 CMES, vol.59, no.1, pp.79-105, 2010

A Preconditioned JFNK Algorithm Applied to Unsteady Incompressible Flow and Fluid Structure Interaction Problems

Peter Lucas[1] **and Alexander H. van Zuijlen**[1] **and Hester Bijl**[1]

Abstract: Despite the advances in computer power and numerical algorithms over the last decades, solutions to unsteady flow problems remain computing time intensive.
In previous work [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)], we have shown that a Jacobian-free Newton-Krylov (JFNK) algorithm, preconditioned with an approximate factorization of the Jacobian which approximately matches the target residual operator, enables a speed up of a factor of 10 compared to nonlinear multigrid (NMG) for two-dimensional, large Reynolds number, unsteady flow computations. Furthermore, in [Lucas, P., Zuijlen, A.H. van, and Bijl, H. (2010)] we show that this algorithm also greatly outperforms NMG for parameter studies into the maximum aspect ratio, grid density and physical time step: speeds ups, up to a factor of 25 are achieved.
The goal of this paper is to demonstrate the wider applicability of the preconditioned JFNK algorithm by studying incompressible flow and an incompressible fluid structure-interaction (FSI) case. It is shown that the preconditioned JFNK algorithm is able to tackle the stiffness induced by the low Mach regime, making it possible to apply a compressible flow solver to nearly incompressible flow. Furthermore, it is shown that the preconditioned JFNK algorithm can be readily applied to FSI problems.

Keywords: Jacobian-free Newton-Krylov, low Mach number, unsteady flow, fluid-structure interaction.

1 Introduction

The vast majority of flow fields encountered in engineering applications is unsteady. These flow fields may include inherently unsteady flow separation, un-

[1] Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands

steady boundary- and free shear flows, as well as unsteady boundary conditions, possibly caused by flow actuators. In spite of this, since one is often concerned with only mean flow quantities and steady flow simulations are far less computationally demanding, unsteady flow simulations have historically been rarer. However, knowledge of unsteady loads on, for example, wind turbine blades have proved to be of vital importance for an efficient life cycle design, see [Nijssen, R.P.L. (2006)] and [Veldkamp, H.F. (2006)]. Nowadays, with advances in computer power and numerical algorithms, it has become feasible to perform unsteady flow simulations. However, as solutions to unsteady flow problems remain computing time intensive, efficiency improvements of numerical algorithms remain of great importance.

The solution method used in many aerodynamic production codes, e.g. [Fluent Inc. (2003)], [FUN3d (2007)] and [Numeca International (2007)], is some basic iterative method, possibly used as a smoother for (non)linear multigrid and possibly in combination with implicit residual smoothing. As observed by others and shown in our previous work [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)], convergence obtained with these solution methods make them impracticable to compute unsteady flows. Amongst others, the stiffness induced by the Reynolds number, maximum aspect ratio and Mach number is poorly tackled with this solution method. Furthermore, for complex turbulence models, possibly with wall functions, (non)linear multigrid might not be consistent over the series of coarser grids.

A more obvious solution method to compute unsteady flows is Newton linearization in combination with a Krylov subspace technique. Newton linearization comes natural for unsteady flow computations, because the initial solution available from the previous time step is usually accurate enough to avoid nonlinear stall or divergence. Although Newton-Krylov methods have only recently been introduced in the computational fluid dynamics (CFD) community, promising results for the computation of unsteady flows have already been obtained, see e.g. [Bijl, H. and Carpenter, M.H. (2005)], [Isono, S. and Zingg, D.W. (2004)], [Jothiprasad, G., Mavriplis, D.J., and Caughey, D.A. (2003)] and [Qin, N., Ludlow, D.K., and Shaw, S.T. (2000)].

In previous work [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)], we have shown that a Jacobian-free Newton-Krylov (JFNK) algorithm, preconditioned with an approximate factorization (AF) of the Jacobian that approximately matches the target residual operator, enables a speed up of a factor of 10 compared to nonlinear multigrid (NMG) for two-dimensional, large Reynolds number, unsteady flow computations. Although an AF preconditioner based on the target residual operator closely resembles A^{-1}, these preconditioners are unpopular because of a possible lack of robustness [Chow, E. and Saad, Y. (1997)] and a large memory consumption. In our previous work [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)], however, we

have shown that the issue of a large memory consumption is overcome with partly lumping the Jacobian before computing the AF preconditioner, whereas the issue of a liable performance is overcome with enhanced diagonal dominance. In [Lucas, P., Zuijlen, A.H. van, and Bijl, H. (2010)] we show that the preconditioned JFNK algorithm also greatly outperforms NMG for parameter studies into the maximum aspect ratio, grid density and physical time step: speed ups of up to a factor 25 are achieved.

The goal of this paper is to demonstrate the wider applicability of the preconditioned JFNK algorithm. Firstly, it is demonstrated that the preconditioned JFNK algorithm can tackle the stiffness induced by the low Mach number regime, making it possible to apply a compressible flow solver to almost incompressible flow. Secondly, it is demonstrated that the preconditioned JFNK algorithm can readily be applied to incompressible fluid-structure interaction (FSI) problems. Because the preconditioned JFNK algorithm is very efficient in dealing with the stiffness induced by low Mach numbers, it is a promising solution method for incompressible FSI problems as these problems are typically more difficult to solve with most common subiteration techniques.

This paper is organized as follows. In Section 2 the original solution method and the preconditioned JFNK algorithm are discussed. In Section 3 the results are discussed. Finally, in Section 4 the conclusions are drawn.

2 Solution method

The general purpose, compressible, CFD solver is Hexstream [Numeca International (2007)]. This finite volume code describes conservation of mass, momentum and energy with the Reynolds averaged Navier-Stokes equations closed with the turbulence model of [Spalart P.R. and Allmaras S.R. (1992)]. Integrated over a control volume Ω with boundary $d\Omega$, conservation of mass, momentum and energy can be written as:

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{u}\mathrm{d}V+\int_{\mathrm{d}\Omega}\mathbf{F}\cdot\mathbf{n}\mathrm{d}S=\int_{\mathrm{d}\Omega}\mathbf{G}\cdot\mathbf{n}\mathrm{d}S, \tag{1}$$

with $\mathbf{u}$ the set of conservative variables and $\mathbf{F}$ and $\mathbf{G}$ the inviscid and viscous tensor respectively. Spatial discretization is performed with a cell centered finite volume method with added artificial scalar Jameson dissipation [Jameson, A., Schmidt, W., and Turkel, E. (1981)]. The code is designed for unstructured grids, however can also handle structured grids. See [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)] for more information on the spatial discretization operator.

Temporal discretization is performed with a third order implicit ESDIRK (Explicit first stage, Single Diagonal coefficient, Implicit Runge-Kutta) scheme because of

a higher efficiency compared to the standard used second order time integration schemes, see e.g. [Bijl, H., Carpenter, M.H., Vatsa, V.N., and Kennedy, C.A. (2002)], [Carpenter, M.H., Kennedy, C.A., Bijl, H., Viken, S.A., and Vatsa, V.N. (2005)] and [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)]. This scheme consists of three second order accurate, implicit stages within each time step. After solving the intermediate, nonlinear stages, the high order solution at the next time level is obtained by combining the solutions at the intermediate stages with certain weigh factors, see [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)] for more information.

The original solver uses explicit pseudo time stepping in combination with nonlinear multigrid to solve the system of nonlinear equations. Therefore, in order to be able to solve unsteady problems, dual time stepping is used. Hence, for unsteady flow computations, (1) is replaced with:

$$\frac{\partial}{\partial \tau}\int_{\Omega}\mathbf{u}\mathrm{d}V+\frac{\partial}{\partial t}\int_{\Omega}\mathbf{u}\mathrm{d}V+\int_{\partial\Omega}\mathbf{F}\cdot\mathbf{n}\mathrm{d}S=\int_{\partial\Omega}\mathbf{G}\cdot\mathbf{n}\mathrm{d}S, \tag{2}$$

where τ is the pseudo time step. For nearly incompressible flows, say for a Mach number smaller than 0.1, pseudo time stepping is inappropriate because of the large disparity in the convective and acoustic eigenvalues. For example, for a one-dimensional Euler flow, the convective and acoustic eigenvalues are given by, respectively, v, $v-a$ and $v+a$, with v the velocity and a the speed of sound. Hence, for a Mach number that approaches zero, the eigenvalues are given by ≈ 0, $-a$ and a, which explains the large disparity in eigenvalues and which makes standard pseudo time stepping inappropriate.

The problem of a large disparity in eigenvalues has long been known and has, amongst others, motivated the development of a low Mach preconditioner to provide fast convergence for low Mach number flows, see e.g. [Turkel, E. and Vatsa, V.N. (2005)], [Potsdam, M.A., Sankaran, V., and Pandya, S.A. (2007)], [Darmofal, D.L. and Siu, K. (1999)], [Hakimi, N. (1997)], [Vigneron, D., Vaassen, J.-M., and Essers, J.-A. (2008)] and [Choi, Y.-H. and Merkle, C. L. (1993)]. Note, another approach to compute low Mach number flows is to use a purely incompressible flow solver, possibly with correction terms to include some compressibility, see e.g. [Nicolás, A. and Bermúdez, B. (2007)] and [Nicolás, A. and Bermúdez, B. (2004)]. However, because we aim at developing a CFD code that can solve flows in all Mach number regimes, a compressible code is required because the correction terms are inaccurate when relatively large Mach number flows are computed.

A low Mach preconditioner modifies the *pseudo* time derivatives (hence the temporal accuracy is not effected by the low Mach preconditioner) in order to remove the large disparity in eigenvalues and thus to improve convergence. In addition, also the artificial scalar Jameson dissipation can be modified. The idea is that for

low Mach number flows less artificial dissipation is required in order to stabilize the scheme, which improves the accuracy of the computation [Turkel, E. and Vatsa, V.N. (2005)]. Because the low Mach preconditioner developed by [Hakimi, N. (1997)] is readily available in our CFD code, we use this low Mach preconditioner in this paper. In order to introduce the preconditioning parameters the preconditioned version of (2) is given here:

$$\frac{\partial}{\partial \tau}\int_{\Omega}\Gamma^{-1}\tilde{\mathbf{u}}\mathrm{d}V+\frac{\partial}{\partial t}\int_{\Omega}\tilde{\mathbf{u}}\mathrm{d}V+\int_{\partial\Omega}\mathbf{F}\cdot\mathbf{n}\mathrm{d}S=\int_{\partial\Omega}\mathbf{G}\cdot\mathbf{n}\mathrm{d}S, \tag{3}$$

where the only difference with the regular equations shown in (2) is due to the preconditioning matrix Γ and a different solution vector $\tilde{\mathbf{u}}$. Following [Hakimi, N. (1997)], for a three dimensional flow with a one-equation turbulence model, Γ^{-1} is given by:

$$\Gamma^{-1}=\begin{pmatrix} \frac{1}{\beta^2} & 0 & 0 & 0 & 0 & 0 \\ \frac{(1+\alpha)u}{\beta^2} & \rho & 0 & 0 & 0 & 0 \\ \frac{(1+\alpha)v}{\beta^2} & 0 & \rho & 0 & 0 & 0 \\ \frac{(1+\alpha)w}{\beta^2} & 0 & 0 & \rho & 0 & 0 \\ \frac{\alpha\mathbf{v}^2+H_\delta-\beta^2}{\beta^2} & 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

where α and β are the preconditioning parameters, u,v,w are the velocity components and H is the enthalpy. Note that the turbulence transport equation is not modified. The solution vector $\tilde{\mathbf{u}}$ is defined by:

$$\tilde{\mathbf{u}}=(p_\delta,u,v,w,H_\delta,\tilde{\nu})^T \tag{5}$$

where $\tilde{\nu}$ is the Spalart-Allmaras turbulence quantity [Spalart P.R. and Allmaras S.R. (1992)] and the subscript δ means zero referenced at the undisturbed pressure or undisturbed enthalpy.

Next, the modification of the artificial dissipation is discussed. In [Turkel, E. and Vatsa, V.N. (2005)] it is stated that a necessary condition on convergence can be achieved by scaling the artificial dissipation with a term proportional to the Mach number squared. In our code the artificial dissipation is modified by scaling the dissipation term with a factor $f(\beta)$. The factor $f(\beta)$ satisfies the following conditions:

$$f(\beta=a_\infty)=1, \quad \text{and} \quad f(\beta<a_\infty)<1. \tag{6}$$

Hence, for $\beta < a_\infty$, the added artificial dissipation is lower than added without the low Mach preconditioner. For $\alpha = 0$ and $\beta^2 = a^2$ the unpreconditioned equations are obtained. Finally, the parameter β is locally scaled with the velocity and the speed of sound as follows:

$$\beta = \beta_{\text{user}} \left(\min \left(1, \frac{v_{\text{user}}}{a_\infty} \right) \right)^2, \tag{7}$$

where β_{user} and v_{user} are user defined parameters and the subscript ∞ refers to the free stream conditions. The idea of this scaling is that, for $v_{\text{user}} < a_\infty$ (it is recommended to set $v_{\text{user}} < v_\infty$) a lower artificial dissipation is achieved in low speed regions, i.e. in a low speed region there holds: $\beta < a_\infty$, which lowers the added artificial dissipation with a factor $f < 1$. From (4) and (7), it follows that the parameters α, β_{user} and v_{user} determine the convergence rate and the pseudo steady state solution. In [Numeca International (2007)] it is shown that $\alpha = -1$ yields the best clustering of eigenvalues and thus optimal convergence rate (unless otherwise noted, a value of -1 for α will be used throughout this paper). Finally, β should be chosen dependent on the Reynolds number, ranging from 10 to 1 for a Reynolds number that ranges from 100 to 1000. For larger Reynolds numbers β should be chosen equal to 1. In the next sections the various elements of the preconditioned JFNK algorithm are briefly described.

2.1 Newton linearization

The basis of a JFNK algorithm is Newton linearization. After spatial and temporal discretization, both (2) and (3) can be casted into the following form:

$$\mathbf{r}\left(\mathbf{u}^k\right) = \mathbf{0}, \tag{8}$$

where $\mathbf{r}\left(\mathbf{u}^k\right)$ is the nonlinear residual at stage k. A Taylor expansion of the nonlinear residual $\mathbf{r}\left(\mathbf{u}^k\right)$ around $\mathbf{u}^N$ (for all stages), neglecting higher order terms and setting $\mathbf{r}(\mathbf{u}^k) = \mathbf{0}$, yields Newton's method:

$$J\left(\mathbf{u}^N\right)\delta\mathbf{u} = -\mathbf{r}\left(\mathbf{u}^N\right), \tag{9}$$

or, in basic linear algebra notation:

$$A\mathbf{x} = \mathbf{b}, \tag{10}$$

where $\mathbf{u}^N$ is the N^{th} Newton iterate for the solution $\mathbf{u}^k$, $A = J(\mathbf{u}^N)$ is the Jacobian evaluated at $\mathbf{u}^N$ and $\mathbf{b} = -\mathbf{r}(\mathbf{u}^N)$ is the nonlinear residual at $\mathbf{u}^N$. The Jacobian and

the Newton iterate contain all flow parameters including turbulence. The solution of (9) is the Newton update, i.e.:

$$\delta \mathbf{u} = \mathbf{u}^{N+1} - \mathbf{u}^{N}. \tag{11}$$

Newton updates are performed until the solution is converged, i.e. until $||\mathbf{r}(\mathbf{u}^{N+1})||_2 <$ tol, where 'tol' is an absolute tolerance. The system of equations given in (9) is real, non-symmetric and typically very ill-conditioned. Typically, the solution variables are grouped per grid cell. For the regular equations the solution vector $\mathbf{u}$ is given by:

$$\mathbf{u} = (\rho_1,\ \rho u_1,\ \rho v_1,\ \rho w_1,\ \rho E_1,\ \tilde{\nu}_1\ \cdots,\ \rho_n,\ \rho u_n,\ \rho v_n,\ \rho w_n,\ \rho E_n\ \tilde{\nu}_n)^T, \tag{12}$$

where ρ is the density, u, v, w are the velocity components, E is the internal energy, $\tilde{\nu}$ is the Spalart-Allmaras turbulence quantity and n is the number of grid cells. Grouping the solution variables in this way results in a better performance than grouping the solution variables per aerodynamic quantity (which would imply that the first n entries are filled with $\rho_1, \cdots, \rho_n$), which is in accordance with results found in the literature, see e.g. [Chow, E. and Saad, Y. (1997)].

Although more a problem of steady flow problems, nonlinear convergence is not guaranteed with a Newton-based method. Therefore, a line search algorithm is used by repacing (11) with:

$$\mathbf{u}^{N+1} = \mathbf{u}^{N} + \zeta\, \delta \mathbf{u}. \tag{13}$$

Starting with 1.0, ζ is multiplied with 0.75 until the L_2 norm of the nonlinear residual decreases with the computed Newton update. Hence, under relaxation is applied to the Newton update ($\delta \mathbf{u}$) until the norm of the new nonlinear residual ($\mathbf{r}\left(\mathbf{u}^{N+1}\right)$) is smaller than the old one ($\mathbf{r}\left(\mathbf{u}^{N}\right)$). It is our experience that the line-search algorithm is only required for relatively large time steps (say, less than 8 time steps per physical period) and few nonlinear multigrid iterations before Newton linearization (say, less than 5, see also Section 2.4). See possibly [Knoll, D.A. and Keyes, D.E. (2004)] for more information and more sophisticated algorithms to guarantee a decrease in nonlinear residual for each Newton update.

2.2 Linear solver

The system of linear equations of (10) is solved with a Krylov subspace iterative method (the Newton updates therefore become approximate Newton updates). The Generalized Minimal RESidual algorithm (GMRES) introduced by [Saad, Y. and Schultz, M.H. (1986)] is used to solve the system of linear equations. This algorithm uses the modified Gram-Schmidt algorithm to construct an orthonormal

basis $\mathbf{v}^1,\ldots,\mathbf{v}^m$ for the Krylov subspace $K^m(A;\mathbf{r}^0)$. For the sake of completness, this solution method is summarized below:

$$\mathbf{y} = \min_{\mathbf{y}\in\Re^k} ||\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}||_2, \tag{14}$$

$$\mathbf{x} = \mathbf{x}^0 + V_m\mathbf{y}, \tag{15}$$

where β is the L_2 norm of the initial residual that corresponds to the initial guess $\mathbf{x}^0$, $\mathbf{e}_1 = (1,0,\ldots,0)^T$, $\overline{H}_m$ is a Hessenberg matrix which entries are formed by the modified Gram-Schmidt algorithm and V_m is formed by the vectors $\mathbf{v}^1,\ldots,\mathbf{v}^m$, see [Saad, Y. and Schultz, M.H. (1986)] for more information. The linear solver is terminated once the L_2 norm of the linear residual has dropped 1.5 orders compared to its initial value, where a null-solution is used as the initial guess.

As also observed by many others (e.g. [Wong, P. and Zingg, D.W. (2008)], [Syamsudhuha and Silvester, D.J. (2003)] and [Blanco, M. and Zingg, D.W. (2006)]) the GMRES algorithm does not require the matrix A: it only requires the action of A on a vector. It is, therefore, not necessary to compute the Jacobian given in (9) which saves significant CPU costs and memory consumption. The Jacobian-vector products are commonly approximated with the following finite difference:

$$J\mathbf{v} \approx \frac{\mathbf{r}(\mathbf{u}+\varepsilon\mathbf{v}) - \mathbf{r}(\mathbf{u})}{\varepsilon}. \tag{16}$$

The matrix-free approximation can cause an unphysical flow representation as the density and turbulence quantities are not guaranteed to be positive. It is, however, our experience that clipping the turbulence quantities (we did not encounter negative values for the density) in the matrix-free approximation leads to more GMRES iterations to convergence. The turbulence quantities for the solution $\mathbf{u}^{N+1}$ are, however, always clipped such that the solution is physical.

Finally, based on [Chisholm, T.T. and Zingg, D.W. (2009)], we have improved the accuracy of the Newton updates (compared to the performance reported in [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)]) with the following row scaling:

$$D_r A\mathbf{x} = D_r\mathbf{b}, \tag{17}$$

where the row scaling D_r is based on the L_2 norms of the residuals corresponding

to the conservation laws:

$$D_r = \begin{pmatrix} r_1 & & & & & & 0 \\ & r_2 & & & & & \\ & & \ddots & & & & \\ & & & r_{\mathrm{n_{var}}} & & & \\ & & & & r_1 & & \\ 0 & & & & & \ddots & \end{pmatrix}, \tag{18}$$

with

$$\begin{aligned} r_1 &= \sqrt{\frac{\sum_i^{\mathrm{n_{cells}}} \mathbf{r}\left(\mathbf{u}_{(i-1)n_{\mathrm{var}}+1}\right)}{\mathrm{n_{cells}}}}, \\ r_2 &= \sqrt{\frac{\sum_i^{\mathrm{n_{cells}}} \mathbf{r}\left(\mathbf{u}_{(i-1)n_{\mathrm{var}}+2}\right)}{\mathrm{n_{cells}}}}, \\ &\vdots \\ r_{n_{\mathrm{var}}} &= \sqrt{\frac{\sum_i^{\mathrm{n_{cells}}} \mathbf{r}\left(\mathbf{u}_{(i-1)n_{\mathrm{var}}+n_{\mathrm{var}}}\right)}{\mathrm{n_{cells}}}}. \end{aligned}$$

This row scaling results in L_2 norms of the residuals corresponding to the conservation laws, that are the same order of magnitude, which yields a more balanced drop in the linear residual. As the (nonlinear) residuals corresponding to the various conservation laws can differ several orders of magnitude, nonlinear convergence can be poor when not scaled, as the iterative linear solver initially neglects the smaller residuals.

2.3 Preconditioning

In order to achieve rapid convergence of the linear problem for each Newton update preconditioning is used. We precondition the linear systems with an AF that approximately matches the target residual operator. In [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)] we have shown that partly lumping the Jacobian before constructing the AF preconditioner yields a modest memory consumption. Robustness is ensured with enhanced diagonal dominance. Note the difference between the AF preconditioner used to precondition the linear systems (that arise after Newton linearization) and the low Mach preconditioner. The low Mach preconditioner only tackles the disparity in acoustic and convective eigenvalues, whereas the AF preconditioner should tackle all sources of stiffness.

The Jacobian is approximately factored with an incomplete lower upper factorization [Saad, Y. *et al.* (2006)] based on the footprint of the Jacobian (ILU(k)), because incomplete multilevel ILU factorizations [Bollhöfer, M., Saad, Y., and Schenk, O.

(2006)] and incomplete dual thresholds factorizations [Bollhöfer, M., Saad, Y., and Schenk, O. (2006)] were found to be much less effective, as also reported and observed by others [Chow, E. and Saad, Y. (1997)], [Pueyo, A. and Zingg, D.W. (1998)] and [Wong, P. and Zingg, D.W. (2008)]. ILU(k) implies that a zero entry in the Jacobian may only become a nonzero in the ILU when it is up to k positions away from an existing nonzero element in the Jacobian. A level of fill-in of 3 is used because this significantly improves performance for relatively hard cases, whereas it makes the relatively easy cases only marginally more expensive compared to when a level of fill-in of 1 is used.

Finally, the update frequency of the AF preconditioner is once per physical time step. This implies that the preconditioner is computed for the first Newton update of the first Runge-Kutta stage, after which it is recycled for all Runge-Kutta stages and Newton updates within one physical time step. For temporal errors in the order of 1 to 5 percent, an update frequency of once per physical time step minimizes computing time [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)].

2.4 Successive combination NMG and JFNK

Although unsteady flows are computed and thus a relatively accurate initial solution is available, Newton's method is not guaranteed to converge, i.e. when the initial solution is not in the ball of convergence of the Newton method, nonlinear stall or divergence may occur. However, once in the ball of convergence, super linear to quadratic convergence is possible.

Nonlinear multigrid, on the other hand, exhibits a rapid 'initial' convergence (the 'shape' of the solution is captured in only a few iterations), whereas asymptotic convergence is slow. In [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)] and [Bijl, H. and Carpenter, M.H. (2005)] we have shown that a successive combination of respectively NMG and JFNK can significantly reduce the computing time. In [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)] computing times are reduced with 20% to 40% for a two-dimensional turbulent flow around a wind turbine profile (Mach number is 0.121, Reynolds number between $1.0 \cdot 10^4$ and $1.0 \cdot 10^6$). It is our experience that the successive coupling is virtually insensitive to the level of fill-in used in the preconditioner.

The successive coupling is, unfortunately, sensitive to the test case. For example, when the number of grid cells is increased, typically more multigrid iterations before starting JFNK is beneficial, typically because NMG converges slower for the more difficult cases. As of yet we have not found a suitable norm that indicates when to switch from NMG to JFNK. Therefore, 5 NMG iterations are performed before the first Newton update is computed. Five NMG (rather than 1 as reported in [Lucas, P., Bijl, H., and Zuijlen, A.H. van (2010)]) iterations significantly improves

performance for relatively hard cases, whereas it makes the relatively easy cases only marginally more expensive compared to when only 1 NMG iteration is used.

3 Numerical results

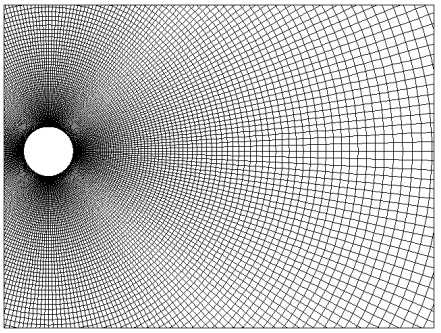
In this section the results are discussed for the low Mach number flow over a circular cylinder and the incompressible FSI case. The solver settings used are the following: the physical time step is chosen such that engineering order of accuracy is achieved, the number of NMG iterations before Newton linearization is 5 for the cylinder flow, whereas it is $25 \times \left(\frac{1}{2}\right)^{n_{\text{sub}}}$ for the FSI CASE (n_{sub} is the current id of the subiteration; this strategy results in relatively many NMG iterations for the harder subiterations, whereas it minimizes computational costs for the easier subiterations), the nonlinear convergence is set such that the iteration error is below the temporal error (which is obtained by means of temporal convergence study for the cylinder flow), the linear convergence criterion is set to -1.5, the level of fill-in used in the AF is 3 (update frequency is once per physical time step). Finally, the grid spacing is chosen such that engineering order of accuracy is achieved (based on the experience gained in [Lucas, P., Zuijlen, A.H. van, and Bijl, H. (2009)]).

3.1 Low Mach number flow around a circular cylinder

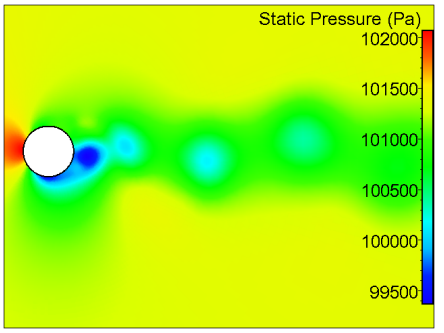
The performance of the preconditioned JFNK algorithm is investigated by comparing it with NMG. The unsteady, laminar circular cylinder flow has the following characteristics: the Reynolds number is 1000, approximately 16 points per physical period are used and the grid contains 43.7k cells. See Fig. 1 for the grid, a snapshot of pressure, time history of c_l and c_d and the curl of velocity. A structured o-grid is used because irregular cells reduce the numerical accuracy for low Mach flows in the CFD solver [Numeca International (2007)]. The following Mach numbers are tested: 0.003, 0.01, 0.03 and 0.1.

In order to investigate whether JFNK can tackle the stiffness due to the low Mach regime, the regular flow equations given in (2) need to be solved. Also the low Mach preconditioned flow equations are studied because we want to get insight into the effectiveness of the low Mach preconditioner on the numerical accuracy and on the performance of NMG and JFNK. First, however, it is investigated whether this low Mach preconditioner improves the numerical accuracy for low Mach number flows.

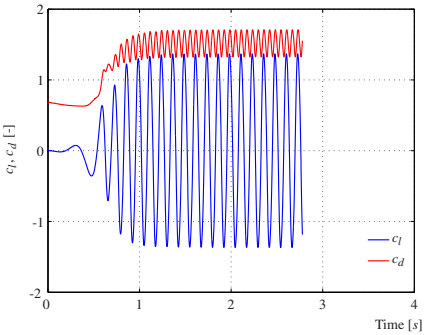
In Tab. 1 the amplitude for lift (a_{c_l}) and drag (a_{c_d}), the mean value for drag ($\bar{c_d}$) and the Strouhal (St) number are given for various settings of the preconditioning parameters. Because compressibility effects can physically be neglected for the given range of Mach numbers, the aforementioned coefficients should be independent of



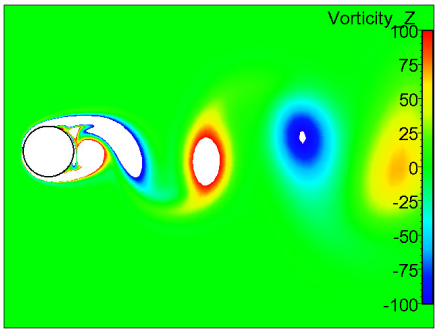
(a) Structured o-grid.



(b) Snapshot static pressure.



(c) Time history c_l and c_d.



(d) Snapshot curl$_z$ of the velocity.

Figure 1: Grid, static pressure, time history c_l c_d and snapshot curl$_z$ velocity laminar, incompressible, two dimensional cylinder flow. Mach number is 0.1, Reynolds number is $1.0 \cdot 10^3$ and grid contains 43.7k cells.

the Mach number. Table 1 shows that this is only the case for the low Mach preconditioned equations ($v_{\text{user}} = v_\infty$) and not for the regular equations ($\beta_{\text{user}} = 1$ and $v_{\text{user}} = a_\infty$), e.g. for the regular equations, the amplitude in lift varies between 1.58 and 2.74. This implies that a modification of the artificial dissipation is indispensable in order to achieve high numerical accuracy.

Next, the performance of JFNK is investigated for the low Mach preconditioned and regular equations. For the low Mach preconditioned case we set $\beta_{\text{user}} = 1$, because this leads to the highest numerical accuracy (lowest added artificial dissipation). In Tab. 2 the average number of NMG and GMRES iterations to solve one nonlinear system are shown as function of the Mach number and whether or not the equations are low Mach preconditioned. Nonlinear convergence is judged by comparing the

Table 1: Amplitude of lift (a_{c_l}) and drag (a_{c_d}), averaged drag ($\bar{c_d}$) and Strouhal number as function of the preconditioning settings ($\beta_{\text{user}} = 1$ is abbreviated as β_u).

M		$v_{\text{user}} = v_\infty$			$v_{\text{user}} = a_\infty$, $\beta_u = 1$
		$\beta_u = 1$	$\beta_u = 3$	$\beta_u = 30$	(regular eqns.)
0.003	a_{c_l}	2.71	2.76	2.76	1.58
	a_{c_d}	0.447	0.447	0.413	0.0717
	$\bar{c_d}$	1.50	1.52	1.52	1.14
	St	0.25	0.24	0.24	0.14
0.01	a_{c_l}	2.71	2.76	2.76	2.27
	a_{c_d}	0.447	0.447	0.413	0.217
	$\bar{c_d}$	1.50	1.52	1.52	1.35
	St	0.25	0.24	0.24	0.20
0.03	a_{c_l}	2.71	2.76	2.76	2.59
	a_{c_d}	0.447	0.446	0.413	0.301
	$\bar{c_d}$	1.50	1.52	1.52	1.47
	St	0.25	0.24	0.24	0.22
0.1	a_{c_l}	2.71	2.76	2.76	2.74
	a_{c_d}	0.446	0.46	0.413	0.385
	$\bar{c_d}$	1.50	1.52	1.52	1.52
	St	0.24	0.24	0.24	0.24

iteration error in lift and drag with the temporal error in lift and drag, where the error is defined as the discrete L_2 error over one complete physical period. In order not to contaminate the time-accurate solution, the iteration error should be smaller than the temporal error (obtained by means of a temporal convergence study).

From Tab. 2 several observations can be made. Firstly, the number of NMG and GMRES iterations heavily depends on the Mach number for the regular equations, while it is virtually constant for the low Mach preconditioned equations. The low Mach preconditioner, therefore, does what it has been designed for: making the convergence independent on the Mach number (and enhancing the numerical accuracy). Secondly, for a Mach number of 0.03 and 0.1, more NMG iterations are required for the low Mach preconditioned than for the regular equations. This makes sense because the low Mach preconditioning results in a lower artificial dissipation. Thirdly, for the regular equations, both the number of NMG and GMRES increase significantly for a reduction in Mach number. The increase in GMRES iterations is, however, less dramatic (factor 5.07) than the increase in NMG iterations (factor 26.4). Finally, for the low Mach preconditioned equations solved with NMG, the

Table 2: Average number of NMG and GMRES iterations to solve one nonlinear system as function of the Mach number and it. convergence criterion (it. conv. crit.), $\varepsilon_{\text{iter}}$ stands for iteration error, $\varepsilon_{\text{temp}}$ stands for the temporal error.

low Mach precond.	it. conv. crit.	M = 0.003		M = 0.01		M = 0.03		M = 0.1	
		NMG	GMR	NMG	GMR	NMG	GMR	NMG	GMR
yes	$\varepsilon_{\text{iter}_L} < \varepsilon_{\text{temp}_L}$	229	20.0	229	20.0	228	20.0	221	19.9
	$\varepsilon_{\text{iter}_D} < \varepsilon_{\text{temp}_D}$	27	7.69	83	7.58	25	7.50	30	7.73
no	$\varepsilon_{\text{iter}_L} < \varepsilon_{\text{temp}_L}$	792	105	246	57.6	87	30.5	30	20.7
	$\varepsilon_{\text{iter}_D} < \varepsilon_{\text{temp}_D}$	605	105	231	57.6	82	30.5	23	20.7

iterative convergence criterion based on the accuracy in lift results in much more NMG iterations (on average 227) than the iterative convergence criterion based on the drag. However, because the iterative error may not contaminate the solution, the comparison in performance is based on the convergence criterion based on the lift.

In Tab. 3 the performance of JFNK is compared with the performance of NMG. This table clearly shows that JFNK greatly outperforms NMG for all Mach numbers and whether or not the flow equations are low Mach preconditioned. On average, the largest average reduction in CPU time is achieved for the low Mach preconditioned cases. The largest speed up is, however, achieved for the regular equations and is as much as a factor of 16.2! Note, for the regular equations and a Mach number of 0.1, the speed up achieved is 'only' a factor 1.48 (approximately 50 percent). This is because this test case is a rather easy one once the stiffness induced due to the Mach number is removed. As NMG is much more suited to compute such flows, the speed up achieved is modest due to a much better performance of NMG. Finally, Tab. 3 shows that costs for JFNK increase from 837 seconds to 2023 (factor 2.42) seconds when the Mach number is reduced from 0.1 to 0.003. Compared to the increase in GMRES iterations (factor of 5.07), the increase in total CPU costs is much lower because the linear solver only partly contributes to the total CPU costs.

To conclude, whether or not low Mach preconditioned, the performance of JFNK is not severely influenced by the Mach number for Mach numbers as lows as 3.0×10^{-3}. This makes it possible to set the value for β based solely on numerical accuracy considerations.

3.2 Incompressible FSI case

In this section the incompressible, strongly coupled, partitioned FSI case is considered. Such flow problems are difficult to compute because a small change in

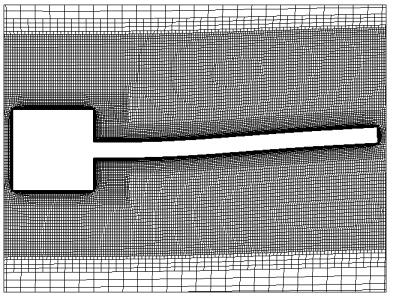
Table 3: CPU costs NMG and JFNK in seconds.

low Mach precond.	Mach number	NMG	JFNK	ratio
yes	0.003	9288	639	14.5
	0.01	9288	639	14.5
	0.03	9287	639	14.5
	0.1	8964	639	14.1
no	0.003	32694	2023	16.2
	0.01	10155	1331	7.63
	0.03	3591	966	3.72
	0.1	1238	837	1.48

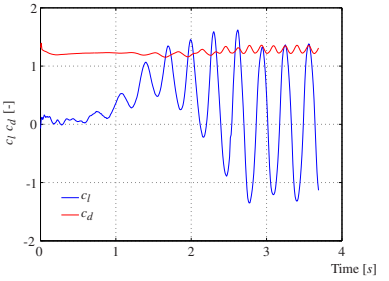
the flow can cause a strong deformation of the structure and vice versa. In order to properly solve strongly coupled, partitioned FSI cases typically subiterations are required. This implies that in order to march forward one physical time step, the flow and the structure need to be solved multiple times. In this section it is investigated whether NMG, and in particular the preconditioned JFNK algorithm, can cope with this additional complexity.

The test case discussed in this section is based on the benchmark problem by [Turek, S. and Hron, J. (2006)]. The original test case as proposed by [Turek, S. and Hron, J. (2006)] concerns a laminar incompressible channel flow around a circular cylinder connected to an elastic flap which results in a self-induced oscillation of the flap. The test case that is considered in this section concerns a *turbulent* incompressible channel flow around a *square* cylinder connected to an elastic flap. See Fig. 2 for a close up of the grid, the time history of c_l and c_d and snapshots of the pressure and curl of velocity and the moment of maximum upward deflection, no deflection and maximum downward deflection. Turbulent rather than laminar flow must be considered because this is more challenging for JFNK. Typically, turbulence can cause difficulties for a Newton-based method because turbulence can cause strong nonlinearities. The flow around a square rather than a circular cylinder is considered, because this results in larger deflections of the flap which also makes this problem more challenging as larger deflections imply a greater nonlinearity. Because we have deviated significantly from the original benchmark problem, it is not possible to compare the numerical accuracy achieved. However, because the focus is on the nonlinear solver this is not considered an issue.

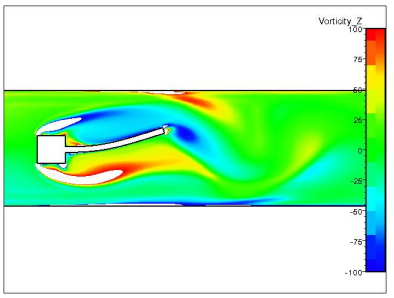
The relevant parameters for this two-dimensional, turbulent, incompressible test case are as follows. The density ratio of the fluid and flap is 1, the Reynolds number is 1.0×10^5 and the Mach number is 5.9×10^{-3}. The grid contains 56.7k grid cells,



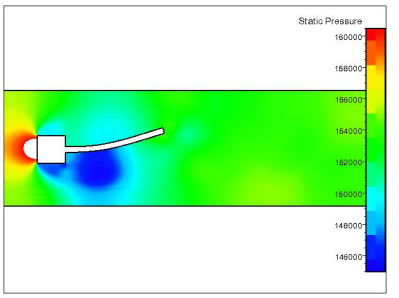
(a) Grid wing and mirror plan.



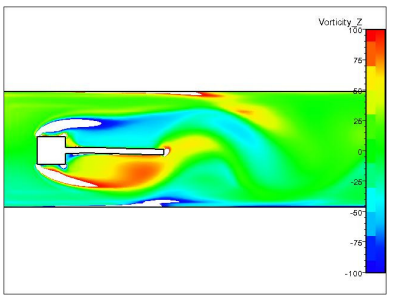
(b) Time history c_l and c_d.



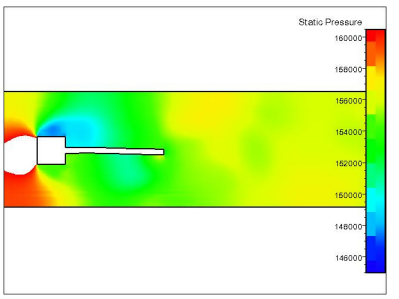
(c) curl(V) at largest positive deflection.



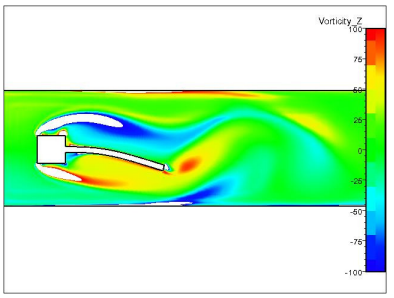
(d) Pressure at largest positive deflection.



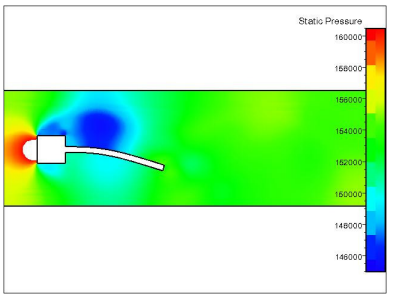
(e) curl(V) at nearly zero deflection.



(f) Pressure at nearly zero deflection.



(g) curl(V) at largest negative deflection.



(h) Pressure at largest negative deflection.

Figure 2: Close up grid, time history c_l, c_d, curl velocity (curl(V)) perpendicular to mirror plane, and static pressure at different positions in time.

whereas the ESDIRK-3 scheme with approximately 30 time steps per physical period is used to march in time. The flow is low-Mach preconditioned with the following settings: $\beta_{\text{user}} = 3$ and $v_{\text{user}} = v_\infty$.

Aitken subiterations are performed until the coupling error (L_2 norm of pressure) is reduced below 0.01. This algorithm can be summarized as follows: when the structure receives the aerodynamic loads $\mathbf{f}_a^i$ after subiteration i, the actual loads applied to the structure $\hat{\mathbf{f}}_a$ for subiteration $i+1$ yield:

$$\hat{\mathbf{f}}_a^{i+1} = \hat{\mathbf{f}}_a^i + \theta^{i+1}\left(\mathbf{f}_a^i - \hat{\mathbf{f}}_a^{i+1}\right), \tag{19}$$

wherein the underrelaxation parameter θ is obtained using the Aitken method

$$\theta^{i+1} = \theta^i \left(1 \frac{(\Delta\mathbf{e}^i)\cdot(\mathbf{e}^i)}{(\Delta\mathbf{e}^i)\cdot(\Delta\mathbf{e}^i)}\right), \tag{20}$$

wherein $\mathbf{e}^i = \mathbf{f}_a^i - \hat{\mathbf{f}}_a^i$, the difference between the loads applied to the structure at the start of the subiteration and the loads computed by the flow solver at the end of the subiteration and $\Delta\mathbf{e}^i = \mathbf{e}^i - \mathbf{e}^{i-1}$. At the start of a new time step or stage, the structure receives the aerodynamic loads from the flow solver and $\hat{\mathbf{f}}_a^1 = \mathbf{f}_a^0$. After the first sub-iteration, underrelaxation is applied using (19) with an underrelaxation $\theta = \min(\theta_{\text{init}}, \theta_{\text{old}})$, which is the minimum value of a prescribed initial value and the underrelaxation value obtained at the end of the previous time step or stage. In this paper we choose $\theta_{\text{init}} = 0.1$, which prevents extreme over or undershoots for the second subiteration. Every next subiteration the underrelaxation parameter is obtained using (20). Should the obtained value for θ be larger than 1, it is limited to 1.

In the literature various subiteration techniques are proposed in order to try to limit the number of required subiterations. In this section the relatively simple Aitken [Mok, D., Wall, W., and Ramm, E. (2001)] subiteration technique is applied. This subiteration technique performs well when compressible FSI problems are computed. However, when incompressible FSI problems are computed, typically much more subiterations are required needed to converge the coupling error [Förster, C., Wall, W.A., and Ramm, E. (2007)]. Hence, the Aitken subiteration technique is not the first choice when simulating an incompressible FSI case. Methods that require more implementation effort are e.g. interface GMRES Michler, C., Brummelen, E.H., and Borst, R. de (2005) or reduced order modeling Vierendeels, J., Lanoye, L., Degroote, J., and Verdonck, P. (2007). They have the advantage over the Aitken method that they have faster convergence. Recently a quasi Newton method has been proposed Haelterman, R., Degroote, J., and van and Vierendeels, J. (2009) and applied to fluid-structure interaction coupling Degroote, J., Bathe, K.-J., and

Vierendeels, J (2009) that has a similar implementation complexity as the Aitken method but with superior convergence.

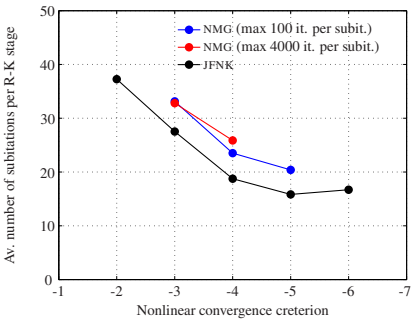
The following two reasons explain why we apply the Aitken subiterations to an incompressible FSI case. One, we are mainly interested whether JFNK can cope with the additional complexity caused by the fluid structure interaction. It is our experience that the Aitken subiteration technique can lead to significant over- and under shoots. A large overshoot typically implies a relatively complex flow configuration (large deflation flap) and relatively a poor initial solution. Hence, when the preconditioned JFNK algorithm can deal with this situation, it can probably also be applied to compressible FSI cases, or to incompressible FSI cases solved with a more sophisticated subiteration technique. Two, when JFNK can successfully be applied to this incompressible FSI case, we want to investigate if JFNK compared to NMG, can help to reduce the number of subiterations required to converge the coupling error.

In order to study the performance of JFNK and NMG, the following investigations are carried out. Firstly, we investigate the performance when 10 physical time steps are computed once a more or less periodic oscillation of the flap is obtained, see Fig. 2(b). Ten physical time steps should be sufficient to extrapolate the total CPU costs to compute one physical period, however saves computational resources. Secondly, we investigate the performance for the very first physical time step. Because the flow is started from uniform flow conditions, the first physical time step is relatively difficult due to the high nonlinearities involved (turbulence needs to be formed which can cause large over and under shoots of the flap). Because JFNK is based on Newton linearization, which may stall or diverge when the initial solution is not in the ball of convergence, it is important to consider the start up phase.

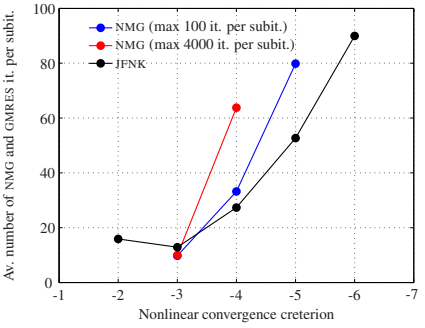
3.2.1 Ten physical time steps during periodic oscillation

It has been our experience that, for this test case, the Aitken subiteration technique leads to a slow convergence of the coupling error, often in an oscillatory manner. Furthermore, for some solver settings (e.g. by a rather loose criterion on the flow iteration error) this method does not converge. Therefore, it is first investigated how the iteration error in the flow solution influences the convergence of the coupling error.

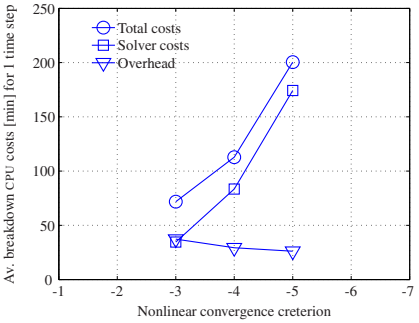
In Fig. 3 the nonlinear convergence criterion is given on the horizontal axis: going from left to right the iteration error in the flow solution is reduced. A nonlinear convergence criterion of -3 implies that the solver is converged once the iteration error has reduced 3 orders in magnitude, compared with the iteration error corresponding to the uniform flow conditions. The convergence criteria tested are: -2, -3, -4, -5 and -6. For NMG and a convergence criterion of -2, the solver diverged



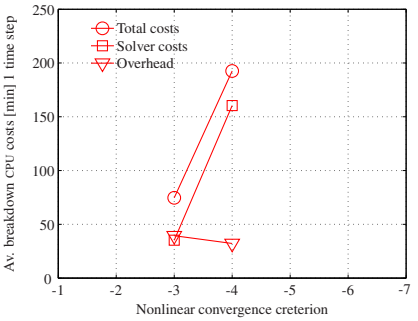
(a) Average number of subiterations per stage.



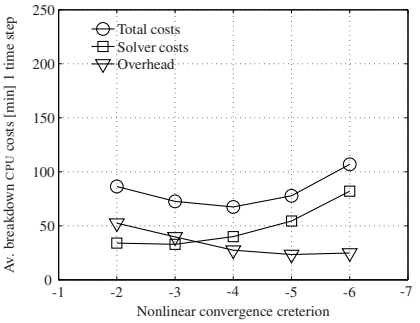
(b) Average number of NMG and GMRES iterations per subiteration.



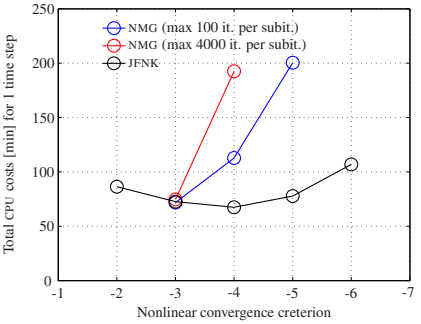
(c) Breakdown CPU costs NMG with a maximum of 100 iterations.



(d) Breakdown CPU costs NMG with a maximum of 4000 iterations.



(e) Breakdown CPU costs JFNK.



(f) Comparison total CPU costs NMG and JFNK.

Figure 3: Average (av.) number of subiterations (subit.) per Runge-Kutta (R-K) stage as function of the nonlinear residual, average number of NMG and GMRES iterations (it.) per subiteration as function of the nonlinear convergence criterion, average breakdown of the CPU costs NMG and JFNK for one physical time step and a comparison for the total CPU costs for one physical time step.

such that no results have been obtained. The other cases are not shown for NMG because we have terminated these computations due to excessive computing times.

In Fig. 3(a) the average number of subiterations per Runge-Kutta stage is shown as function of the nonlinear convergence criterion. In Fig. 3(b) the average number of NMG and GMRES iterations per subiteration are shown as function of the nonlinear convergence criterion. For NMG two settings for the maximum number of iterations per subiterations are shown: maximal 100 NMG iterations such that the work is limited and maximal 4000 NMG iterations such that the target for the iteration error is satisfied. Note, for a maximum of 100 iterations the nonlinear convergence criterion is satisfied once the coupling error is near convergence.

Figure 3(a) shows a clear trend: the further the flow iteration error is reduced, the less subiterations are required to converge the coupling error. However, also Fig. 3(b) shows a clear trend: the further the flow iteration error is reduced, the more NMG and GMRES iterations are required. Hence, the total CPU costs are the combined effect of these two phenomena (discussed later). For a relatively loose convergence criterion more subiterations are required, because the iteration error is not taken into account in the Aitken subiteration technique. Hence, for a relative inaccurate flow solution, the structure is deformed based on the flow solution including iteration error which explains the slower convergence of the coupling error. Note that for all nonlinear convergence criteria, JFNK requires less subiterations than NMG. This is because JFNK convergence in only in a few of Newton updates. Consequently, the level of convergence is typically higher than the target set. We do not have a sound explanation for the minimum in the number of subiterations observed (see Fig. 3(a) for a criterion of -5).

Figures 3(c) to 3(e) show the breakdown of the CPU costs for NMG (maximum of iterations is 100 or 4000) and JFNK for one physical time step. The breakdown is for the CPU costs to solve the nonlinear systems and the overhead. The overhead is mainly due to the grid deformation, rebuilding the agglomerated multigrid levels and data structure and is approximately constant per subiteration. From Figs. 3(c) to 3(e) the following observations can be made. Firstly, the CPU costs for NMG increase rapidly for a reduction in the required flow iteration error. Secondly, the overhead costs reduce significantly for a reduction in the flow iteration error because less subiterations are required. Thirdly, for JFNK there is an optimum for the nonlinear convergence criterion with respect to the total CPU costs. The minimum in CPU costs is the combined effect of less subiterations for a stricter convergence criterion and higher CPU costs to achieve the required drop in nonlinear residual.

Finally, in Fig. 3(f) a comparison is made for the total CPU costs of NMG and JFNK for the various convergence criteria. This figure shows that minimal values for the CPU costs for NMG and JFNK are approximately the same (717 minutes versus 674

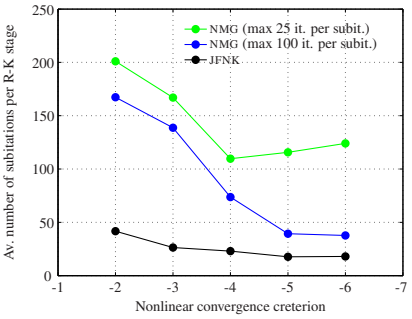
minutes). However, when the number of iterations is limited to 100 and when is started from uniform flow conditions, it has not been possible to get to the periodic oscillation of the flap due to solver divergence. Furthermore, for three-dimensional computations the overhead is relatively more expensive because deformation of the mesh is relatively more expensive in three than in two dimensions. Finally, the robustness increases for a reduction in the number of subiterations (less data transfer between structure and fluid and smaller under and overshoots). Hence, when the goal is to minimize the number of subiterations (which requires a nonlinear convergence criterion of at least -5, see Fig. 3(a)), JFNK is a factor 2.5 faster than NMG (maximum number of iterations limited to 100).

The speed up achieved with JFNK is less impressive than shown in the previous section. This is because the nonlinear convergence criterion with NMG is not satisfied for all subiterations (number of iterations is limited to maximal 100), whereas it is with JFNK. By limiting the number of NMG iterations the coupling strategy is significantly improved (only when started from a periodic solution!): for the first few subiterations the amount of work is limited to 100 iterations, however for the subiterations with a relatively small coupling error, convergence is satisfied because the deformation of the structure is on average smaller for each next subiteration. Hence, by setting a strict convergence criterion while limiting the maximum number of iterations, a relative convergence criterion for the flow is achieved. Consequently, the average number of NMG iterations per subiteration is relatively low, see Fig. 3(b). A similar strategy for limiting the number of Newton updates should, therefore, result in a significant saving. This is, however, considered as future work.

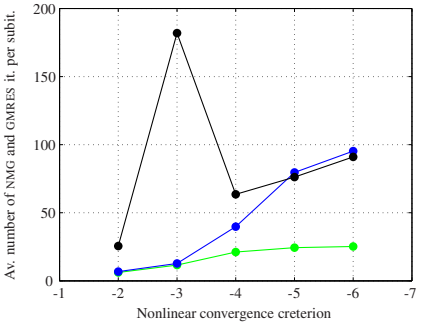
3.2.2 First physical time step

The results achieved for the very first physical time step are presented in a similar way. Figure 4 shows the average number of subiterations per Runge-Kutta stage as function of the nonlinear residual, the average number of NMG and GMRES iterations per subiteration as function of the nonlinear convergence criterion, the breakdown of the CPU costs NMG and JFNK for one physical time step and a comparison of the total CPU costs for one physical time step. In an attempt to further 'improve' the convergence of NMG, the number of multigrid iterations per subiteration is limited to 25 (in addition to a maximum of 100, a maximum of 4000 is discarded because of excessive CPU costs). Even when the number of NMG iterations is limited to 25, the flow solution is converges when the coupling error is sufficiently reduced.

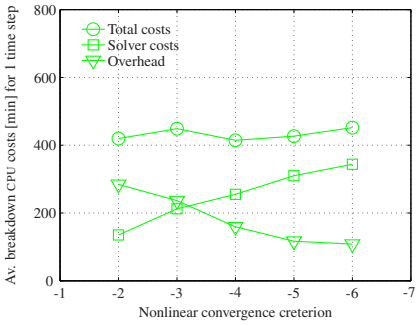
Figure 4(a) shows the number of subiterations per stage. For JFNK and NMG (maximum iterations is 100) a similar trend as in Fig. 3(e) is observed: the stricter the nonlinear convergence the lower the required number of subiterions. For NMG



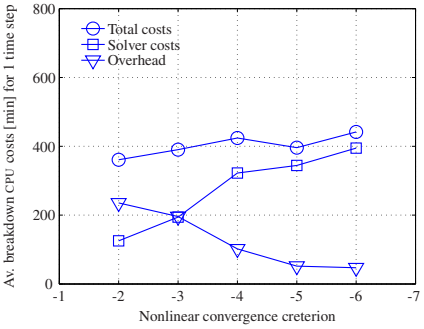
(a) Average number of subiterations per stage versus nonlinear convergence criterion.



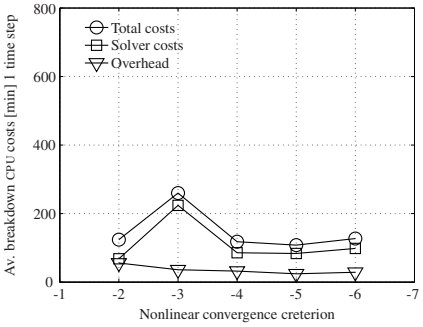
(b) Average number of NMG and GMRES iterations per subiteration (see left figure for legend).



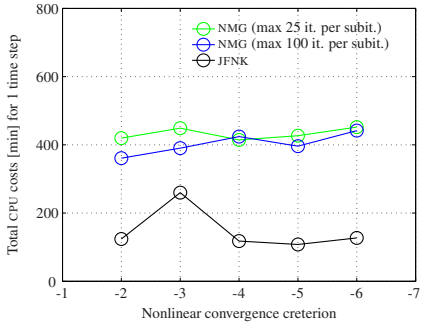
(c) Breakdown CPU costs NMG with a maximum of 25 iterations.



(d) Breakdown CPU costs NMG with a maximum of 100 iterations.



(e) Breakdown CPU costs JFNK.



(f) Comparison total CPU costs NMG and JFNK.

Figure 4: Average (av.) number of subiterations (subit.) per Runge-Kutta (R-K) stage as function of the nonlinear residual, average number of NMG and GMRES iterations (it.) per subiteration as function of the nonlinear convergence criterion, average breakdown of the CPU costs NMG and JFNK for one physical time step and a comparison for the total CPU costs for one physical time step.

(maximum iterations is 25) the required number of subiterations is substantially higher than JFNK. This implies that 25 NMG iterations is not sufficient for the stricter nonlinear convergence criteria.

Figures 4(c) to 4(e) give the breakdown in CPU costs for NMG and JFNK. These results show that again the overhead for one physical time step is reduced with a stricter nonlinear convergence criterion. For NMG the solver and total costs on average increase for a stricter convergence criterion, while for JFNK an optimum is found in the solver and total costs. The peak seen for JFNK and a convergence criterion of -3 is because the flow solver stagnated for one (intermediate) subiteration.

Finally, in Fig. 4(f) the CPU costs obtained with NMG and JFNK are shown. Compared to the results shown in Fig. 3(f), a larger reduction in CPU costs is achieved with JFNK compared to NMG. The minimal computing time with JFNK is 108 minutes, while it is 360 minutes for NMG. Hence, JFNK enables a speed up of a factor 3.33 over NMG.

4 Conclusions

The results discussed in this paper show that the preconditioned JFNK algorithm can tackle nearly incompressible flow, whether or not low Mach preconditioned. This makes it possible to set the preconditioning parameters based solely on numerical accuracy considerations. In addition, it has been shown that the preconditioned JFNK algorithm can be readily applied to FSI problems. For strongly coupled FSI problems it has been shown that the stability and the convergence of the subiterations improved for relative large reductions of the flow iteration error. Hence, when one aims for a minimum number of subiterations, JFNK enables a significant speed up over NMG as this solution method is relatively more efficient in achieving large reductions in the flow iteration error.

References

Bijl, H.; Carpenter, M.H. (2005): Iterative solution techniques for unsteady flow computations using higher order time integration schemes. *International Journal for Numerical Methods in Fluids*, vol. 47, pp. 857–862.

Bijl, H.; Carpenter, M.H.; Vatsa, V.N.; Kennedy, C.A. (2002): Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: laminar flow. *Journal of Computational Physics*, vol. 179, pp. 313–329.

Blanco, M.; Zingg, D.W. (2006): A Newton-Krylov algorithm with a loosely-coupled turbulence model for aerodynamic flows. In *44^{th} AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA. AIAA Paper 2006-691.

Bollhöfer, M.; Saad, Y.; Schenk, O. (2006): Ilupack - preconditioning software package, 2006. Available online at http://www.math.tu-berlin.de/ilupack.

Carpenter, M.H.; Kennedy, C.A.; Bijl, H.; Viken, S.A.; Vatsa, V.N. (2005): Fourth-order Runge-Kutta schemes for fluid mechanics applications. *Journal of Scientific Computing*, vol. 25, pp. 157–194.

Chisholm, T.T.; Zingg, D.W. (2009): A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows. *Journal of Computational Physics*, vol. 228, pp. 3490–3507.

Choi, Y.-H.; Merkle, C. L. (1993): The application of preconditioning in viscous flows. *Journal of Computational Physics*, vol. 105, no. 2, pp. 207–223.

Chow, E.; Saad, Y. (1997): Experimental study of ILU preconditioners for indefinite matrices. *Journal of Computational and Applied Mathematics*, vol. 86, pp. 387–414.

Darmofal, D.L.; Siu, K. (1999): A robust multigrid algorithm for the Euler equations with local preconditioning and semi-coarsening. *Journal of Computational Physics*, vol. 151, pp. 728–756.

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

Fluent Inc. (2003): Fluent 6.1 user's guide. Technical report, Lebanon, NH 03766, 2003.

Förster, C.; Wall, W.A.; Ramm, E. (2007): Artificial added mass instabilities in sequantial staggered coupling of nonlinear structures and incompressible viscous flows. *Computer methods in applied mechanics and engineering*, vol. 196, pp. 1278–1293.

FUN**3d** (2007): FUN3d fully unstructured Navier-Stokes, 2007. http://fun3d.larc.nasa.gov.

Haelterman, R.; Degroote, J.; van and Vierendeels, J., H. (2009): The Quasi-Newton Least Squares method: A new and fast secant method analyzed for linear systems. *SIAM Journal on Numerical Analysis*, vol. 47, pp. 2347–2368.

Hakimi, N. (1997): *Preconditioning methods for time dependent Navier-Stokes equations - Application to enverionmental and low speed flows*. PhD thesis, Vrije Universiteit Brussel, 1997.

Isono, S.; Zingg, D.W. (2004): A Runge-Kutta-Newton-Krylov algorithm for fourth-order implicit time marching applied to unsteady flows. In *42nd AIAA*

Aerospace and Science Meeting and Exhibit, Reno, Nevada, USA. AIAA Paper 2004-433.

Jameson, A.; Schmidt, W.; Turkel, E. (1981): Numerical solutions of the Euler equations by finite volume methods with Runge-Kutta time stepping schemes. In *14th AIAA Fluid and Plasma Dynamics Conference*, pp. 23–25, Palo Alto, California, USA. AIAA Paper 81-1259.

Jothiprasad, G.; Mavriplis, D.J.; Caughey, D.A. (2003): Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes. *Journal of Computational Physics*, vol. 191, pp. 542–566.

Knoll, D.A.; Keyes, D.E. (2004): Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, vol. 193, pp. 357–397.

Lucas, P.; Bijl, H.; Zuijlen, A.H. van (2010): Efficient unsteady high Reynolds number flow computations on unstructured grids. *Computers & Fluids*, vol. 39, pp. 271–282.

Lucas, P.; Zuijlen, A.H. van; Bijl, H. (2009): An automated approach for solution based mesh adaptation to enhance numerical accuracy for a given number of grid cells - *Applied to steady flow on hexahedral grids*. *CMES: Computer Modeling in Engineering & Sciences*, vol. 41, no. 2, pp. 147–175.

Lucas, P.; Zuijlen, A.H. van; Bijl, H. (2010): Fast unsteady flow computations on unstructured grids. *Submitted to Journal of Computational Physics*.

Michler, C.; Brummelen, E.H.; Borst, R. de (2005): An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, vol. 47, no. 10-11, pp. 1189–1195.

Mok, D.; Wall, W.; Ramm, E. (2001): Accelerated iterative substructuring schemes for instationary fluid-structure interaction. In *First MIT Conference on Computational Fluid and Solid Mechanics*, Cambridge, MA, USA. p.p. 1325-1328.

Nicolás, A.; Bermúdez, B. (2004): 2D incompressible viscous flows at moderate and high Reynolds numbers. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6, no. 5, pp. 441–451.

Nicolás, A.; Bermúdez, B. (2007): Viscous incompressible flows by the velocity-vorticity Navier-Stokes equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 20, no. 2, pp. 73–83.

Nijssen, R.P.L. (2006): *Fatigue life prediction and strength degradation of wind turbine rotor blade composits*. Phd thesis, Delft University of Technology, Delft, The Netherlands, 2006. Available online at darenet.nl.

Numeca International (2007): User Manual FINE^{TM}/Hexa v2.5 (including Hexstream) - Flow Integrated Environment. Technical report, Brussels, 2007. www.numeca.be.

Potsdam, M.A.; Sankaran, V.; Pandya, S.A. (2007): Unsteady low Mach preconditioning with application to rotorcraft flows. In 18^{th} *AIAA Computational Fluid Dynamics Conferences*, Miami, Florida, USA. AIAA Paper 2007-4473.

Pueyo, A.; Zingg, D.W. (1998): Efficient Newton-Krylov solver for aerodynamic computations. *American Institute of Aeronautics and Astronautics Journal, Inc.*, vol. 36, no. 11, pp. 1991–1997.

Qin, N.; Ludlow, D.K.; Shaw, S.T. (2000): A matrix-free preconditioned Newton/GMRES method for unsteady Navier-Stokes solutions. *International Journal for Numerical Methods in Fluids*, vol. 33, pp. 223–248.

Saad, Y.; Schultz, M.H. (1986): GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869.

Saad, Y. *et al.* (2006): Itsol - a library of iterative solvers for general sparse linear systems of equations, 2006. Available online at http://www-users.cs.umn.edu/∼saad/software/ITSOL/.

Spalart P.R.; Allmaras S.R. (1992): A one-equation turbulence model for aerodynamic flows. In *AIAA* 30^{th} *Aerospace Sciences Meeting & Exhibit*, Reno, Nevada, USA. AIAA Paper 92-0439.

Syamsudhuha; Silvester, D.J. (2003): Efficient solution of the steady-state Navier-Stokes equations using a multigrid preconditioned Newton-Krylov method. *International Journal for Numerical Methods in Fluids*, vol. 43, pp. 1407–1427.

Turek, S.; Hron, J. (2006): Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In Bungartz, H.-J.; Schäfer, M(Eds): *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pp. 371–385. Springer Berlin Heidelberg. ISSN: 1439-7358.

Turkel, E.; Vatsa, V.N. (2005): Local preconditioners fro steady and unsteady flow applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 39, no. 3, pp. 515–535. DOI: 10.1051/m2an:2005021.

Veldkamp, H.F. (2006): *Chances in Wind Energy: A probabilistic approach to wind turbine fatigue design*. Phd thesis, Delft University of Technology, Delft, The Netherlands, 2006. Available online at darenet.nl.

Vierendeels, J.; Lanoye, L.; Degroote, J.; Verdonck, P. (2007): Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Computers and Structures*, vol. 85, no. 11-14, pp. 970–976.

Vigneron, D.; Vaassen, J.-M.; Essers, J.-A. (2008): An implicit finite volume method for the solution of 3d low Mach number viscous flows using a local preconditioning technique. *Journal of Computational and Applied Mathematics*, vol. 215, no. 2, pp. 610–617.

Wong, P.; Zingg, D.W. (2008): Three-dimensional aerodynamic computations on unstructured grids using a Newton-Krylov approach. *Computers & Fluids*, vol. 37, pp. 107–120.