Section 5.

To determine the cost factors of the flow solver, linear regression with two independent variables is performed, where the independent variables are N^f and N^c while the dependent value is the total time spent in the flow solver, i.e., $\sum_{\bar{k}=1}^{N^c} C^f(\bar{k}) = N^c \cdot C^f_{\text{fix}} + N^f \cdot C^f_{\text{iter}}$. An analogous approach is followed for the structure solver. For C^c, a linear regression is performed with one independent variable N^c and as dependent value the part of the run time not spent on the flow and structure solver.

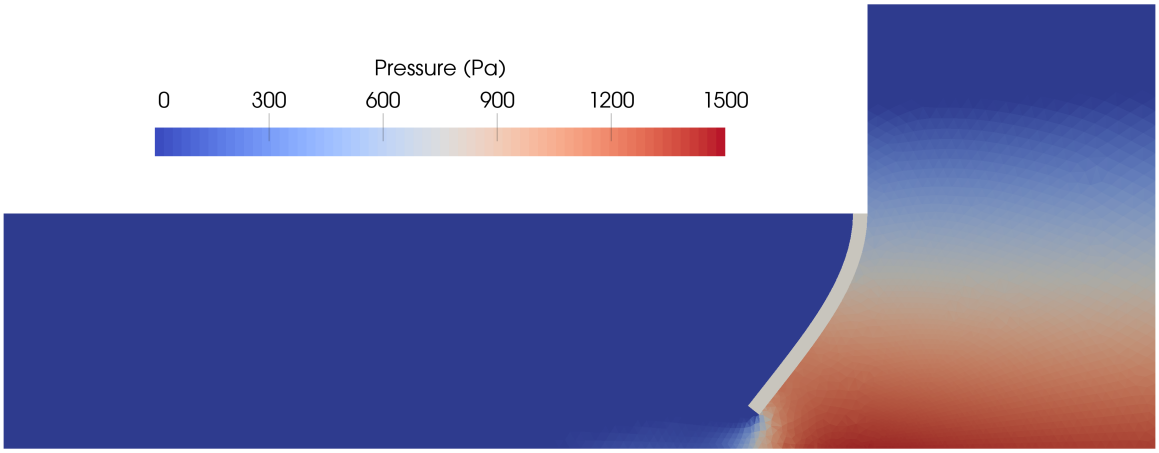


Figure 1: Deformation of the flexible gate and pressure field of the air and water at $t = 0.1\,\mathrm{s}$.

4 TEST CASE

The breaking dam simulation [5] serves as test case. It consists of a rubber gate, fixed at the top, behind which a water column is located. At the start of the simulation, this water column is released and, due to the effect of gravity, the water starts to flow underneath the flexible gate, pushing it away, as illustrated by Fig. 1. The geometry and boundary conditions are sketched in Fig. 2. The water density and dynamic viscosity are $1000\,\mathrm{kg/m^3}$ and $0.001\,\mathrm{Pa\,s}$, the values for air are $1.225\,\mathrm{kg/m^3}$ and $1.7894 \times 10^{-5}\,\mathrm{Pa\,s}$ and the rubber gate has an elasticity modulus of $1 \times 10^7\,\mathrm{N/m^2}$, a Poisson ratio of 0.4 and a density of $1100\,\mathrm{kg/m^3}$. The simulated time is typically 0.4 s, divided in 400 time steps of 0.001 s. To reduce the computational effort of the parameter study in the next section, only 50 time steps are calculated.

The flow domain is simulated using the finite-volume (FV) method with ANSYS Fluent [6] and its deformation is included using the arbitrary Lagrangian-Eulerian (ALE) frame of reference. The motion of the triangular mesh is obtained by combining spring-based smoothing with remeshing and the free surface is tracked with the volume of fluid (VOF) method. Furthermore, the flexible gate is modeled with finite elements (FE) in the Structural Mechanics Application of the Kratos Multiphysiscs code [7]. A hyperelastic neo-Hookean plane strain material model captures the flexible behavior of the rubber. Finally, the in-house, open source code CoCoNuT [3] achieves the FSI coupling and can be found in the GitHub repository pyfsi/coconut. Radial basis interpolation [8] realizes the data exchange on the non-matching interface.

The flow and structure domain have 7276 triangular cells (initially) and 273 total Lagrangian quadrilateral first-order elements, respectively. The flow solver is considered fully converged when the scaled continuity and velocity residuals are lower than 1×10^{-5}. For the structural solver this is the case when the norm of the residual divided by the number of degrees of freedom (DOF) becomes smaller than 1×10^{-6}. The used coupling algorithm is IQN-ILS [9] without reuse ($q = 0$) and without filtering.

Although obtaining a physically accurate solution is not essential for the research question of this work, Fig. 3 compares the displacement of the tip of the gate, centrally over the thickness, to those found in literature. The observed deviations with the experimental and numerical results in literature stem from the material model used for the structure. While both Antoci

Parameter	Value	Parameter	Value
L	0.079 m	A	0.1 m
S	0.005 m	H	0.14 m
G	0.0025 m		

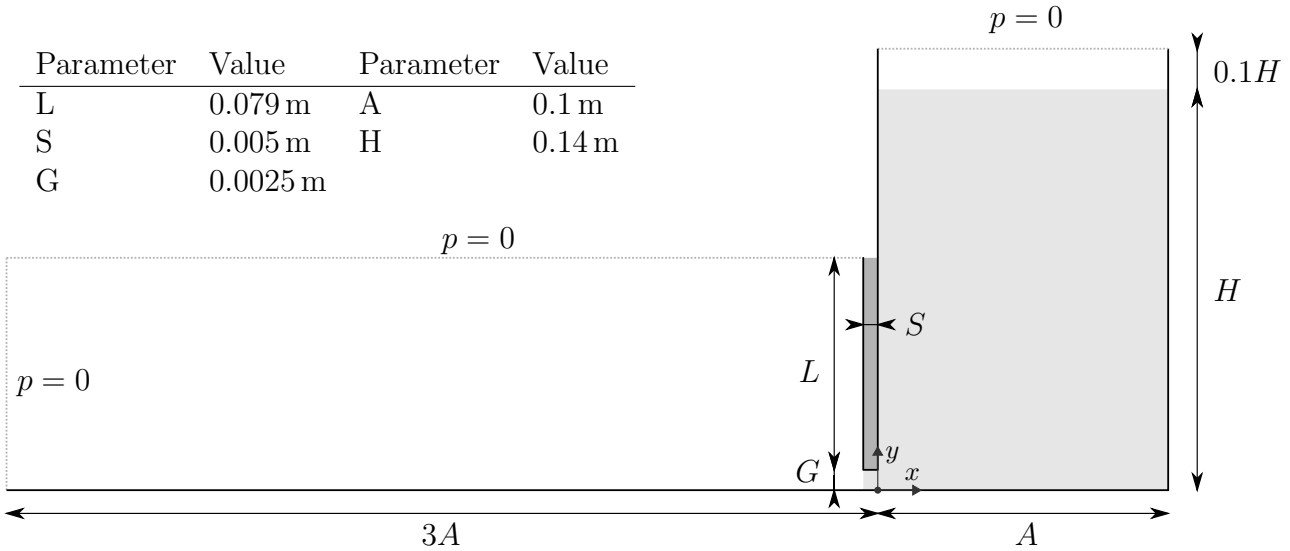


Figure 2: Sketch of the geometry and boundary conditions. The initial gap has been exaggerated for clarity.

et al. [5] and Yang et al. [10] employ the non-linear Mooney-Rivlin material model, fitted to the stress-strain curve of the material used in the experiments, this work resorts to a neo-Hookean model, since the Mooney-Rivlin model is not available within Kratos Multiphysiscs. The proposed explanation is supported by the agreement with the results obtained with a linear material model by Yang et al.

5 RESULTS

The goal of this work is to investigate to what extent the computational effort of a partitioned FSI simulation can be reduced, by limiting the number of subproblem iterations in a solver call. With that goal in mind, a parameter study is performed in which both the number of flow and structure subsolver iterations per solver call are limited to n^f_{max} and n^s_{max}, respectively.

The breaking dam test case for which every solver called is converged fully never requires more than 300 fixed-point iterations for the FV flow solver and not more than 3 Newton iterations for the FE structure solver. Therefore, n^f_{max} is varied from 300 to 5 subproblem iterations[3] and n^s_{max} from 3 to 1. With the methodology from Section 3, the numerical values for the cost factors in Table 1 are obtained. This approach assumes constant subproblem iteration cost, which admittedly is not exact, but it is still a better approximation than constant coupling iteration cost, as supported by Fig. 4, which compares both. Before discussing the optimal choice of the maximal number of iteration in each solver call, the variation of the different total number of coupling (N^c), flow subproblem (N^f) and structure subproblem (N^s) iterations is presented.

Figure 5a shows how the number of coupling iterations changes for the different runs. As expected, the minimum is reached for fully converging both the flow and structure solver in every call, since in that case, the maximal accuracy of the intermediate solutions is reached

[3]Since only a few flow solver calls require more than 150 subproblem iterations, the variation for $n^f_{max} > 150$ is small and left out of the following figures for clarity. The full parameter study, including these omitted runs, is presented in Table 2.

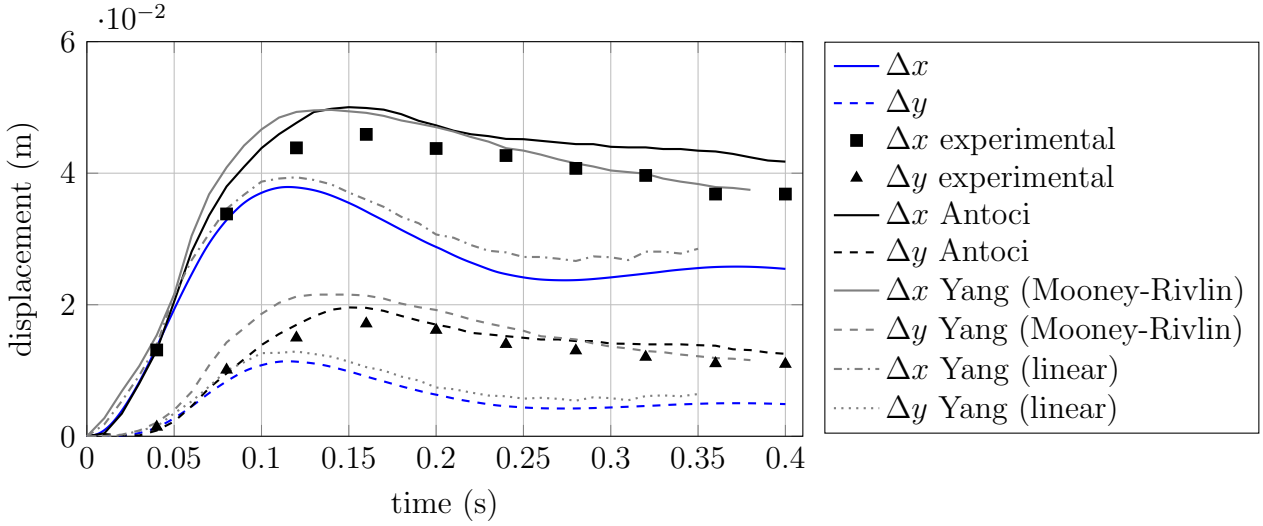


Figure 3: Horizontal (Δx) and vertical (Δy) displacement of the tip of the flexible gate in function of time. The blue lines correspond to the results of this work. The black data are determined by Antoci et al. [5], either experimentally (marked points) or through simulation. The gray data are simulation results obtained by Yang et al. [10] with a linear and non-linear material model.

Table 1: Cost factors in seconds per corresponding iteration.

C^f_{fix}	C^f_{iter}	C^s_{fix}	C^s_{iter}	C^c	$\overline{C}^c$
1.0803	0.0847	0.0497	0.0133	0.0147	1.1447

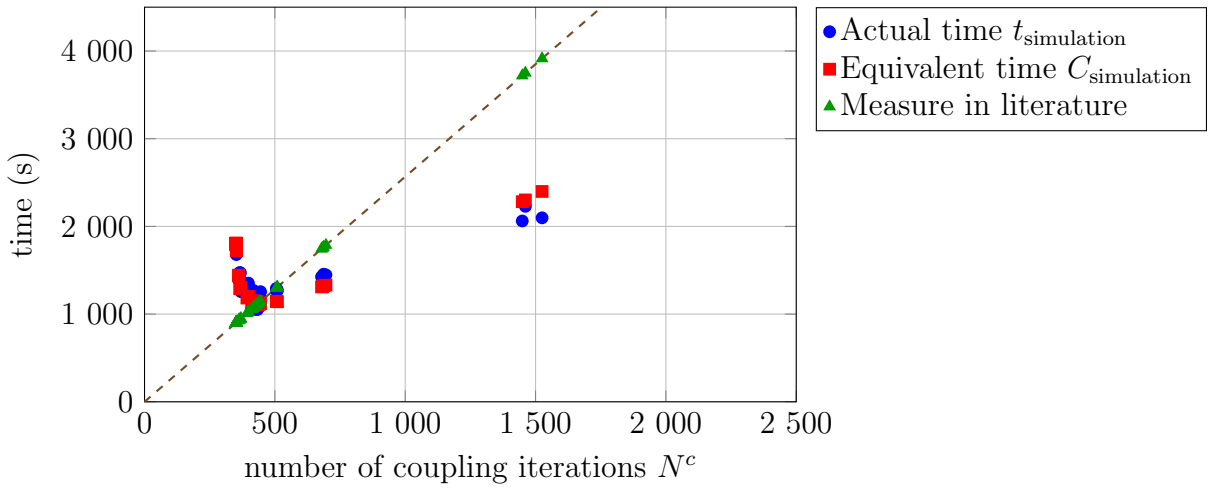
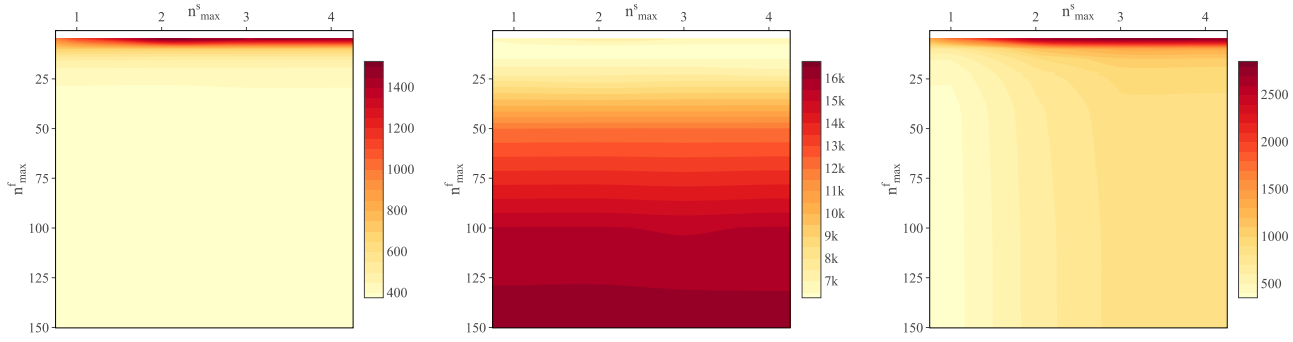


Figure 4: Difference between the actual time, the equivalent time used here, and the equivalent time considering a constant cost per coupling iteration. Note that the equivalent time with constant cost per coupling iteration forms a line through the origin.



(a) Coupling iterations N^c. (b) Flow subproblem iterations N^f. (c) Structure subproblem iterations N^s.

Figure 5: Different iteration counts plotted over n^f_{max} and n^s_{max}.

before coupling back, minimizing the number of coupling steps. Restricting the number of flow subproblem iterations leads to an increase, which becomes significant from $n^f_{max} \approx 25$. The reason is that as less accurate flow solutions are transferred to the structure solver, the exchange needs to happen more frequently. While in general, this is also the case for limiting the structure subproblem iterations, no such trend is obvious here. For $n^f_{max} \lessapprox 5$ the simulation does not converge as a consequence of a too inaccurate flow solution, which when used as boundary condition in the structure solver, also prevents the latter to reach an appropriate solution. While this is true in general, it is especially important when employing quasi-Newton coupling techniques, since they base the construction of the approximate Jacobian on differences between iterations.

Figures 5b and 5c present how the total number of flow and structure iterations vary. Opposite to the coupling iterations trend, fewer subproblem iterations per solver call lead to fewer overall subproblem iterations. This indicates that the decrease in subproblem iterations per solver call outweighs the increase in coupling iterations and that exchanging data before they are fully polished can indeed reduce the time spent in the subsolvers. For the structure subproblem iterations, a secondary trend is visible: decreasing n^f_{max} leads to an augmentation in number of structure subproblem iterations, which occurs together with the increase in coupling iterations shown in Fig. 5a.

Now that the trends in iterations count have been clarified, the optimal choice of n^f_{max} and n^s_{max} follows from a weighted sum, as explained in Section 3. The weighting factors are the cost factors, expressed in seconds, given in Table 1. The result of this weighted sum is shown in Fig. 6, wherein, for clarity, the computational time has been normalized by the value obtained for full convergence. For $n^f_{max} \approx 21$ and $n^s_{max} \approx 1$, an optimum is reached, reducing the calculation time to only 61% of the time required for the full convergence case. Also note that this optimum is rather flat.

As explained, the addition of the plots in Fig. 5 is determined by the cost factors in Table 1. This table shows that C^f_{fix} has by far the greatest value, which is a consequence of the costly remeshing operation at the start of every flow solver call. Of all costs occurring every coupling iteration, it constitutes the biggest part of $\overline{C}^c$. Nevertheless, Fig. 6 most closely resembles Fig. 5b, which shows the number of flow subproblem iterations. This is a consequence of the

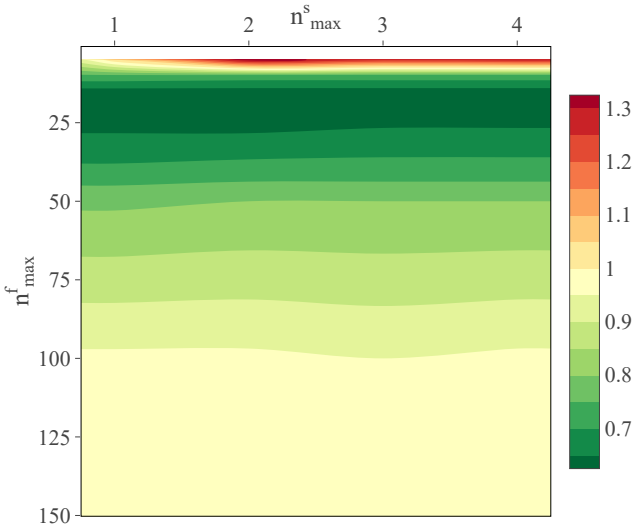


Figure 6: Contour plot of the equivalent time measure, normalized with respect to the value obtained for full convergence.

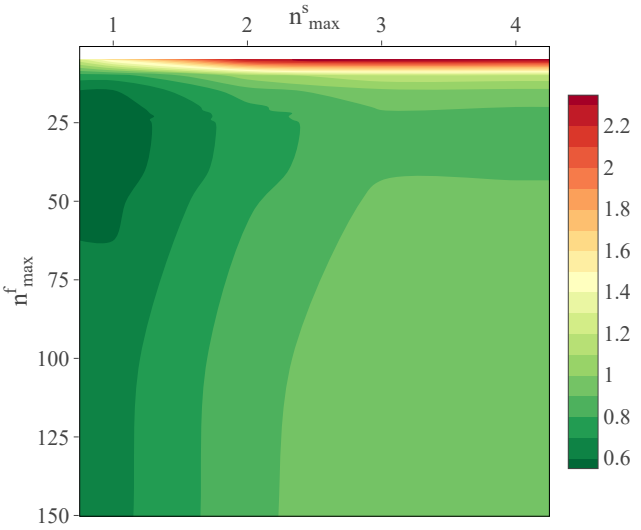


Figure 7: Contour plot of the equivalent time measure with the fictional cost factors $\overline{C}^c = 1\,\mathrm{s}$, $C^f_{\mathrm{iter}} = 0.1\,\mathrm{s}$ and $C^s_{\mathrm{iter}} = 3\,\mathrm{s}$, normalized with respect to the value obtained for full convergence.

large variation in N^f between different runs, as well as its high value, which is about an order of magnitude bigger than N^c and N^s. Comparing N^f and C^f_{iter} to N^s and C^s_{iter} shows that structure solver is much cheaper and therefore has limited effect on the total computational cost. The increase in N^c around $n^f_{max} \approx 5$, on the other hand, is significant and visible in the contour plot of the equivalent time.

Since the cost factors are determined by, among others, the specific problem, the used solvers and the computer hardware, the dominance of the flow cost is not a general conclusion. For example, the cost of the flow solver could be reduced by doubling the number of cores it uses or the structure problem could be refined leading to an increase in structure solver cost. These actions would lead to new cost factors which in turn result in a new optimal choice of n^f_{max} and n^s_{max}. To illustrate this, the equivalent time shown in Fig. 7 has been calculated with the fictional cost factors $\overline{C}^c = 1\,\mathrm{s}$, $C^f_{\mathrm{iter}} = 0.1\,\mathrm{s}$ and $C^s_{\mathrm{iter}} = 3\,\mathrm{s}$. Compared with Fig. 6, the region of optimal performance is situated differently. Likewise, if for example $\overline{C}^c$ would be the dominant contribution, the equivalent time plot would resemble Fig. 5a and the optimum would be found at the fully converged case.

While, up to now, the focus was each time on a single aspect of the parameter study, the complete results are given in Table 2.

6 CONCLUSIONS

This work investigates the effect of limiting the number of subproblem iterations in the flow and structure solver of a partitioned FSI simulation. By means of the breaking dam test case, the effect is systematically evaluated in a parameter study, and, through the use of a new convergence criterion, it is assured that the different runs have the same accuracy.

While minimization of the number of coupling iterations is the typical performance criterion

Table 2: Breaking dam with IQN-ILS (q=0). The row and column header contain the maximal number of subproblem iterations for the flow and solid solver, n^f_{max} and n^s_{max}, respectively. For each case, the **equivalent time** is given, as well as the number of *coupling iterations*, flow solver iterations and solid solver iterations. A missing value indicates that the coupling did not converge.

Fixed-point iterations per coupling iteration - Flow solver	Newton iterations per coupling iteration - Structural solver: 1		2		3		∞	
5	**-**	-	**1.33**	*1525*	**1.26**	*1449*	**1.28**	*1461*
	-	-	7289	2665	6912	2847	6993	2795
10	**0.74**	*696*	**0.73**	*683*	**0.73**	*687*	**0.73**	*679*
	6181	696	6092	1148	6141	1340	6077	1337
15	**0.63**	*510*	**0.63**	*507*	**0.63**	*506*	**0.63**	*506*
	6495	510	6495	889	6494	1080	6492	1080
20	**0.62**	*445*	**0.62**	*443*	**0.62**	*443*	**0.61**	*440*
	7143	445	7137	793	7137	982	7004	980
21	**0.61**	*433*	**0.62**	*436*	**0.62**	*436*	**0.62**	*433*
	7126	433	7265	781	7265	969	7128	972
22	**0.61**	*426*	**0.62**	*427*	**0.62**	*427*	**0.62**	*427*
	7226	426	7246	756	7246	948	7246	948
23	**0.62**	*419*	**0.62**	*426*	**0.63**	*430*	**0.62**	*425*
	7382	419	7418	756	7558	950	7416	944
24	**0.63**	*417*	**0.64**	*424*	**0.63**	*418*	**0.63**	*418*
	7675	417	7703	750	7536	933	7536	933
25	**0.63**	*415*	**0.63**	*411*	**0.64**	*415*	**0.64**	*415*
	7827	415	7656	734	7816	930	7816	930
30	**0.66**	*394*	**0.66**	*393*	**0.67**	*398*	**0.67**	*398*
	8654	394	8544	710	8665	907	8665	907
40	**0.71**	*365*	**0.72**	*368*	**0.72**	*368*	**0.72**	*370*
	10227	365	10245	681	10245	873	10269	874
50	**0.79**	*361*	**0.80**	*362*	**0.80**	*362*	**0.80**	*362*
	11994	361	12012	673	12011	865	12011	865
100	**0.96**	*353*	**0.96**	*353*	**0.95**	*352*	**0.96**	*353*
	15535	353	15544	655	15432	845	15544	847
150	**0.99**	*352*	**1.00**	*352*	**1.00**	*352*	**0.99**	*351*
	16341	352	16348	652	16348	844	16264	843
200	**1.00**	*352*	**1.00**	*351*	**1.00**	*351*	**1.00**	*351*
	16481	352	16391	651	16391	843	16391	843
300	**1.00**	*351*	**1.00**	*351*	**1.00**	*351*	**1.00**	*350*
	16495	351	16499	652	16499	844	16449	842
∞	**1.00**	*350*	**1.00**	*350*	**1.00**	*350*	**1.00**	*350*
	16448	350	16449	650	16449	842	16449	842

in FSI, this work shows that the number of subproblem iterations is equally important. Since minimization of the number of coupling and subproblem iterations are typically conflicting goals, the fastest simulation is only obtained by balancing the two.

To study these effects, a new performance measure has been introduced which describes the computational efficiency of a simulation as a weighted sum of the iteration counts. In this equivalent time, the weighting factors are so-called cost factors that are determined by, among others, the specific problem, solving strategy and computing infrastructure.

Besides allowing to study the trends, the parameter study revealed that a reduction in computational time of 39% could be obtained. In practice, it is of course unpractical to run a parameter study in order to determine the optimum for every new case. In that regard, dynamically adapting the subproblem iteration limits based on, e.g., the coupling residual, presents itself as a viable alternative and even has the potential to further shorten the run time.

REFERENCES

[1] P. Causin, J.-F. Gerbeau, and F. Nobile, "Added-mass effect in the design of partitioned algorithms for fluid-structure problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 42–44, pp. 4506–4527, 2005.

[2] E. H. van Brummelen, "Added mass effects of compressible and incompressible flows in fluid-structure interaction," *Journal of Applied Mechanics*, vol. 76, no. 2, p. 021 206, 2009.

[3] N. Delaissé, T. Demeester, R. Haelterman, and J. Degroote, "Quasi-Newton methods for partitioned simulation of fluid-structure interaction reviewed in the generalized Broyden framework," *Archives of Computational Methods in Engineering*, 2023.

[4] T. Spenke, N. Delaissé, J. Degroote, and N. Hosters, "On the number of subproblem iterations per coupling step in partitioned fluid-structure interaction simulations," *arXiv*, 2023.

[5] C. Antoci, M. Gallati, and S. Sibilla, "Numerical simulation of fluid-structure interaction by SPH," *Computers & Structures*, vol. 85, no. 11–14, pp. 879–890, 2007.

[6] *ANSYS Fluent theory guide, 2019R3*, ANSYS, Inc., Aug. 2019.

[7] V. M. Ferrándiz, P. Bucher, R. Zorrilla, *et al.*, *Kratosmultiphysics/kratos: Release 9.2*, 2022.

[8] M. Lombardi, N. Parolini, and A. Quarteroni, "Radial basis functions for inter-grid interpolation and mesh motion in FSI problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 256, pp. 117–131, 2013.

[9] J. Degroote, K.-J. Bathe, and J. Vierendeels, "Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction," *Computers & Structures*, vol. 87, no. 11–12, pp. 793–801, 2009.

[10] Q. Yang, V. Jones, and L. McCue, "Free-surface flow interactions with deformable structures using an SPH-FEM model," *Ocean Engineering*, vol. 55, pp. 136–147, 2012.