INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **47**:857–862
Published online 7 January 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.861

Iterative solution techniques for unsteady flow computations using higher order time integration schemes

H. Bijl[1,2,*,†] and M. H. Carpenter[1,2]

[1]*Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands*
[2]*Computational Modeling and Simulation Branch, NASA Langley Research Center, Hampton, VA, U.S.A.*

SUMMARY

In this paper iterative techniques for unsteady flow computations with implicit higher order time integration methods at large time steps are investigated. It is shown that with a minimal coding effort the standard non-linear multigrid method can be combined with a Newton–Krylov method leading to speed-ups in the order of 30%. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: unsteady flows; implicit time integration methods; iterative solvers

1. INTRODUCTION

The vast majority of flowfields encountered in engineering applications are unsteady. These flowfields may include inherently unsteady flow separation, unsteady 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 flowfields are less computationally demanding, unsteady flow simulations have historically been more rare. Contemporary computer resources now render three-dimensional unsteady flowfield computation viable [1]. As these simulations consume vast computational resources, their efficiency is of great importance.

In a previous paper [2] we studied the relative efficiency of implicit time integration methods, concluding that already for engineering levels of accuracy fourth-order Runge Kutta methods are more efficient than second-order methods, for the problems that were tested. As these methods allow for large time steps, major driver for efficiency is then the iterative solution method. The iterative solver originally used is pseudo-time stepping in combina-

*Correspondence to: H. Bijl, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands.
†E-mail: h.bijl@lr.tudelft.nl

Contact/grant sponsor: Publishing Arts Research Council; contract/grant number: 98-1846389

Received 27 April 2004
Revised 28 September 2004
Accepted 28 September 2004

tion with non-linear multigrid (MG) and implicit residual smoothing. As also observed by others, asymptotic convergence of this type of iterative solver for the computation of turbulent flows on high aspect ratio grids is poor. The question is whether efficiency of the iterative method can be improved for unsteady flow calculations with limited changes to the current, Jacobian-free, code. An obvious candidate for this is Newton linearization in combination with a Krylov subspace technique. However, this method may fail when the initial guess is far removed from the domain of convergence of the non-linear problem, which can occur due to our large time steps. Furthermore, efficiency of this method crucially depends on the preconditioner. Combining the methods, using multigrid as a preconditioner and starting with the non-linear multigrid pseudo-time stepping scheme only switching to Newton–Krylov when convergence rates drop might be a good alternative.

There has been considerable success in applying the multigrid method as a preconditioner on linear problems. As a result of this success researchers have applied linear multigrid as a preconditioner to a Jacobian-free Newton–Krylov (JFNK) method, for a complete overview see [3]. Recently, Mavriplis [4] has considered non-linear multigrid as a preconditioner to JFNK with encouraging results. For the steady-state Navier–Stokes equations at high Reynolds numbers, non-linear multigrid as a preconditioner generally outperformed non-linear multigrid as a solver. However, the overall winner was consistently linear multigrid as a preconditioner to JFNK. For unsteady flow computations Jothiprasad *et al.* [5], recently obtained an efficiency improvement with a multigrid preconditioned Newton–Krylov method for relatively small time steps with a second-order backward differencing time integration scheme. Another candidate preconditioner is a Krylov method itself, as it can be easily implemented in a matrix-free fashion. Flexible GMRES [6] and GMRES-R [7] methods were developed to address the issue of using a Krylov preconditioner that may vary within the GMRES iteration. These flexible accelerators open up a number of preconditioning options such as using a relaxation method preconditioner with variable termination from outer to inner iteration.

2. THE METHOD

A generalized form of the classical unsteady thin-layer Navier–Stokes equations which includes all normal components of the viscous terms is used to model the flow. For turbulent flows the Reynolds average formulation is used in combination with the one equation Spalart–Allmaras [8] turbulence model. The spatial terms are discretized using a standard cell-centred finite volume scheme. The convection terms are discretized with second-order central differences with (second- and fourth-order difference) scalar/matrix dissipation added to suppress odd–even decoupling and oscillations in the vicinity of shock waves and stagnation points [9]. The viscous terms are central differences with a second-order formula. The semi-discrete system of equations is denoted by:

$$\mathbf{R}(\mathbf{U}) = \frac{d\mathbf{U}}{dt} + \mathbf{S}(\mathbf{U}(t)) = 0 \tag{1}$$

where $\mathbf{U}$ is the conserved variable vector and the vector $\mathbf{S}$ results from semi-discretization of the fluid mechanic equations. It contains convection and diffusion. Temporal integration of Equation (1) is performed using an implicit fourth-order explicit first-stage diagonally implicit Runge Kutta method (ESDIRK) which was shown to be very efficient even for engineering levels of accuracy [2]. Using this multi-stage scheme, each stage a system of equations has to be solved. In this paper the following iterative solution techniques are compared: a non-linear multigrid or full approximation scheme (FAS) scheme and a Jacobian-free Newton–Krylov method, unpreconditioned, preconditioned by the FAS scheme or used in a recursive fashion.

2.1. The FAS scheme

Each stage of the implicit time integration results in a non-linear algebraic set of equations (see Equation (1)), that is solved using a FAS multigrid scheme. The smoother used on each grid is derived by adding a pseudo-time term to Equation (1) and advancing two pseudo-time steps with an explicit–implicit (IMEX) RK scheme. Local time stepping and implicit residual smoothing are used to further increase the smoothing characteristics. Convergence in pseudo-time is monitored and multigrid iterations are terminated when a specified tolerance is reached. For details, see References [2, 9].

2.2. The Jacobian-free Newton–Krylov method

In the Newton–Krylov method the system of non-linear equations is linearized using a Newton method. The Newton method for $\mathbf{R}(\mathbf{U})=0$ is equal to (see Reference [3] for more details):

$$\mathbf{A}\delta\mathbf{U} = -\mathbf{R}(\mathbf{U}^N), \quad \mathbf{U}^{N+1} = \mathbf{U}^N + \delta\mathbf{U} \tag{2}$$

given $\mathbf{U}^0$. Here, $\mathbf{R}(\mathbf{U})$ is the vector-valued function of non-linear residuals, $\mathbf{A}\equiv\partial\mathbf{R}/\partial\mathbf{U}$ is its associated Jacobian matrix, $\mathbf{U}^{N+1}$ is the $N+1$th Newton approximation of the state vector at the new time level to be computed, and N is the iteration index of the previous Newton iteration.

To solve the linear system Equation (2) a Krylov subspace method is used. We have chosen the general conjugate residual (GCR) method, which is equal to the generalized minimal residual method (GMRES) storing not only the residual, but also the solution. This GMRES variant with twice the storage makes it possible to use a variable preconditioner. The unpreconditioned GCR algorithm to solve $\mathbf{Ax}=\mathbf{b}$, with $\mathbf{b}=-\mathbf{R}(\mathbf{U}^N)$ and $\mathbf{x}=\delta\mathbf{U}$, which stores two search directions $\mathbf{p}_k$ and $\mathbf{q}_k$, is:

GCR algorithm

```
$\mathbf{x}_0 = 0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $k = -1$
while $\|\mathbf{r}_{k+1}\|_2 > \text{tol}$ do
    $k = k + 1$
    $\mathbf{p}_k = \mathbf{r}_k$
    $\mathbf{q}_k = \mathbf{A}\mathbf{p}_k$
    for $i = 0, 1, \ldots, k-1$ do
        $\alpha_i = \mathbf{q}_i^{\mathrm{T}}\mathbf{q}_k$, $\mathbf{q}_k = \mathbf{q}_k - \alpha_i\mathbf{q}_i$, $\mathbf{p}_k = \mathbf{p}_k - \alpha_i\mathbf{p}_i$
    endfor
    $\mathbf{q}_k = \mathbf{q}_k/\|\mathbf{q}_k\|_2, \mathbf{p}_k = \mathbf{p}_k/\|\mathbf{q}_k\|_2$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k\mathbf{q}_k^{\mathrm{T}}\mathbf{r}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{q}_k\mathbf{q}_k^{\mathrm{T}}\mathbf{r}_k$
endwhile
```

In order to limit the amount of programming required to speed-up our original Jacobian-free solver, we use a Jacobian-free GCR implementation using the following first-order finite difference approximation:

$$\mathbf{A}\mathbf{p}_k \approx \frac{\mathbf{R}(\mathbf{U}^N + \varepsilon \mathbf{p}_k) - \mathbf{R}(\mathbf{U}^N)}{\varepsilon} \tag{3}$$

This approximation is sensitive to scaling. If ε is too large, it is a poor approximation, if it is too small the finite difference is contaminated by floating point round-off error. From the ample choices of ε reported for similar situations in the literature, for an overview see Reference [3], we have chosen for

$$\varepsilon = \frac{\sqrt{(1 + \|\mathbf{U}^N\|)\varepsilon_{\text{mach}}}}{\|\mathbf{p}_k\|} \tag{4}$$

with $\varepsilon_{\text{mach}}$ equal to machine precision.

Following Mavriplis [4] preconditioning by a FAS scheme is applied. Both left and right preconditioning was implemented. Using Equation (3) left preconditioning in the GCR algorithm comes down to

$$\mathbf{q}_k = \mathbf{P}^{-1}\mathbf{A}\mathbf{p}_k \approx \frac{\mathbf{P}^{-1}\mathbf{R}(\mathbf{U}^N + \varepsilon \mathbf{p}_k) - \mathbf{P}^{-1}\mathbf{R}(\mathbf{U}^N)}{\varepsilon} \tag{5}$$

where $\mathbf{P}^{-1}$ is an approximation of $\mathbf{A}^{-1}$. FAS preconditioning of the terms in Equation (5) is formulated using Equation (2) as:

$$-\mathbf{P}^{-1}\mathbf{R}(\mathbf{U}^N) = \mathbf{P}^{-1}\mathbf{A}\delta\mathbf{U} = \mathbf{U}^{\text{MG}} - \mathbf{U}^N \tag{6}$$

where $\mathbf{U}^{\text{MG}}$ is the solution vector obtained after one or more FAS iterations, starting with initial guess $\mathbf{U}^N$. The formulation for FAS right preconditioned GCR is obtained by:

$$\mathbf{p}_k = \mathbf{P}^{-1}\mathbf{r}_k \approx \mathbf{P}^{-1}\mathbf{R}(\mathbf{U}^N + \mathbf{x}_k) \tag{7}$$

which is computed using Equation (6). Also recursive GCR [7] was implemented. This is a special right preconditioning, where GCR is used to compute $\mathbf{U}^{\text{MG}}$ in Equation (6).

3. RESULTS

The efficiency of the iterative solvers is investigated for an unsteady flow test case. The test problem is flow around a two-dimensional circular cylinder at a Mach number of 0.3. Both laminar flow at a Reynolds number of 1000 and turbulent flow with a Reynolds number of 10^6 is considered. For the laminar case a mesh of 129×65 was used; for the turbulent case a mesh of 129×129. The computations were started from a solution in the periodic part of time history, which was computed beforehand. For a grid refinement study and a comparison of various time integration schemes for this problem see Reference [2].

In Figure 1, convergence of the non-linear FAS scheme for one time step in the laminar case is compared with the turbulent case. The five spikes are the initial residuals of the

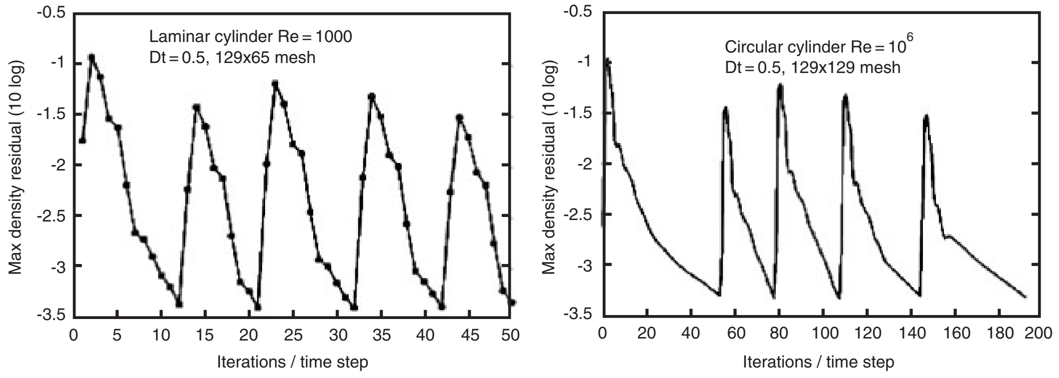

Figure 1. Convergence history of FAS MG solver for laminar and turbulent flow problem.

Table I. Work for laminar flow problem using various iterative solvers.

	No MG first	3 MG first	6 MG first
FAS MG	144 (12+0)	144 (12+0)	144 (12+0)
GCR5-20	234 (0+12)	185 (3+9)	107 (6+3)
GCR5	270 (0+54)	186 (3+30)	107 (6+7)
GCR5_rec2	1125 (0+75)	261 (3+14)	147 (6+5)
GCR5_rprec1MG	324 (0+6)	204 (3+3)	132 (6+1)

five implicit stages of the ESDIRK method. It can be seen that convergence deteriorates significantly for the stiffer case with the high Reynolds number and the high aspect ratio grid. For the same non-dimensional time step of 0.5 and the same iterative tolerance of 0.005 for the turbulent case 193 FAS iterations are required compared to 50 for the laminar case. Possible speed-up of the iterative solution technique for both the laminar and turbulent case using the Jacobian-free Newton–Krylov method are subsequently investigated.

In the rows of Table I the work, measured in residual evaluations, is shown for the first stage of the first time step (later time steps showed similar results) for various combinations of iterative solution techniques for laminar flow with a time step of 0.5. The columns show after how many original non-linear MG cycles the iterative solver is switched on. Between brackets the total number of MG cycles and Newton cycles are listed. For this case unpreconditioned GCR5-20, which uses 5 vectors in the first Newton and 20 at all the rest, and GCR5 switched on after 6 MG cycles are the best, leading to a speed-up of 30% compared to the original solver. As a reference: use of full GCR, which is efficiency-wise optimal, but impractical due to the immense memory requirements, required 78 residual evaluations. The two forms of preconditioning: 1 cycle of FAS MG or 2 recursive iterations of GCR decreased the number of Newton iterations, but at increased overall costs.

The same trends are found for the turbulent flow case with a time step of 0.5 shown in Table II. Although for this harder case GCR5-20 outperforms GCR5. Again the two types of preconditioning are too expensive to lead to an overall performance improvement.

Table II. Work for turbulent flow problem using various iterative solvers.

	7 MG first	10 MG first
FAS MG	276 (23+0)	276 (23+0)
GCR5-20	349 (7+12)	204 (10+5)
GCR5	399 (7+63)	210 (10+18)
GCR5_rec2	594 (7+34)	230 (10+8)

4. CONCLUSIONS

It can be concluded that with a minimal coding effort efficiency can be increased up to 30% combining the original non-linear FAS MG and a Jacobian-free Newton–GCR method. However, for this the methods should not be nested but be applied consecutively. In the first iterations the convergence ratio of FAS MG is hard to beat, therefore FAS MG should be used. At some point, when the convergence rate has dropped, GCR should be switched on. Preconditioning by the full FAS MG scheme or by applying GCR recursively does not lead to a more efficient method.

There are still issues for further research. To begin with, the indicator for the switch from MG to Krylov should be further investigated. Possibilities are absolute or relative residual or absolute or relative convergence rates. Secondly, less expensive preconditioners should be investigated. Finally, a full automatic convergence controller should be designed, which performs the optimization of the iterative solution technique without any interference of the user.

ACKNOWLEDGEMENTS

The authors would like to thank D. J. Mavriplis and C. Vuik for their helpful comments.

REFERENCES

1. Badcock KJ, Cantariti F, Hawkins I, Woodgate M, Dubuc L, Richards BE. Simulation of unsteady turbulent flows around moving airfoils using the pseudo-time method. *International Journal for Numerical Methods in Fluids* 2000; **32**(5):585–601.
2. Bijl H, Carpenter MH, Vatsa VN, Kennedy CA. Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flows. *Journal of Computational Physics* 2002; **179**:313–329.
3. Knoll DA, Keyes DE. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics* 2004; **193**:357–397.
4. Mavriplis DJ. An assessment of linear versus non-linear multigrid methods for unstructured mesh solvers. *Journal of Computational Physics* 2002; **175**:302–325.
5. Jothiprasad G, Mavriplis DJ, Caughey DA. Higher-order time integration schemes for the unsteady Navier–Stokes equations on unstructured grids. *Journal of Computational Physics* 2003; **191**(2):542–566.
6. Saad Y. A flexible inner–outer preconditioned GMRES algorithm. *SIAM Journal on Scientific and Statistical Computing* 1993; **14**:461–469.
7. Van der Vorst HA, Vuik C. GMRESR: a family of nested GMRES methods. *Numerical Linear Algebra with Applications* 1994; **1**(4):369–386.
8. Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 92-439*, 1992.
9. Vatsa VN, Wedan BW. Development of a multigrid code for the 3D Navier–Stokes equations and its application to a grid-refinement study. *Computers and Fluids* 1990.