IV International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2011
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

COARSE LEVEL NEWTON-KRYLOV ACCELERATION OF SUB-ITERATIONS IN PARTITIONED FLUID-STRUCTURE INTERACTION

Alexander H. van Zuijlen* **and Hester Bijl†**

*,†Faculty of Aerospace Engineering
Delft University of Technology
P.O. Box 5058, 2600 GB Delft, The Netherlands
e-mail: A.H.vanZuijlen@tudelft.nl

Key words: High order time integration, fluid-structure interaction, partitioned integration

Abstract. Computational fluid-structure interaction is commonly performed using a partitioned approach. For strongly coupled problems sub-iterations are required, increasing computational time as flow and structure have to be resolved multiple times every time step. Reductions in computing times can be achieved by e.g. improving the convergence of the sub-iteration technique and/or performing sub-iterations on a coarse level, but also by improving the iterative solver used in the flow solver.

In this paper we investigate the combination of a multilevel acceleration technique for sub-iterations which employs a Newton-Krylov solver on the coarse level to obtain high convergence for the correction term and a multigrid solver which performs only a limited amount of iterations on the fine level to reduce memory and computing requirements. For switching between a coarse grid correction and fine grid solve, an automated coarse grid ACG(r) selection algorithm is proposed. The algorithm is applied to an academic, two dimensional test case with incompressible flow. Compared to sub-iterating with a (memory intensive) JFNK algorithm on the fine mesh, the hybrid algorithm already requires 10% less computing time. When compared to sub-iterating with multigrid, the performance increase for the hybrid scheme is a factor 3.

1 INTRODUCTION

Fluid-structure interaction in the incompressible flow regime is often encountered in engineering problems, e.g. aeroelasticity of wind turbine blades, flapping wing flight of micro aerial vehicles or underwater propulsion using deformable bodies. Especially in the field of bio inspired propulsion/control, one encounters strongly coupled fluid-structure interaction in incompressible media. These interactions are challenging to simulate in

a partitioned procedure, which is a common way of simulating complex fluid-structure interaction problems as it allows the usage of sophisticated mono-disciplinary solvers. Every time step sub-iterations are required between flow and structure solver to resolve the fully coupled solution, reducing computational efficiency.

A robust and effective sub-iteration technique with incompressible flow domains is by no means a trivial task as the coupling gets stronger as the time step is reduced [1] and straight forward sub-iterations such as block Gauss-Seidel iterations diverge. More sophisticated algorithms such as the Aitken [2] and interface quasi-Newton [3] method are required. Still, a number of sub-iterations need to be performed at every time step.

To reduce the computational work for the sub-iterations, we previously proposed a multilevel algorithm [4], which resolves a defect-correction on a coarse fluid mesh for the initial sub-iterations when still a large part of the partitioning error has a large wave number component. This part of the partitioning error can be effectively reduced on a coarse fluid mesh at a low computational expense compared to solving the fluid equations on the fine mesh. The algorithm was successfully applied for compressible flow using a density based finite volume code with nonlinear multigrid. For these cases, the flow solution of the previous sub-iteration is continuously closer to the fully coupled solutions as the sub-iterations converge. Therefore the flow does not have to be resolved to a very high precision at every sub-iteration, which limits the computational cost per sub-iteration. However, when using the same density based nonlinear multigrid solver with incompressible flow, the number of multigrid iterations required per sub-iteration was much higher [5] for the accurate representation of the pressure field, increasing computational cost per sub-iteration.

In order to overcome the poor convergence of the nonlinear multigrid solver for incompressible flow problems using a density based finite volume code, preconditioning of the equations can be applied [6]. Another approach is to use a different iterative solver, e.g. a Jacobian Free Newton Krylov (JFNK) solver [7]. This solver has the advantage that flow field (pressure field) is solved globally and the method therefore does not experience the same reduction in convergence rate as the nonlinear multigrid solver for incompressible flow. On the other hand JFNK has the disadvantage that a preconditioner is required, e.g. an ILU (incomplete LU decomposition), for fast convergence and that this preconditioner is expensive both in terms of memory and computing time. For unsteady flow the preconditioner can be reused for a number of time steps so that computational costs are reduced, but for sub-iterations in a fluid-structure interaction setting, the intermediate solutions during the sub-iterations may differ too much to use the same preconditioner especially during the initial sub-iterations [5].

In this paper we propose a hybrid algorithm which uses a multilevel acceleration for the sub-iterations, using a fixed number of nonlinear multigrid iterations on the finest mesh and a JFNK algorithm for the coarse grid correction computation. By choosing a fixed number of multigrid iterations on the finest level, the computational costs of a fine level sub-iteration is reduced at the expense of a lower convergence. This way both

a high level of convergence can be reached on the coarse level for limited computational effort and the memory requirements stay within limits as the ILU preconditioner is only required for the coarse mesh.

The question is, however, whether the relaxation of the convergence for the fine level sub-iterations, reduces the effectiveness of sub-iterating to such an extent that the computational speed gained by not iterating the solution to a high convergence on the fine mesh level is eliminated by the increase in sub-iterations to converge to the fully coupled solution. To answer this question, the hybrid algorithm is applied to an incompressible, two-dimensional test case [8] and its efficiency compared to only sub-iterating on the fine level and multilevel acceleration without JFNK.

First the two dimensional incompressible test case is given in Section 2. Next the hybrid multilevel algorithm is introduced in Section 3, where after the algorithm is used to solve the coupled problem in Section 4. Finally conclusions and recommendations are given in Section 5.

2 COUPLED PROBLEM

In this paper a fluid-structure interaction problem is addressed which is based on the benchmark problem by Turek [8]. The original problem consists of an incompressible fluid around a circular cylinder with a flexible trailing flap. The coupled problem consists of a fluid domain Ω_f, see Fig. 1, which is modeled as an incompressible fluid, and a structure

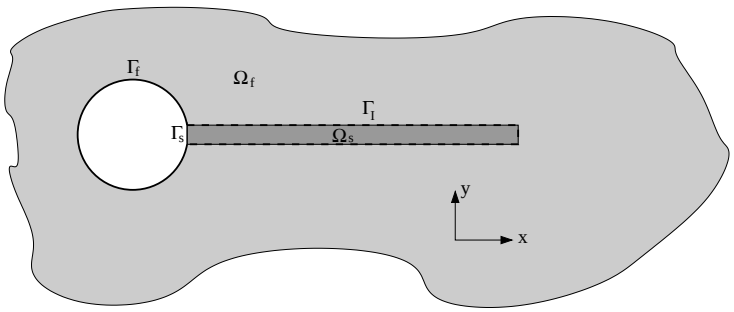

Figure 1: Computational domain.

domain Ω_s which is here modeled as a linear elastic body. The boundaries of the domain are given by Γ_s, Γ_f, which result in boundary conditions for the structure and fluid dynamics respectively. In the partitioned approach the interface boundary between the fluid and the structure domains Γ_I is denoted by two boundaries Γ_{sf} and Γ_{fs} which close the structure and fluid domains so that each domain can be treated separately from the other. The coupling between flow and structure is introduced in the boundary conditions that are imposed on Γ_{sf} and Γ_{fs} and that should yield continuity of displacement (of the interface) and stresses. Since the flap has the highest flexibility in y-direction and the shear stresses mainly act in x-direction, we simplify the continuity of stresses to a

continuity in pressure so that the conditions at the interface are

$$d_{\Gamma_{fs}} = d_{\Gamma_{sf}}, \tag{1}$$
$$p_{\Gamma_{sf}} = p_{\Gamma_{fs}}, \tag{2}$$

where d denotes the displacement of the interface boundary and p the pressure. At the moment it is still assumed that the spatial coupling is continuous and the temporal coupling instantaneous. Since the domains have been split and the coupling is performed using boundary conditions, readily available flow and structure solvers can be used to discretize and resolve their own dynamics on their own domains. Therefore, we do not address the specific spatial discretization of the solvers and simply write

$$\frac{\mathrm{d}\mathbf{w}_s}{\mathrm{d}t} + \mathbf{D}_s(\mathbf{w}_s, \mathbf{p}_{\Gamma_{sf}}) = \mathbf{S}_s, \tag{3}$$
$$\frac{\mathrm{d}\mathbf{w}_f}{\mathrm{d}t} + \mathbf{D}_f(\mathbf{w}_f, \mathbf{d}_{\Gamma_{fs}}) = \mathbf{S}_f, \tag{4}$$

wherein $\mathbf{w}_s$ and $\mathbf{w}_f$ are the discrete state vectors for the structure state and fluid state respectively. They contain e.g. the structural displacement or the fluid density. The spatial discretization of the governing equation is simplified by the operator $\mathbf{D}$, which depends both on the state $\mathbf{w}$ and on the fluid-structure interface conditions $\mathbf{p}_{\Gamma_{sf}}$ (the discrete pressure acting on the structure) and $\mathbf{d}_{\Gamma_{fs}}$ (the displacement of the fluid domain boundary). The right hand side may contain terms $\mathbf{S}$ that may arise from boundary conditions on Γ_s and Γ_f. Equations (3) and (4) are in semi-discrete form. We assume that the time integration is performed by the same implicit scheme in both domains. The structure and flow solver programs can then be described as solution techniques that can find solutions $\mathbf{w}_s^{n+1}$ and $\mathbf{w}_f^{n+1}$ under the boundary conditions $\mathbf{p}_{\Gamma_{sf}}$ and $\mathbf{d}_{\Gamma_{fs}}$ such that they satisfy (or minimize)

$$\mathbf{r}_s(\mathbf{w}_s^{n+1}, \mathbf{p}_{\Gamma_{sf}}) - \mathbf{s}_s = \mathbf{0}, \tag{5}$$
$$\mathbf{r}_f(\mathbf{w}_f^{n+1}, \mathbf{d}_{\Gamma_{fs}}) - \mathbf{s}_f = \mathbf{0}, \tag{6}$$

wherein $\mathbf{r}$ the residual function (discretized representation of the governing equations), $\mathbf{s}$ a constant source term within the time step that can depend on e.g. previous solutions or boundary conditions, $\mathbf{p}_\Gamma$ the discrete pressures in the boundary nodes and $\mathbf{d}_\Gamma$ the discrete displacements of the boundary nodes. The subscript s, f denotes that the discrete quantities belong to the structure and fluid domains respectively. A Computational Structure Dynamics (CSD) package is capable of finding a $\mathbf{w}_s^{n+1}$ such that (5) is satisfied for a given pressure load $\mathbf{p}_{\Gamma_{sf}}$. A Computational Fluid Dynamics (CFD) package is able to find a $\mathbf{w}_f^{n+1}$ such that (6) is satisfied for a given boundary displacement $\mathbf{d}_{\Gamma_{fs}}$. A fully implicit (or fully coupled) solution would require

$$\mathbf{r}_s(\mathbf{w}_s^{n+1}, \mathbf{p}_{\Gamma_{sf}}^{n+1}) - \mathbf{s}_s = \mathbf{0}, \tag{7}$$
$$\mathbf{r}_f(\mathbf{w}_f^{n+1}, \mathbf{d}_{\Gamma_{fs}}^{n+1}) - \mathbf{s}_f = \mathbf{0}, \tag{8}$$

wherein the superscript $^{n+1}$ denotes the discrete solution at the new time level t_{n+1}. The coupling between (7) and (8) now poses a problem in a partitioned approach as the pressure acting on the structure interface $\mathbf{p}_{\Gamma_{sf}}^{n+1}$ depends on the fluid state $\mathbf{w}_f^{n+1}$ and the displacement of the fluid boundary $\mathbf{d}_{\Gamma_{fs}}^{n+1}$ depends on the structure state $\mathbf{w}_s^{n+1}$. Both the spatial coupling (transferring data from the flow to the structure mesh and vice versa) and the temporal coupling (obtaining an implicitly coupled solution) are addressed in the next sections.

2.1 Spatial coupling

The coupling between flow and structure takes place at the fluid-structure boundary. In the continuous case, this boundary Γ_I would be identical for both fluid and structure domains, however, at the discrete level, the boundary Γ_{sf} and Γ_{fs} do not have to be matching and gaps or overlaps may occur. In this paper a radial basis function interpolation is used [9], both for the interpolation of displacements from structure to fluid mesh and for the interpolation of surface pressure from fluid to structure mesh. The interpolations $\mathcal{I}_{fs}$ and $\mathcal{I}_{sf}$ respectively denote the interpolation from structure to flow and vice versa.

2.2 Temporal coupling

In partitioned fluid-structure interaction, obtaining the coupled solution described by (7) and (8) would require sub-iterating, e.g. when a sequential algorithm is used

$$\mathbf{r}_s(\mathbf{w}_s^i, \hat{\mathbf{p}}_{\Gamma_{sf}}^i) - \mathbf{s}_s = \mathbf{0}, \tag{9}$$

$$\mathbf{r}_f(\mathbf{w}_f^i, \mathbf{d}_{\Gamma_{fs}}^i) - \mathbf{s}_f = \mathbf{0}, \tag{10}$$

wherein the superscript i denotes the i-th sub-iteration and $\hat{\mathbf{p}}_{\Gamma_{sf}}^i$ is the *estimation* of the fluid pressure acting on the structure for the i-th sub-iteration and $\mathbf{d}_{\Gamma_{fs}}^i = \mathcal{I}_{fs}(\mathbf{d}_{\Gamma_{sf}}^i)$ is the fluid boundary displacement obtained by interpolation of the structure boundary displacement. The simplest choice for the pressure estimation is

$$\hat{\mathbf{p}}_{\Gamma_{sf}}^i = \mathcal{I}_{sf}(\mathbf{p}_{\Gamma_{fs}}^{i-1}), \tag{11}$$

which results in a block-Gauss-Seidel type of iteration, but which is not guaranteed to be stable. To increase robustness under-relaxation can be applied, but generally at the expense of slower convergence rate. In this paper we focus on the widely applied Aitken method [2], which applies an adaptive under-relaxation to the estimation for the next time step

$$\hat{\mathbf{p}}_{\Gamma_{sf}}^{i+1} = \hat{\mathbf{p}}_{\Gamma_{sf}}^i + \theta^{i+1}(\mathbf{p}_{\Gamma_{sf}}^i - \hat{\mathbf{p}}_{\Gamma_{sf}}^i), \tag{12}$$

for which the under-relaxation parameter θ^{i+1} is obtained from

$$\theta^{i+1} = \theta^i \left(1 - \frac{(\Delta\mathbf{e}^i)^T(\mathbf{e}^i)}{(\Delta\mathbf{e}^i)^T(\Delta\mathbf{e}^i)}\right), \tag{13}$$

with $\mathbf{e}^i = \mathbf{p}^i_{\Gamma_{sf}} - \hat{\mathbf{p}}^i_{\Gamma_{sf}}$ the error between the estimated and the resulting pressure after solving (9) and (10) for iteration i and $\Delta\mathbf{e}^i = \mathbf{e}^i - \mathbf{e}^{i-1}$. For the first under-relaxation step a θ has to be chosen as $\mathbf{e}^{i-1}$ is not available yet. One can use last known value and at the very start of the computation any (sufficiently small) value can be taken.

3 HYBRID MULTILEVEL ACCELERATION

In order to reduce the computational time for updating the fluid solution during sub-iterations, the multilevel acceleration scheme solves for a correction of the fluid solution on a coarse mesh. In the following we introduce the subscripts h and H to denote that variables or operators are defined on the fine or coarse fluid mesh respectively. In order to obtain the coarse level correction, it is first assumed that a fluid solution $\mathbf{w}^{i-1}_{h,f}$ is known (e.g. from the previous sub-iteration or previous time step). When this solution is substituted in (10) a residual is identified

$$\mathbf{r}_{h,f}(\mathbf{w}^{i-1}_{h,f}, \mathbf{d}^i_{h,\Gamma_{fs}}) - \mathbf{s}_{h,f} = \hat{\mathbf{r}}^i_{h,f}, \tag{14}$$

wherein $\hat{\mathbf{r}}_{h,f}$ denotes the residual in the fluid domain on the fine fluid mesh h. Subtraction of (14) from (10) gives a relation for the change in the solution $\Delta\mathbf{w}^i_{h,f} = \mathbf{w}^i_{h,f} - \mathbf{w}^{i-1}_{h,f}$ that is desired for the current sub-iteration

$$\mathbf{r}_{h,f}(\mathbf{w}^{i-1}_{h,f} + \Delta\mathbf{w}^i_{h,f}, \mathbf{d}^i_{h,\Gamma_{fs}}) - \mathbf{r}_{h,f}(\mathbf{w}^{i-1}_{h,f}, \mathbf{d}^i_{h,\Gamma_{fs}}) = -\hat{\mathbf{r}}^i_{h,f}. \tag{15}$$

Solving the correction $\Delta\mathbf{w}^i_{h,f}$ with an iterative solution technique is as expensive as iteratively solving (10) directly for $\mathbf{w}^i_{h,f}$. Therefore (15) is approximated on a coarse grid, reducing computational costs. It should be noted that (15) does not include the source terms $\mathbf{s}_{h,f}$ derived from boundary conditions and that the smaller the right-hand-side term (in general) the smaller the correction term that is computed, which is fundamentally different from solving (10) on a coarse mesh.

3.1 Coarse grid correction

The coarse grid correction is obtained by approximation of (15) on a coarse mesh

$$\mathbf{r}_{H,f}(\mathbf{w}^{i-1}_{H,f} + \Delta\mathbf{w}^i_{H,f}, \dot{\mathbf{x}}^i_{H,f}) - \mathbf{r}_{H,f}(\mathbf{w}^{i-1}_{H,f}, \dot{\mathbf{x}}^i_{H,f}) = -\hat{\mathbf{r}}_{H,f}, \tag{16}$$

wherein, instead of displacement of the boundary, the mesh velocity $\dot{\mathbf{x}}$ caused by the displacement of the mesh is denoted to represent the coupling between structure and flow. The mesh velocity is obtained after the displacement of the mesh by imposing the Discrete Geometric Conservation Law (DGCL) [10]. In (16) we use geometric restriction operators to obtain coarse level variables

$$\mathbf{w}^{i-1}_{H,f} = R_{\mathbf{w}}\mathbf{w}^{i-1}_{h,f}, \tag{17}$$

$$\hat{\mathbf{r}}^i_{H,f} = R_{\mathbf{r}}\hat{\mathbf{r}}^i_{h,f}, \tag{18}$$

wherein $R_{\mathbf{w}}$ and $R_{\mathbf{r}}$ are the restriction operators for the solution and residual respectively. The restriction operators can differ, depending on the variables that are stored in $\mathbf{w}$ and $\hat{\mathbf{r}}$ are already (cell) volume scaled. If the variables are volume scaled, the restriction operator is simply summation, otherwise the restriction computes a volume weighted average on the coarse level. The coarse level mesh velocity follows from imposing the DGCL on the coarse level. When $\Delta\mathbf{w}^i_{H,f}$ is obtained on the coarse level, the solution on the fine level is corrected by prolongation of the correction term to the fine grid

$$\mathbf{w}^{i+1}_{h,f} = \mathbf{w}^i_{h,f} + P_{\mathbf{w}}\Delta\mathbf{w}^i_{H,f}, \tag{19}$$

wherein $P_{\mathbf{w}}$ is the prolongation operator for the fluid state variables $\mathbf{w}$. In this paper a linear interpolation is used as prolongation operator. Since the coarse grid operator $\mathbf{r}_{H,f}$ is a nonlinear operator, obtaining $\Delta\mathbf{w}^i_{H,f}$ requires an iterative solution technique.

3.2 Coarse level iterative solver

In order to reuse the iterative solvers already available for the fine mesh, the coarse grid correction (16) can be cast into the following form

$$\mathbf{r}_{H,f}(\mathbf{w}^i_{H,f}, \dot{\mathbf{x}}^i_{H,f}) + \mathbf{s}_{H,f} + \hat{\mathbf{s}}_{H,s} = \mathbf{0}, \tag{20}$$

which is equal to the origingal problem (10) on a coarse mesh with an additional source term

$$\hat{\mathbf{s}}_{H,s} = \hat{\mathbf{r}}_{H,f} - \mathbf{r}_{H,f}(\mathbf{w}^{i-1}_{H,f}, \dot{\mathbf{x}}^i_{H,f}) - \mathbf{s}_{H,f}. \tag{21}$$

The source term is constant during the coarse level iterations. If we denote $\mathbf{w}^{i,j}_{H,f}$ the j-th iterative approximation of $\mathbf{w}^{i,j}_{H,f}$, then the fluid residual during the coarse level iterations is defined by

$$\epsilon^j_{it} = \mathbf{r}_{H,f}(\mathbf{w}^{i,j}_{H,f}, \dot{\mathbf{x}}^i_{H,f}) + \mathbf{s}_{H,f} + \hat{\mathbf{s}}_{H,s}. \tag{22}$$

For incompressible flow with the current density based finite volume flow solver, an accurate representation of the pressure field (and thus the loads on the structure) requires a high level of iterative convergence (very low value of ϵ_{it}). E.g. when for a compressible medium only 20 multigrid iterations were required to obtain a sufficiently accurate pressure field, in the incompressible case it is roughly ten times larger. As explained in the introduction, either a nonlinear multigrid algorithm is used to minimize ϵ^j_{it} or a Jacobian Free Newton-Krylov algorithm. The JFNK algorithm initially needs more computing time to build the ILU preconditioner. However, once the preconditioner is built, the Newton updates can be efficiently evaluated using a GMRES algorithm and only few Newton updates are required to obtain a high iterative convergence.

3.3 Automated level selection

The steps above describe how a sub-iteration can be performed by computing a coarse grid correction. However, still a criterion is needed to determine whether to perform a sub-iteration on the coarse level or that it should be performed on the fine level. Previously

either a fixed number of coarse level iterations was performed followed by one fine level iteration, or the coupling was first fully resolved using coarse level iterations after which only fine level iterations were performed. The latter option has the drawback that if the coupling is resolved fully on the coarse level, still a partitioning error is present on the fine level that will not be reduced by further sub-iterating on the coarse level. The first option is more efficient, but it had to be determined manually how many coarse level sub-iterations gave the highest efficiency. Therefore a ratio is proposed to automatically determine the mesh level on which the sub-iteration is performed. The idea is that when the residual from the fine fluid mesh is restricted to the coarse mesh, any high wave number mode is filtered out by the averaging (or summation) of neighboring cells. This also means that the norm of the residual on the coarse level is reduced compared to the norm of the residual on the fine level. When it is assumed that the correction that is computed scales approximately with the norm of the residual, it can be estimated that when the residual on the coarse level is much smaller than the residual on the fine level, higher wave number errors remain dominant and are not reduced by the coarse level correction and a fine level solve is desired. Using the L_2 norms of the residual on the fine and coarse levels

$$L_{2,h} = \sqrt{\sum_{i=1}^{N_h} \frac{\hat{\mathbf{r}}_{h,i}^2}{\Omega_{h,i}}}, \tag{23}$$

$$L_{2,H} = \sqrt{\sum_{i=1}^{N_H} \frac{\hat{\mathbf{r}}_{H,i}^2}{\Omega_{H,i}}}, \tag{24}$$

wherein it is assumed that $\hat{\mathbf{r}}$ is volume scaled, so that taking the square of the residual and dividing it by the cell volume Ω is effectively multiplying the unscaled residual squared by the cell volume. This way the norms on the fine and coarse level give exactly the same value when the residual can be perfectly represented on the coarse level (e.g. residual is constant). These norms can be determined for each governing equation separately to avoid an inaccurate estimation when the equations are not solved dimensionless. The high wave number residual on the fine grid that cannot be represented on the coarse level (and hence no correction computed), can be estimated as the difference $L_{2,h} - L_{2,H}$. We now define the coarse grid residual ratio as

$$r_{CG} = \frac{L_{2,H}}{L_{2,h} - L_{2,H}}. \tag{25}$$

This ratio approaches zero when more sub-iterations are performed on the coarse level. When it is equal to one, it is estimated that the coarse level correction is of the same order as the partitioning error still remaining on the fine level due to higher wave number errors. The Automated Coarse Grid selection algorithm is now denoted by ACG(r) and

selects the coarse level correction when

$$r_{CG} > r. \tag{26}$$

The smaller the value for r, the more sub-iterations are performed on the coarse level and the number of fine level sub-iterations is reduced. However, the lower the value of r more coarse grid corrections are performed that are hardly effective. Especially depending on the difference in computing time for a fine level and coarse level solve, the value of r can be typically chosen between 0.2 and 1. The faster the coarse grid correction compared to a fine level sub-iteration, the lower the value for r can be.

4 RESULTS

The proposed multilevel acceleration technique is applied to a strongly coupled two-dimensional test problem based on [8]. The test case consists of a circular cylinder of diameter 0.1m in a channel with height $H = 0.41$m, length $L = 2.5$m, with an elastic flap behind it of length $l = 0.35$m and thickness $h = 0.02$m, see Fig. 2. The inflow is a parabolic

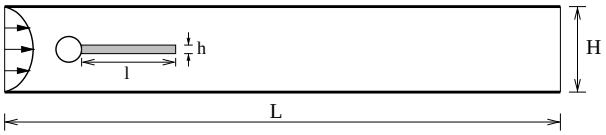

Figure 2: Two-dimensional laminar testcase.

velocity profile (see [8] for details) with a mean velocity of 2 and a maximum velocity of 3m/s. The Reynolds number is based on the mean velocity and cylinder diameter and is $Re = 200$. The structure is modeled as a linear elastic structure with a density equal to the flow density $\rho = 1000\text{kg/m}^3$ and a Young's modulus of $E = 5.6 \cdot 10^6\text{kg/(m.s}^2)$. Time integration is performed by an implicit, third-order accurate, multistage Runge-Kutta scheme with a time step $\Delta t = 0.01$s (for details on time integration see [10]). Each implicit Runge-Kutta stage is subiterated until the discrete pressure forces on the structure mesh $||\mathbf{p}^i_{\Gamma_{sf}} - \hat{\mathbf{p}}^i_{\Gamma_{sf}}||_2 \leq 10^{-3}$. The multigrid levels in the fluid are obtained by agglomeration of fine level cells such that the coarsening ratio is close to the theoretical optimal value of 4 (for two-dimensional applications). However, as the mesh is deforming, the lay-out of the coarse grid may change and even the amount of cells on the coarse level may differ. Therefore, the ILU preconditioner for the JFNK algorithm on the coarse level cannot be reused and has to be rebuild for every new coarse grid sub-iteration. In this paper only the fine mesh (21307 cells) and the medium mesh (5859 cells) are used in the multilevel acceleration. However, the ACG(r) can just as well be used to identify even coarser meshes to perform sub-iterations on.

First the computational work is determined for each part of the algorithm; a distinction is made between solving the fluid domain on the fine grid (FG) or solving the coarse

Table 1: Computational work (scaled by FG/MG(50)) and convergence for solving single sub-iteration.

Algorithm	*CPU time* [-]	*Overhead*	*Convergence* [^{10}log]
FG/MG(50)	1	15%	-0.01
FG/MG(500)	8.8	2%	-0.1
FG/JFNK	1.75	9%	< -6
CG/MG(50)	0.35	43%	-0.005
CG/MG(500)	1.6	9%	-0.05
CG/JFNK	0.6	25%	< -6

Table 2: Number of sub-iterations and computational work (scaled by the work for FG/JFNK) for different fine grid and ACG(0.5) solver settings for solving one time step.

Algorithm	*# CG/stage*	*# FG/stage*	*CPU time* [-]
FG/JFNK	–	17	1
FG/MG(50)	–	220.7	6.7
FG/MG(500)	–	31	7.8
ACG(0.5)-CG/MG(50)-FG/MG(50)	306	46.7	4.9
ACG(0.5)-CG/MG(500)-FG/MG(50)	45.7	14	2.6
ACG(0.5)-CG/MG(500)-FG/MG(500)	33	9	3.9

grid correction (CG) and whether the multigrid solver is used with x iterations (MG(x)) or that the JFNK algorithm is applied (JFNK). The computational work includes any overhead costs such as the agglomeration of the mesh and file I/O. The computational work with respect to 50 multigrid iterations on the fine grid, MG(50), is given in Table 1, including the percentage of overhead and the (average) reduction in residual reached. To investigate the influence of not reaching the required convergence level before starting a new sub-iteration, the number of sub-iterations (per implicit Runge-Kutta stage) and computational work is compared for iterating only on the fine level with either JFNK, FG/MG(500) and FG/MG(50) and compared to the ACG(0.5) algorithm with multigrid solvers used on both the fine and the coarse level, see Table 2. The first part of the table shows that the FG/JFNK algorithm, with full convergence of the flow solver, requires the least amount of sub-iterations. The multigrid solver performs poorly as the computational time is increased by at least a factor 6.7. The second part of the table shows that when an automated multilevel algorithm with multigrid solves on both levels is used, an increased convergence on the coarse mesh strongly reduces the number of sub-iterations on both the coarse level (reduction of 85%) and even fine grid sub-iterations (reduction of 70%). The additional work performed on the coarse level results in a net reduction in computing time of 47%. However, when the convergence level on the fine grid is increased as well, the additional work performed for the fine grid solver is not compensated by the further reduction of sub-iterations. Therefore we choose the hybrid scheme to work with JFNK on the coarse level to obtain high convergence and MG(50) on the fine level to reduce

Table 3: Number of sub-iterations and computational work (scaled by the work for FG/JFNK) for the hybrid ACG(r) scheme for solving one time step.

Algorithm	*# CG/stage*	*# FG/stage*	*CPU time* [-]
ACG(0.25)-CG/JFNK-FG/MG(50)	41.3	4.7	0.98
ACG(0.5)-CG/JFNK-FG/MG(50)	32.3	8	0.90
ACG(1.0)-CG/JFNK-FG/MG(50)	24.3	13.7	0.93
ACG(0.5)-CG/JFNK-FG/JFNK	15.3	6	0.67

computational costs and memory requirements. The results for the hybrid scheme are presented in Table 3. The influence of the residual ratio in the ACG(r) algorithm is as expected: the smaller the value for r, the more the reduction in fine level sub-iterations at the expense of an increased number of coarse level sub-iterations. The smaller the amount of work for a coarse level sub-iteration compared to the fine level iteration, the smaller the value of r can be. In our case ACG(0.5) has the lowest computational time and although the fine grid sub-iterations only use 50 multigrid cycles and do not reduce the residual as much as JFNK, it is even 10% faster than FG/JFNK. When compared to the multilevel algorithms that use the multigrid solver on both fine and coarse grids, the hybrid scheme has a reduction in computing time by a factor of 3. Finally, the ACG(0.5) using JFNK on both the fine and coarse level shows the highest reduction in computing time of 33% compared to FG/JFNK.

5 CONCLUSIONS

In this paper a hybrid multilevel algorithm is proposed which uses JFNK to solve the coarse grid correction term and nonlinear multigrid using only a specified number of iterations to solve the fine grid fluid equations. For incompressible flow the multigrid solver has a low convergence rate and the number of sub-iterations to obtain the fully converged solution increase by a factor of 12 when only 50 multigrid iterations are performed compared to converging every sub-iteration to a high precision. However, sub-iterations can be effectively reduced by imposing a high level of convergence only for the coarse grid correction term. With the proposed automated coarse grid selection algorithm with a ratio $r = 0.5$, a reduction of 10% in computing time was obtained with the hybrid scheme compared to sub-iterating only on the fine mesh with the JFNK solver. When the hybrid scheme is compared to a multilevel algorithm which only uses the multigrid algorithm, a reduction in computing time of a factor of 3 is achieved. Although the multilevel algorithm with JFNK on both the fine and the coarse level shows even a reduction of 33% in computing time, the advantage of the hybrid algorithm is that the memory intensive JFNK is not required on the fine mesh.

Currently the fine level JFNK solver reuses the ILU preconditioner during sub-iterating, reducing the computational cost of updating the preconditioner continuously. Currently the same strategy is not employed for the coarse level JFNK solver, as the coarse mesh

is rebuild at every sub-iteration and the agglomeration procedure can result in different coarse level meshes. When the agglomeration would remain fixed, an additional performance increase could be expected for the coarse level JFNK as the ILU preconditioner could then be reused.

REFERENCES

[1] C. Forster, W. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Comput. Methods Appl. Mech. Engrg. 196 (7) (2007) 1278–1293.

[2] D. Mok, W. Wall, E. Ramm, Accelerated iterative substructure schemes for instationary fluid-structure interaction, in: First MIT Conference on Computational Fluid and Solid Mechanics, 2001, pp. 1325–1328.

[3] R. Haelterman, J. Degroote, D. v. Heule, J. Vierendeels, The quasi-newton least squares method: A new and fast secant method analyzed for linear systems, SIAM J. Numer. Anal. 47 (2009) 2347–2368.

[4] A. v. Zuijlen, S. Bosscher, H. Bijl, Two level algorithms for partitioned fluid-structure interaction computations, Comput. Methods Appl. Mech. Engrg. 196 (2007) 1458–1470.

[5] P. Lucas, A. v. Zuijlen, H. Bijl, A preconditioned jfnk algorithm applied to unsteady incompressible flow and fluid structure interaction problems, Comput. Model. in Engrg. & Sci. 59 (2010) 79–105.

[6] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.

[7] P. Lucas, A. v. Zuijlen, H. Bijl, Fast unsteady flow computations with a jacobian-free newton krylov algorithm, J. Comput. Phys. 229 (2010) 9201–9215.

[8] S. Turek, J. Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, in: H.-J. Bungartz, M. Schaefer (Eds.), Fluid-Structure Interaction: Modelling, Simulation, Optimisation, Springer, 2006.

[9] A. d. Boer, A. v. Zuijlen, H. Bijl, Review of coupling methods for non-matching meshes, Comput. Methods Appl. Mech. Engrg. 196 (2007) 1515–1525.

[10] A. v. Zuijlen, A. d. Boer, H. Bijl, Higher order time integration through smooth mesh deformation for 3d fluid-structure interaction simulations, J. Comput. Phys. 224 (2007) 414–430.