NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2010; **17**:307–324
Published online 10 December 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla.687

Algebraic multigrid within defect correction for the linearized Euler equations

Anna Naumovich[1,*,†], Malte Förster[2] and Richard Dwight[1,3]

[1]*German Aerospace Center, Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7, D-38108 Braunschweig, Germany*
[2]*Fraunhofer Institute for Algorithms and Scientific Computing, Schloss Birlinghoven, D-53754 Sankt Augustin, Germany*
[3]*Aerodynamics Group, TU Delft, Kluyverweg 2, 2629HT Delft, The Netherlands*

SUMMARY

Given the continued difficulty of developing geometric multigrid methods that provide robust convergence for unstructured discretizations of compressible flow problems in aerodynamics, we turn to algebraic multigrid (AMG) as an alternative with the potential to automatically deal with arbitrary sources of stiffness on unstructured grids. We show here that AMG methods are able to solve linear problems associated with first-order discretizations of the compressible Euler equations extremely rapidly. In order to solve the linear problems resulting from second-order discretizations that are of practical interest, we employ AMG applied to the first-order system within a defect correction iteration. It is demonstrated on two- and three-dimensional test cases in a range of flow regimes (sub-, trans- and supersonic) that the described method converges rapidly and robustly. Copyright © 2009 John Wiley & Sons, Ltd.

Received 15 May 2009; Revised 28 October 2009; Accepted 2 November 2009

KEY WORDS: defect correction; algebraic multigrid; linearized compressible Euler equations; computational fluid dynamics

1. INTRODUCTION

Rapid and robust solution of the equations of fluid flow is of the utmost practical concern within the aerospace engineering community, where the demand for accurate flow solutions is growing rapidly due to an increase in confidence in computational fluid dynamics (CFD).

*Correspondence to: Anna Naumovich, German Aerospace Center, Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7, D-38108 Braunschweig, Germany.
†E-mail: anna.naumovich@dlr.de, anna.nmv@gmail.com

Contract/grant sponsor: European Regional Development Fund, Economic Development Fund of the Federal German State of Lower Saxony; contract/grant number: W3-80026826

In most of today's production Euler and Navier–Stokes codes, the acceleration methods used for convergence to a steady state strongly resemble methods proposed in the early 80s by Jameson and co-workers [1–3]. They employ geometric multigrid (GMG) smoothed with either explicit Runge–Kutta methods, or an implicit scheme with some approximate Jacobian [4–6]. Despite the investigation of many variations on this theme over the years [6–10], satisfactory grid-insensitive convergence has not been achieved for the Navier–Stokes equations. For two-dimensional single-element aerofoil geometries and high-quality conformal structured grids, rapid grid independent convergence *is* possible by employing either line-relaxation or directional coarsening in the direction normal to solid walls (where boundary layers are present). However this approach fails even for finite three-dimensional wings, as wall-normal lines fail to capture all stiff modes of the problem (such as wing-tip vortex shedding) [11, 12].

For this reason we are led to consider algebraic multigrid (AMG) methods that are potentially able to automatically capture stiffness on arbitrary grids, and simultaneously offer grid independent convergence.

Nowadays AMG is rarely used for compressible CFD. Raw [13] and Weiss *et al*. [14] are two of the very few examples of the use of AMG in finite volume discretizations of the compressible flow equations.

In this paper the linearized (discrete) inviscid equations are investigated, as a preliminary step toward developing algorithms for the non-linear viscous equations. This allows us to apply AMG directly, avoid start-up issues, perform linear stability analysis and work entirely with linear operators, while preserving the numerical character of the problem. Using the fact that the asymptotic convergence rate of any fixed-point iteration (FPI), when applied to the discretized non-linear problem, will be identical to that of the same iteration applied to the linearized discrete problem [15], we can make direct inferences about the asymptotic performance of the methods in the non-linear case. In addition the linearized equations are themselves of importance for evaluation of solution derivatives, *a posteriori* error estimation and frequency-domain methods for periodic flows [16–18].

The view is taken that the spatial discretization is pre-specified, and the objective is to reduce the time required to obtain steady-state convergence for this particular discretization. We consider a second-order accurate finite volume method on unstructured grids, with either upwind or central convective fluxes, as implemented in the DLR *TAU*-code [4], a commercial aerodynamics solver, extensively used in the European aerospace industry.

Our initial investigations attempted to apply conventional AMG methods for systems, using the SAMG package [19], to the problem resulting from the linearization of these (second-order) discretizations. It was seen that this fails for almost all test cases, often already at the coarsening stage. On the other hand, applying the same AMG solvers to the corresponding first-order discretizations was seen to give excellent convergence provided the coarsening was based on an appropriately chosen sub-matrix. However, a first-order discretization is insufficiently accurate to be useful in practice. This suggests applying AMG within a defect correction (DC) iteration, solving a first-order accurate problem at each step and iterating toward the solution of the second-order accurate problem.

This algorithm is examined for sub-, trans- and supersonic flow regimes, and is seen to significantly out-perform existing GMG methods in all cases. This is by virtue of the efficiency of AMG, which requires only 1–2 iterations to give sufficient accuracy of the inner problem at each DC step. Furthermore as the first-order matrix is constant, only one AMG setup procedure (consisting of

DOI: 10.1002/nla

coarsening, construction of prolongation, restriction and coarse grid operators) must be performed for the entire calculation.

In Section 2 we state the non-linear governing equations and spatial discretization, and describe how the linear problem is formed. Sections 3 and 4, respectively, discuss the particulars of DC and AMG as applied to this problem. Numerical results are presented in Section 5 using the existing GMG method [4, 6] as a reference.

2. GOVERNING EQUATIONS AND SPATIAL DISCRETIZATION

In this paper we are not concerned with solving the flow equations themselves, but with the problem obtained by linearizing a discretization of these equations. The easiest way to present this linear problem is by describing the discretization of the *non-linear* equations on which it is based. As already mentioned, provided that the linearization is exact, the asymptotic convergence rate of the non-linear problem and of the linear problem will coincide for a given FPI.

2.1. Compressible Euler equations

The two-dimensional compressible Euler equations may be written

$$\frac{\partial W}{\partial t}+\nabla\cdot f^c(W)=0, \tag{1}$$

where the conservative state vector is defined by $W=(\rho,\rho u,\rho v,\rho E)^{\mathrm{T}}$, where ρ is density, $U=(u,v)^{\mathrm{T}}$ is the velocity vector and E is total energy. The vectors of convective fluxes are given by

$$f_x^c=\begin{bmatrix}\rho u\\ \rho uu+p\\ \rho uv\\ \rho Hu\end{bmatrix},\quad f_y^c=\begin{bmatrix}\rho v\\ \rho vu\\ \rho vv+p\\ \rho Hv\end{bmatrix}, \tag{2}$$

where p is pressure, and $H=E+p/\rho$ is the total enthalpy. The equation of state for a calorically perfect gas

$$p=(\gamma-1)\rho(E-\tfrac{1}{2}U^2),$$

where γ is the gas dependent ratio of specific heats, closes the system.

2.2. Discretization of the flow equations

The flow equations (1) are discretized separately in space and time. The spatial discretization is performed on unstructured grids of non-overlapping dual-cells of a hybrid primary grid, the so-called cell-vertex metric. Following the standard procedure of finite volume method, integrating over a control volume Ω_i and applying the divergence theorem, we have

$$\int_{\Omega_i}\frac{\partial W}{\partial t}\,\mathrm{d}\Omega_i+\int_{\partial\Omega_i}f^c(W)\cdot n\,\mathrm{d}(\partial\Omega_i)=0, \tag{3}$$

DOI: 10.1002/nla

where $\partial\Omega_i$ is the boundary of Ω_i, and n is outward-facing normal vector on $\partial\Omega_i$. The values of the flow variables are approximated as either constant or affine on a control volume, depending on the particular scheme. The exact convective fluxes $f^c(W)$ are approximated by numerical fluxes $\hat{f}^c(W_L, W_R; n)$, where W_L and W_R represent the reconstructed left and right states at the cell boundaries.

Given that the quantity W_i refers to the solution at the barycentre of the faceted control volume Ω_i, and that the fluxes are evaluated at the barycentre of the faces, then the integrals of (3) may be written

$$|\Omega_i|\frac{\partial W_i}{\partial t} = -\sum_{j\in\mathcal{N}(i)} \hat{f}^c(W_i, W_j; n_{ij}) - \sum_{j\in\mathcal{B}(i)} \hat{f}^c_b(W_i; n_j) =: -R_i(W), \tag{4}$$

where $\mathcal{N}(i)$ is the set of neighbors of Ω_i, $\mathcal{B}(i)$ the set of boundary faces, $\hat{f}_b$ a numerical boundary flux, and R therefore represents the full spatial residual including boundary conditions.

In this work we will be concerned with both first- and second-order discretizations, which may be achieved with either upwind or central schemes. In the case of upwind schemes, $\hat{f}^c$ is always an upwind flux such as the Roe flux [20] or the van Leer flux [21], which are explicit functions of their arguments. The order of accuracy of the method is determined by the order of reconstruction of the solution within cells. If the solution reconstruction is constant, i.e. $W_L := W_i$, $W_R := W_j$, then these methods are first-order accurate, if the solution is linearly reconstructed second-order may be achieved.

For the central Jameson–Schmidt–Turkel (JST) scheme [1], constant solution reconstruction is always used, and $\hat{f}^c$ is constructed as an average of the exact fluxes at $W_L \equiv W_i$ and $W_R \equiv W_j$, plus an artificial dissipation term which penalizes discontinuities and adds background dissipation for stability. The central difference on the exact flux is second-order, so the form of the artificial dissipation term determines the order of the scheme. In the JST scheme this term is chosen as a combination of a first-difference and a third-difference of the solution, weighted such that the latter is inactive near solution discontinuities (where it would otherwise lead to oscillations). The weighting is also chosen such that in smooth regions both terms are of order h^2 in the mesh spacing—so that overall the scheme is second-order accurate. There is no completely natural first-order equivalent of this scheme, but one option is to simply remove the weighting and the third-difference. The resulting scheme is identical to the first-order upwind Lax–Friedrichs scheme, which is known to have the highest possible level of artificial dissipation of any stable upwind scheme.

2.3. Derivation of the linear problem

The solution of solely the *linearized* discrete equations is investigated in this paper. These equations arise e.g. when considering the evaluation of derivatives of some functional of the flow solution $J(W, \alpha)$ with respect to some parameter α, subject to the constraint $R(W, \alpha) = 0$, i.e. that W satisfies the discrete (non-linear) governing equations.

Applying the chain rule to $\mathrm{d}J/\mathrm{d}\alpha$ we obtain

$$\frac{\mathrm{d}J}{\mathrm{d}\alpha} = \frac{\partial J}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}\alpha} + \frac{\partial J}{\partial \alpha},$$

which is an expression for $\mathrm{d}J/\mathrm{d}\alpha$ in terms of $\mathrm{d}W/\mathrm{d}\alpha$. The remaining partial derivative are obtainable explicitly by differentiating (the discrete expression for) J.

DOI: 10.1002/nla

By considering that $R(W,\alpha)\equiv 0$, $\forall\alpha$, and therefore $\mathrm{d}R/\mathrm{d}\alpha\equiv 0$, we have

$$\frac{\mathrm{d}R}{\mathrm{d}\alpha}=\frac{\partial R}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}\alpha}+\frac{\partial R}{\partial \alpha}=0,$$

or

$$\frac{\partial R}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}\alpha}=-\frac{\partial R}{\partial \alpha}, \tag{5}$$

a linear system for $\mathrm{d}W/\mathrm{d}\alpha$. The character of the system matrix depends on the choice of discretization used in R—in particular on the choice of first- or second-order accuracy.

Given the solution of (5), one can evaluate derivatives of scalar *cost functions* (e.g. lift and drag coefficients) with respect to the design variables according to:

$$\frac{\mathrm{d}C_L}{\mathrm{d}\alpha}=\frac{\partial C_L}{\partial \alpha}+\frac{\partial C_L}{\partial W}\frac{\mathrm{d}W}{\mathrm{d}\alpha}. \tag{6}$$

In the following we develop iterative methods for the solution of the linear system (5). As mentioned in the introduction, this is seen primarily as a preliminary step for developing methods for the non-linear problem $R(W)=0$—although the linear problem itself is also of practical importance. It is not the authors' intention to construct a Newton method for $R(W)=0$, which previous studies have shown is unlikely to be effective [6]. Rather we employ the result that the asymptotic convergence rate of a FPI applied to $R(W)=0$ will be identical to that of the same iteration applied to (5) [15]. So if an efficient method can be developed in the more manageable linear case, there is a reason to believe it will be effective in a non-linear setting also.

3. DEFECT CORRECTION METHOD

3.1. Description of the algorithm

The goal of this work is the solution of linear systems resulting from *second*-order accurate discretizations. First-order discretizations are not useful in practice because of the extremely fine meshes necessary to keep numerical dissipation in boundary layers and other flow features at acceptable levels. However second-order discretizations lead to stiff linear systems with poor diagonal dominance. Construction of a suitable smoother for these schemes is problematic. As an example, a standard Gauss–Seidel iteration diverges in almost all cases, and standard solution methods involve addition of a pseudo-timestep and iteration to steady state. Hence the application of multigrid methods is difficult to make efficient.

An alternative approach to solving second-order accurate problems is the DC method [22, 23] which employs an auxiliary first-order accurate discretization of the same continuous problem.

If problem (5) based on second-order accurate R is written in a compact way as

$$A_{(2)}w=f_{(2)}, \tag{7}$$

and the corresponding problem based on a first-order accurate R is written

$$A_{(1)}w=f_{(1)}, \tag{8}$$

DOI: 10.1002/nla

then the DC approach for the solution of (7) may be written

$$w^{(0)} := A_{(1)}^{-1} f_{(1)}, \quad A_{(1)} w^{(n+1)} = \tilde{f}^{(n)}, \quad n = 0, 1, \ldots, \tag{9}$$

where the 'corrected' right-hand side $\tilde{f}^{(n)}$ is

$$\tilde{f}^{(n)} := f_{(2)} - [A_{(2)} w^{(n)} - A_{(1)} w^{(n)}]. \tag{10}$$

Note that in (9) and (10) the second-order target discretization is only used to evaluate the right-hand side, and only the first-order system must be solved at each iteration with the chosen numerical solver. In this way direct solution of the problematic second-order discretization is avoided.

This is of great benefit because the first-order accurate system is usually comparatively well-conditioned, sparser, and as a rule efficient solution is possible with multigrid methods. As a single first-order problem has to be solved at each iteration of the DC method, the efficiency of solution of the first-order system is crucial to the performance of the algorithm as a whole. In this work, we propose an AMG solver for this purpose.

The spectral radius of the iteration matrix of the DC method, $\rho(I - A_{(1)}^{-1} A_{(2)})$ determines the asymptotic convergence of the algorithm. Roughly speaking, the further the spectral radius is bounded away from one (the better $A_{(1)}$ approximates $A_{(2)}$) the faster the asymptotic convergence of the method.

3.2. The choice of first-order operators

In order to apply the DC method, a suitable first-order operator should be defined for each second-order discretization.

For the second-order upwind Roe and van Leer schemes, the corresponding first-order upwind schemes, as defined in Section 2, are a natural choice.

In the case of the central JST scheme, such a natural first-order equivalent does not exist. However one can define a corresponding first-order scheme, e.g. by dropping completely the term corresponding to the third-order dissipation and by prescribing the coefficient of the first-order dissipation to be equal to one. The first-order method thus obtained is usually relatively easy to solve with standard iteration techniques. Correspondingly it has a very large amount of numerical diffusion, and it may be the case that—especially in test cases with discontinuities (which are significantly smoothed by the first-order schemes)—convergence of the resulting DC method would be rather poor. The root of this is a mismatch between the low- and high-order operators, the (second order) JST scheme being the *least* dissipative method considered here. Therefore, in this work we also employ the first-order van Leer operator in DC iterations for the JST scheme. The resulting method is seen to have markedly superior convergence, but needs stabilization with outer iterations of a Krylov method for some test cases.

4. ALGEBRAIC MULTIGRID

4.1. Basic ideas of AMG

By AMG methods we refer to a class of multilevel solvers which do not explicitly use geometric information present in the problem. This independence of the methods from the geometry of

DOI: 10.1002/nla

underlying grids makes them particularly attractive for problems which are defined in complex domains and/or discretized on unstructured grids. The basic ideas of AMG were first introduced in the early 1980's by Brandt, McCormick, Ruge and Stüben [24–27].

As in the case of GMG, the algebraic multigrid algorithm relies on the complementary solution correction properties of smoothing and coarse grid correction. However, AMG operates using a different definition of smoothness.

The error is defined to be algebraically smooth if it is slow to converge under the chosen smoothing procedure. The coarse levels are constructed with respect to this smoothing procedure, complementing it in such a way that error components of *all* frequencies are rapidly reduced. The coarse-grid hierarchy as well as interpolation operators are defined purely algebraically and rely solely on the system matrix, whereas restriction operators and coarse grid operators are defined in a *Galerkin* manner (see e.g. [27]).

The AMG algorithm consists of two phases. The phase during which coarse levels, intergrid transfer operators and coarse grid operators are constructed is called the setup phase. The setup phase is followed by the solution phase, where the constructed AMG components are employed in a standard multigrid cycle.

AMG was originally developed for the solution of discretized elliptic scalar PDEs, for which it is known to be very robust and efficient. The matrices of these discretizations are often symmetric and positive definite (or close to SPD), have all positive diagonal and all negative off-diagonal entries, *and* are (weakly) diagonally dominant. Extensions of classical AMG are required in order to solve efficiently non-elliptic PDEs and systems of PDEs. Matrices which come from discretizations of such problems are usually far from SPD. The appearance of large (both positive and negative) off-diagonal entries and significant violation of diagonal dominance typically causes problems for AMG. Special extensions of AMG method are necessary to treat such problems efficiently. Besides this, for the efficiency of AMG for PDE systems it is of practical importance that the solver distinguishes physical unknowns in the system.

One should mention that for complex problems a combination of an AMG method with a Krylov accelerator often improves the performance of stand-alone AMG considerably, resulting in a method which more efficiently reduces the whole spectrum of error frequencies.

4.2. Coupled AMG approach for systems of PDEs

In this work, we employ a *point-based* approach to the treatment of systems of PDEs (see e.g. [27, 28]), using the SAMG package [19]. The approach uses the terminology of *variables*, *unknowns* and *points*. *Variables* in this context are all entries of solution vector u, i.e. all degrees of freedom of the system. *Unknowns* are groups of variables corresponding to the same physical quantity in the underlying system of PDEs, e.g. density, velocity. Finally, *points* are blocks of all unknowns corresponding to a single point in physical space—typically a grid point.

In a point-based approach, coarsening is performed on the level of points, and the same coarse grid hierarchy is used for each physical unknown. In this approach coarsening is based on a (scalar) primary matrix, $P=(p_{ij})_{i,j}$ which represents the weighted point-to-point connectivity of a problem. In practice, there are two ways to choose a primary matrix. One is to pass a user-defined matrix to the AMG setup phase, based for example on geometric considerations. The other way is to define P in terms of the original matrix A. We pursue the latter approach.

DOI: 10.1002/nla

If we assume A to be ordered *point-wise*, we can write it in the form

$$A = \begin{pmatrix} A_{(1,1)} & A_{(1,2)} & \dots & A_{(1,n)} \\ A_{(2,1)} & A_{(2,2)} & \dots & A_{(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ A_{(n,1)} & A_{(n,2)} & \dots & A_{(n,n)} \end{pmatrix} \tag{11}$$

with n being the number of points and $A_{(i,j)}$ being blocks connecting all physical unknowns between points i and j. For each of these point coupling matrices we define a scalar 'coupling' value p_{ij} to describe the strength of the connection between the respective point pairs.

For the matrices which come from the first-order accurate discretizations discussed so far, experience shows that the most efficient scheme results from defining p_{ij} via the x-momentum component. We would like to acknowledge that this is a primarily empirical choice, based on observing the performance of the iteration for different choices of P. However in the test cases considered the far-field flow is directed in the x-direction, so we use the submatrix consisting of the ρu contribution to the momentum conservation equation in the x-direction. In this way, we choose P as a respective submatrix of A which contains only coupling data within the chosen unknown.

One should mention that even in case of a first-order discretization, the system matrix A as a whole is off-diagonally dominant for all of the schemes considered here. However, the submatrices which correspond to the couplings within a single unknown are often rather close to M-matrices. This is due to the influence of the large dissipation term in all the first-order methods. If this term were considered alone its discretization would almost give an M-matrix, and would only weakly couple different unknowns.

In any case, if one defines P using one of these diagonal submatrices of A, the following properties are inherited by P: it has positive diagonal entries, and mostly negative off-diagonal entries (with few exceptions). Moreover, $p_{ii} \approx \sum_{j \neq i} |p_{ij}|$ with deviations small relative to the size of the diagonal elements. Exceptions may be found in the matrices coming from the upwind Roe scheme, where some large positive off-diagonal entries appear in our studies—specifically in supersonic regime, on finer grids. This may slow down converge of AMG, but provided there are not too many such entries, it does not appear to be critical to the stability of the method.

Defined in this way the matrix P makes a good candidate for definition of coarse levels and interpolation. Moreover, as the other diagonal unknown-to-unknown blocks all have the same connectivity structure, it makes sense to expand this coarsening and interpolation for all of the physical unknowns. As the chosen P is close to an M-matrix, both standard AMG coarsening and standard AMG interpolation algorithms (see e.g. [29]) can be applied. The interpolation algorithm based on P defines an interpolation matrix $\hat{I}_H^h$, which is then used to interpolate separately each unknown of the system in the same way. Hence, the full interpolation operator (ordered *unknown-wise*) is a block-diagonal operator and is defined as

$$I_H^h = \mathrm{diag}\{\hat{I}_H^h, \dots, \hat{I}_H^h\}. \tag{12}$$

The coarse level operators are defined using the Galerkin formula, $A^H = I_h^H A^h I_H^h$, where the restriction operator is the transpose of the interpolation operator, $I_h^H = (I_H^h)^{\mathrm{T}}$.

DOI: 10.1002/nla

In order to reduce the memory requirement of the AMG algorithm, on finer levels we employ the aggregation type AMG described in [29, 30]. Used in combination with piecewise-constant interpolation, this approach leads to a very simple Galerkin operator of the form

$$A_H = I_h^H A_h I_H^h = (a_{kl}^H) \quad \text{with } a_{kl}^H = \sum_{i \in I_k} \sum_{j \in I_l} a_{ij}^h, \tag{13}$$

where I_k and I_l are the sets of all variables contained in aggregates with indices k and l, respectively.

Application of the aggregation-based approach significantly reduces costs of both the set-up phase and of the AMG cycle. However, in general this has a negative influence on the convergence rates. In fact, the resulting method loses the h-independency of standard AMG in our numerical tests, but nevertheless turns out to be very efficient in terms of CPU time and cheap in terms of memory. As a compromise we only apply aggregative AMG on the first two levels and continue with classical coarsening on coarser levels.

Owing to an overall off-diagonal dominance in the matrices considered, the smoothing properties of Gauss–Seidel—traditionally used as an AMG smoother (both variable-wise and point-wise)—are not satisfactory for the considered class of problems. On the other hand, line-relaxation versions of the Gauss–Seidel method, which usually work well in convection-dominated problems, are difficult to apply in a purely algebraic context. At the same time, it is well-known that smoothing properties of incomplete lower-upper (ILU) factorization methods are good for a wide class of problems. In this work, we employ the simplest version of ILU-type smoothers—variable-wise ILU(0).

5. NUMERICAL EXPERIMENTS

In order to evaluate the performance of AMG and DC independently, the following metrics were investigated for two-dimensional inviscid flow test cases:

1. The performance of AMG on first-order accurate discretizations.
2. The convergence of DC, including the dependence of its convergence rate on the accuracy to which the inner first-order problems are solved.
3. Acceleration of convergence of the DC method for the central discretization with alternative first-order operators.

To demonstrate further generality of the results both DC and AMG were applied to an inviscid three-dimensional test case. Finally, the performance of the suggested DC algorithm was compared with the performance of the linear geometric MG from the DLR TAU-code [4, 6].

In numerical experiments, the three flux discretizations of the Euler equations described so far were considered, namely the central JST scheme, and upwind Roe, and van Leer schemes. These discretizations and their corresponding linearizations were implemented in the DLR TAU-code.

In the two-dimensional case, numerical experiments were performed on a sequence of uniformly refined unstructured NACA0012 grids (with 1140, 5516, 21 854, and 86 996 points, respectively) for subsonic ($M_\infty = 0.5$), transonic ($M_\infty = 0.81$) and supersonic ($M_\infty = 1.5$) flow regimes at an angle of attack of $\alpha = 1.0$. In the 3d case, tests were performed for an unstructured ONERA M6 wing grid with 108 396 points for transonic regime ($M_\infty = 0.84$) and $\alpha = 3.06$. The coarsest grid used for the two-dimensional tests, and the grid used for the three-dimensional tests are depicted in Figure 1.

DOI: 10.1002/nla

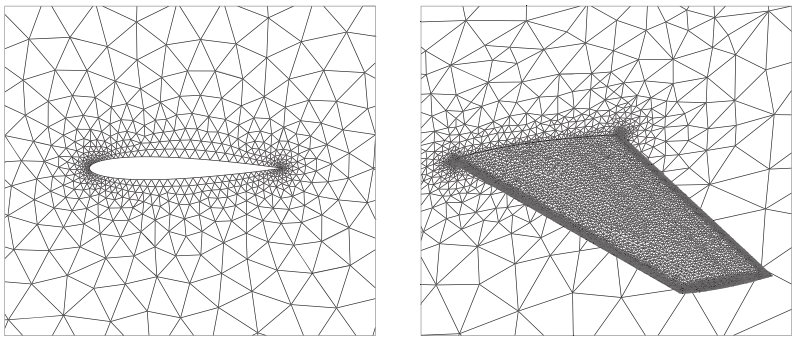

Figure 1. The coarsest NACA0012 airfoil grid (left), ONERA M6 wing grid (right).

In the numerical tests, the L^2-norm of the total residual and the error in the derived quantity of interest $\mathrm{d}C_L/\mathrm{d}\alpha$ were monitored. In the numerical tests evaluating AMG performance on first-order discretizations, the L^2-norm of the total residual of the respective first-order problem was monitored.

In the investigation of the performance of the DC approach, the focus was mainly on the convergence of the solution (as indicated by the reduction of error of the derived quantity of interest $\mathrm{d}C_L/\mathrm{d}\alpha$). This emphasis is justified by the fact that in most practical applications convergence of such solution functionals rather than residual reduction is of principal interest. The quantity $\mathrm{d}C_L/\mathrm{d}\alpha$ was computed at each iteration of the method according to the formula (6) using the transient approximation to the solution $\mathrm{d}W/\mathrm{d}\alpha$. For the completeness of the presentation of the convergence of the DC approach, some results concerning residual reduction of second order-problems are added as well.

5.1. Convergence of AMG applied to first-order discretizations

As at each iteration of DC one first-order problem has to be solved, fast solution of first-order problems is crucial for the efficiency of the algorithm. In this work, first-order problems are solved by the AMG method [19] which is accelerated by bi-conjugate gradient stabilized (BiCGSTAB) [31]. BiCGSTAB was chosen from the family of Krylov methods due to its low memory requirements and a comparable performance with GMRES (generalized minimal residual method) for the class of problems considered here.

The previously described sequence of four grids is employed for the two-dimensional tests. On these grids, three different discretizations as well as three different flow regimes are considered. Convergence results corresponding to these test cases are depicted in Figures 2–4. Note that due to the structure of BiCGSTAB method, the AMG preconditioner must be called twice per iteration of BiCGSTAB, so that each iteration in the figures corresponds to two AMG cycles. Here a V-type cycle was used. The iterations are stopped when the total residual is reduced by five orders of magnitude.

As one can see from Figures 2–4, the convergence of the AMG solver is very rapid, and shows only a weak grid dependence. The solver slows down somewhat on finer grids, most noticeably in transonic and supersonic regimes. Examination of the system matrix shows the appearance of a few large positive off-diagonal entries in these cases—which may be a cause of the problems. Another

DOI: 10.1002/nla

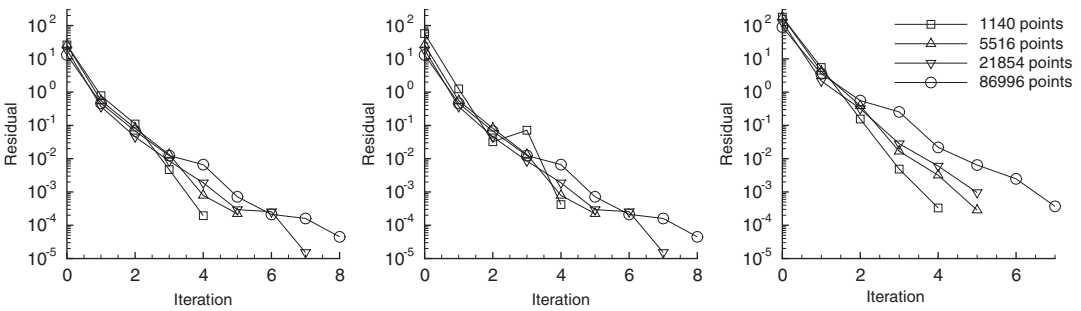

Figure 2. Convergence of AMG-preconditioned BiCGSTAB for the first-order JST central scheme in subsonic (left), transonic (middle), supersonic (right) regimes.

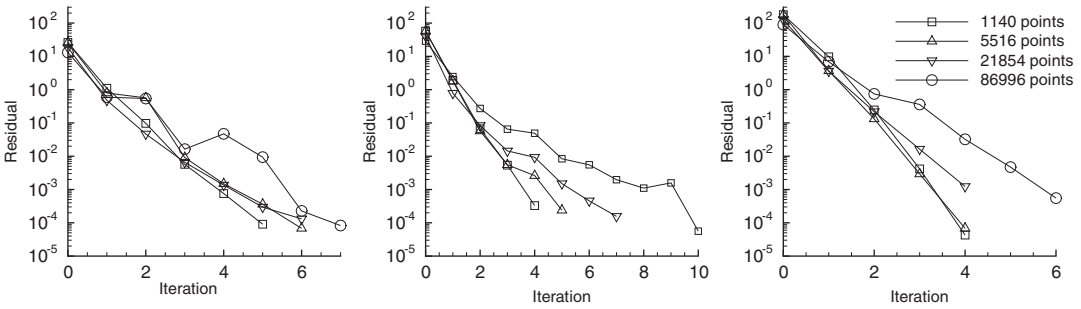

Figure 3. Convergence of AMG-preconditioned BiCGSTAB for the first-order van Leer scheme in subsonic (left), transonic (middle), supersonic (right) regimes.

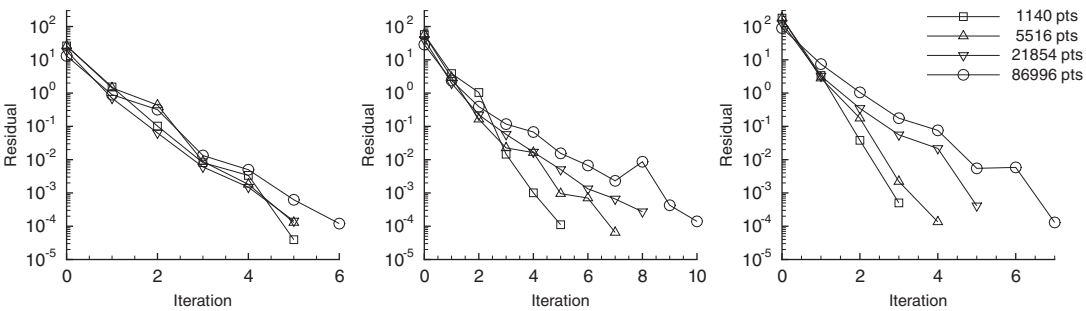

Figure 4. Convergence of AMG-preconditioned BiCGSTAB for the first-order Roe scheme in subsonic (left), transonic (middle), supersonic (right) regimes.

reason for this grid-dependency is the fact that an aggregation-based AMG approach (known to be not absolutely grid-independent) was used rather than the classical one. Note also that a V-cycle was employed to generate the results presented in Figures 2–4. If a W-cycle is used instead, AMG is less grid dependent, but it is more expensive in terms of CPU time.

DOI: 10.1002/nla

Table I. Influence of the threshold ε_1 on the number of DC iterations and CPU time (in brackets, in seconds) required to achieve the prescribed tolerance of $dC_L/d\alpha$: van Leer scheme.

Number of grid points	$\varepsilon_1 = 10^{-1}$	$\varepsilon_1 = 10^{-2}$	$\varepsilon_1 = 10^{-3}$
1140	27 (0.27)	23 (0.52)	23 (0.78)
5516	33 (1.74)	24 (2.25)	23 (3.99)
21 854	38 (7.48)	25 (9.4)	21 (15.25)
86 996	38 (39.3)	21 (40.78)	17 (55.71)

It is worth mentioning that the convergence of the employed AMG is very fast, and by far outperforms the convergence of the standard linear solver employed in DLR TAU-code, applied to the first-order problem. It should be noted however that this solver, an agglomeration-based GMG accelerated with GMRES, was developed for direct application to the second-order discretizations [4, 6], and not tuned for the first-order problems.

5.2. *Convergence of DC*

In this subsection we first investigate the influence of the precision of the solution of the first-order problems on convergence of the DC method in terms of the number of outer iterations, as well as CPU time required. As solution of one first-order problem per DC iteration is required, solution of all to a very high accuracy is expensive. Fortunately experience shows that reduction of the residual of the first-order problems by only one or two orders of magnitude is sufficient. Solving beyond this has no further effect on the convergence of the DC iteration and may be considered over-solving. We demonstrate this on the considered test examples.

Numerical experiments were performed on the sequence of four NACA0012 grids. Residuals of the first-order problems are reduced by at least the prescribed value ε_1 at each DC iteration. The value of ε_1 is varied and its influence on the convergence of the entire defect correction algorithm is investigated. Within AMG a V-type cycle is employed.

We base the stopping criterion for the DC iteration on the convergence of the derivative of the lift coefficient $\mathrm{d}C_L/\mathrm{d}\alpha$, calculated according to (6). Specifically, the iterations are stopped when the error of the transient $dC_L/d\alpha$ with respect to the fully converged solution on the current grid is below a prescribed threshold. For the numerical experiments, we choose this threshold to be $5 \cdot 10^{-5}$ for the subsonic and transonic regimes, and $5 \cdot 10^{-6}$ for the supersonic regime. The smaller threshold for the supersonic regime is chosen due to the smaller absolute value of $\mathrm{d}C_L/\mathrm{d}\alpha$ in this regime.

Results of the experiments for the three different discretizations, van Leer, Roe and central JST schemes for the subsonic regime are summarized in Tables I–III.

From these tables one can see that reduction of residuals of the first-order problem by only one or two orders of magnitude ($\varepsilon_1 = 10^{-1}$ or 10^{-2}) is optimal for fast convergence of DC in terms of CPU time. Similar results hold for the transonic and supersonic cases. In the interests of concise presentation, convergence results of DC for these regimes are presented for $\varepsilon_1 = 10^{-1}$ only (see Table IV).

Although we have no reason to expect grid independence of DC iterations, one can see from Tables I–III that in case of subsonic regime DC is not especially sensitive to a grid size for these cases, especially when more precise solution of the first-order sub-problems is used ($\varepsilon = 10^{-3}$). From Table IV it seems that in non-smooth regimes convergence of DC method is more grid

DOI: 10.1002/nla

Table II. Influence of the threshold ε_1 on the number of defect correction iterations and CPU time (in brackets, in seconds) required to achieve the prescribed tolerance of $dC_L/d\alpha$: Roe scheme.

Number of grid points	$\varepsilon_1 = 10^{-1}$	$\varepsilon_1 = 10^{-2}$	$\varepsilon_1 = 10^{-3}$
1140	19 (0.23)	18 (0.31)	18 (0.45)
5516	26 (1.32)	21 (1.97)	15 (2.27)
21854	29 (5.87)	20 (7.24)	15 (9.69)
86996	30 (32.79)	18 (32.19)	14 (49.39)

Table III. Influence of the threshold ε_1 on the number of DC iterations and CPU time (in brackets, in seconds) required to achieve the prescribed tolerance of $dC_L/d\alpha$: JST scheme.

Number of grid points	$\varepsilon_1 = 10^{-1}$	$\varepsilon_1 = 10^{-2}$	$\varepsilon_1 = 10^{-3}$
1140	55 (0.5)	47 (0.7)	46 (1.0)
5516	59 (2.82)	47 (3.3)	41 (4.9)
21854	64 (11.8)	46 (14.6)	40 (20.9)
86996	86 (64.7)	41 (56.9)	33 (86.6)

Table IV. Number of defect correction iterations based on $\varepsilon_1 = 10^{-1}$, required to achieve the prescribed tolerance of $dC_L/d\alpha$ for the transonic and supersonic regimes.

	Transonic			Supersonic		
Number of grid points	van Leer	Roe	JST	van Leer	Roe	JST
1140	14	17	99	8	7	72
5516	18	22	131	9	8	79
21854	48	35	155	7	8	76
86996	78	47	199	5	12	35

dependent. This dependency remains irrespective of the level to which the first-order problem is solved.

5.3. Special treatment of the JST scheme in DC

From the numerical results presented in Tables I–IV one can see that the convergence of the DC method for the central JST scheme is significantly slower than for the upwind schemes. The most likely reason for this slowness is the fact that the first-order equivalent of the central scheme is too diffusive, and corresponds poorly to the second-order operator. In order to test this hypothesis, and to obtain a faster method for the central discretization, we base defect correction on the less diffusive first-order van Leer operator (we denote this variant as DC2) instead of the first-order central scheme (DC1). However, practical experience shows that DC2 sometimes diverges after a period of fast convergence, suggesting the presence of a minority of unstable modes. Application of an outer Krylov cycle thereby stabilizes the method very efficiently. As the DC employed is not a constant preconditioner (it contains AMG accelerated with BiCGSTAB), we use a flexible

DOI: 10.1002/nla

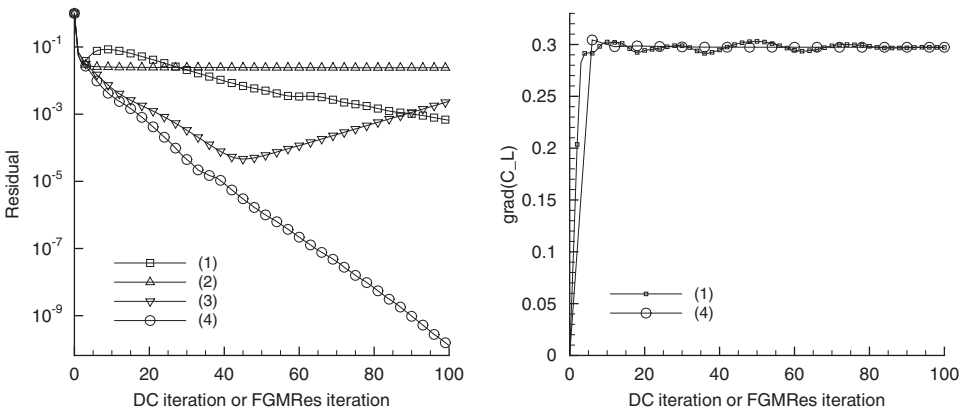

Figure 5. Residual reduction (left) and convergence of $dC_L/d\alpha$ (right) in DC1 stand-alone (1), DC1 accelerated with FGMRES (2), DC2 stand-alone (3) and DC2 accelerated with FGMRES (4).

variant of a Krylov method, namely flexible GMRES (FGMRES), which was introduced by Saad in [32] and can be used with a variable preconditioner.

In Figure 5 convergence of four DC algorithms for the central scheme are presented, namely DC1 and DC2, each stand-alone and accelerated/stabilized with FGMRES. The convergence histories shown are for the finest NACA0012 grid in the transonic case. In the calculations we employed a Krylov space of size 6, one DC iteration was used to precondition FGMRES, and $\varepsilon_1 = 0.1$. As one can see from Figure 5, DC2 accelerated with FGMRES provides the fastest residual reduction and convergence of $\mathrm{d}C_L/\mathrm{d}\alpha$. Another observation is that accelerated version of DC1 results in a stagnating iteration. More preconditioning iterations per FGMRES step help in this case, but the resultant method is not efficient in terms of CPU time. One also observes divergence of non-stabilized DC2.

One should note that stabilization of DC2 with a Krylov method is only required on finer grids. Apparently, on finer grids the iteration matrix of the DC approach, $M = I - A_{(1)}^{-1} A_{(2)}$ has a number of large isolated eigenvalues which lie outside the unit circle, so that the spectral radius of M is greater than one, $\rho(M) > 1$. This causes divergence of the iterations of DC2. Application of a Krylov method as an outer iteration efficiently captures the eigenvectors which correspond to the largest eigenvalues of M and which cause divergence of DC2. Owing to this effect of the Krylov accelerator, the resultant method converges rather fast. On coarser grids, DC2 stand-alone works fast and application of a Krylov accelerator does not have this effect as there are not many large isolated eigenvalues which are responsible for slow convergence. Therefore, on coarse grids the accelerated method has a very similar convergence to the non-accelerated one.

Having identified a performant and robust method, we perform a number of further experiments for FGMRES accelerated DC2, for the NACA0012 on all grids in all regimes. As before the Krylov space was of dimension 6, one DC iteration was used to precondition FGMRES, and $\varepsilon_1 = 0.1$. The stopping criterion is based on the convergence of $dC_L/d\alpha$, as in Tables I–IV. Convergence results are summarized in Table V, where the number of FGMRES iterations for each test case is presented. Note that due to the size of the Krylov space, solution is evaluated only every 6 iterations.

DOI: 10.1002/nla

Table V. Number of FGMRES iterations preconditioned by DC based on the first-order van Leer scheme needed to achieve the described stopping criterion, JST scheme, $\varepsilon_1 = 0.1$.

Number of grid points	Subsonic	Transonic	Supersonic
1140	30	30	18
5516	24	36	12
21 854	30	36	18
86 996	30	42	18

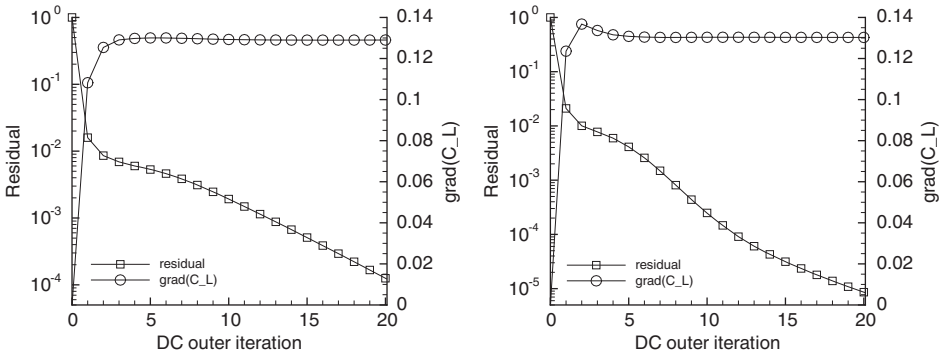

Figure 6. ONERA M6 wing test case: Convergence of DC ($\varepsilon_1 = 10^{-1}$) for the JST scheme (left) and for the Roe scheme (right) in transonic regime.

From Table V it is clear that DC2 accelerated with FGMRES has much faster convergence than DC1 (as compared with the results in Tables I–IV) for the JST scheme. One can also see that convergence of the method is not noticeably sensitive to grid size.

5.4. Performance of AMG and DC for a three-dimensional test case

Finally we present some results concerning the performance of AMG and DC for a three-dimensional example, namely transonic flow ($M_\infty = 0.84$) over an ONERA M6 wing (Figure 1). A grid with 108 396 points is employed, and two discretizations—the JST scheme and the Roe scheme are considered.

Convergence of the AMG method for the first-order problems is fast—reduction of the residual by around 0.1 or 0.2 in one iteration of BiCGSTAB is achieved in the cases where preconditioner is a W-cycle or a V-cycle of AMG respectively.

Convergence of the DC iteration is shown in Figure 6. Note that for the JST scheme we employed DC1 stand-alone, which performed well in this case.

5.5. Residual reduction by DC method

In the previous subsections we focused on the performance of DC as indicated by its reduction of the solution error. Here we show that this approach provides a satisfactory reduction of the residuals of the second-order problem as well.

DOI: 10.1002/nla

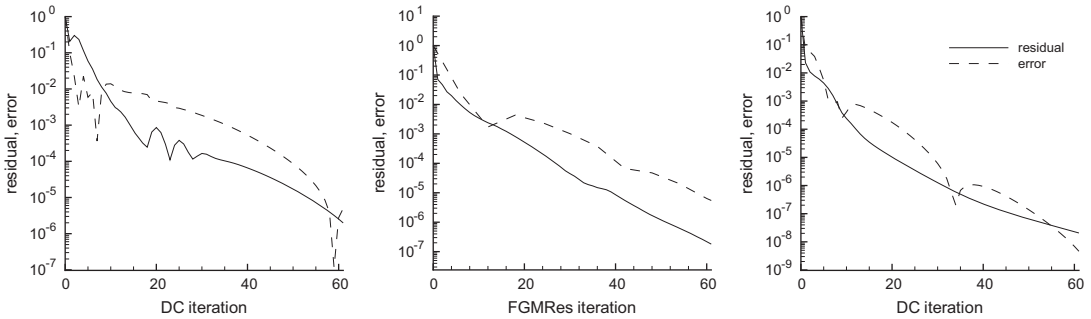

Figure 7. Reduction of residual and solution error for the following test cases: two-dimensional NACA0012 grid with 86 996 points, transonic regime, Roe scheme (left) and Central scheme (middle), and the three-dimensional test case, Roe scheme (right).

In Figure 7 we present reduction of the residuals and of the errors of $\mathrm{d}C_L/\mathrm{d}\alpha$ for three test cases: (1) the two-dimensional NACA0012 grid with 86 996 points, transonic regime, Roe scheme, (2) the same grid and regime, central scheme, approach DC2 accelerated with FGMRES, (3) the three-dimensional test case, described in Subsection 5.4, Roe scheme.

In Figure 7 one sees that not only asymptotically but also within the first iterations, the reduction of the residual as well as of the solution error occurs in general at similar rates (except for some oscillations of the error). This convergence character of the residuals and of the errors is typical for the other test-cases considered as well.

5.6. Comparison with linear GMG

For all of the two- and three-dimensional test cases considered, the convergence of DC approach combined with AMG significantly outperforms convergence of the linear geometric multigrid from the DLR TAU code (applied directly to second-order problems) in terms of CPU time. Speed-up by a factor of about 10 in CPU time for all considered test cases was achieved.

Memory usage of the two methods (in two dimensions) can be related as follows. Suppose that the storage of the finest-level second-order system matrix is 1 memory unit. The first-order matrix on this level has the same dimensions, but is more sparse and requires around $1/3$ memory units.

Memory needed for the DC algorithm is then around 1.7 memory units, including the storage of (a) the fine-level second-order matrix (1 memory unit), (b) the fine-level first-order matrix (around $1/3$ memory units), (c) the AMG hierarchy based on the first-order matrix ($1/15$ memory units in the aggregation approach), and (d) the ILU-decomposition of the first-order matrix ($1/3$ memory unit).

In the case of the GMG employed in the DLR TAU solver, the Jacobian matrices are stored on all levels of the grid hierarchy (on all levels except for the finest, first-order Jacobians are employed). This totals about 1.1 memory units necessary for the GMG.

6. CONCLUSIONS

A solution method has been described for the linearized compressible Euler equations, successfully employing AMG in an aerodynamic context. Typical AMG techniques have been found to be

DOI: 10.1002/nla

ineffective when applied directly to second-order discretizations of the governing equations—a consequence of poor diagonal dominance, lack of M-matrix structure and lack of an effective smoother. This is likely the reason that AMG has not been used extensively in aerodynamics up until now. However, it has been shown to give very efficient solution of systems corresponding to first-order discretizations of the equations.

Given this ability to rapidly solve first-order systems, DC methods become very attractive. In particular, because a system with the same left-hand side operator must be solved at each DC iteration, only a single AMG setup-phase is necessary, mitigating what is usually a significant cost of employing AMG. The resulting DC method is extremely efficient in both two and three dimensions, achieving roughly a factor of 10 reduction in CPU time compared with directly employing the GMG, local time-stepping and approximate implicit schemes [6]. Memory requirements remain within reasonable bounds, as the first-order system has significantly less fill-in than the second-order.

We are very optimistic about the potential of this approach. In particular AMG has so far required little tuning to give the rapid convergence shown here. It is expected that the application to viscous problems will present few difficulties, as AMG automatically accounts for boundary-layer anisotropy. Parallelization is unlikely to be problematic.

DC on the other hand may require more attention. It has already been seen that the choice of first-order operator is critical (for JST), and Krylov stabilization is occasionally needed. It may be necessary to construct low-order operators that better approximate their high-order counterparts if the convergence rate is to be maintained for more complex geometries and flows. Nevertheless, the availability of inexpensive low-order solutions makes DC iterations cheap and attractive.

Future work will concern viscous problems with turbulence modeling, as well as the solution of non-linear problems.

REFERENCES

1. Jameson A, Schmidt W, Turkel E. Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. *AIAA Paper Series*, 1981. AIAA-1981-1259.
2. Jameson A, Baker T. Multigrid solution of the Euler equations for aircraft configurations. *AIAA Paper Series*, 1984. AIAA-84-0093.
3. Jameson A, Martinelli L, Grasso F. A multistage multigrid method for the compressible Navier–Stokes equations. *Notes on Numerical Fluid Mechanics* 1987; **18**:123–128.
4. Gerhold T, Galle M, Friedrich O, Evans J. Calculation of complex 3d configurations employing the DLR TAU-Code. *AIAA Paper Series*, 1997. AIAA-97-0167.
5. Mavriplis DJ. On convergence acceleration techniques for unstructured meshes. *ICASE Report*, 1998, No. 98-44.
6. Dwight RP. Efficiency improvements of RANS-based analysis and optimization using implicit and adjoint methods on unstructured grids, School of Mathematics, University of Manchester, 2006.
7. Jameson A, Caughey DA. How many steps are required to solve the Euler equations of steady compressible flow: in search of a fast solution algorithm. *AIAA Paper Series*, 2001. AIAA-2001-2673.
8. Mavriplis DJ. An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers. *Journal of Computational Physics* 2002; **175**(1):302–325.
9. Swanson RC, Turkel E, Rossow CC, Vatsa V. Convergence acceleration for multistage time-stepping schemes. *AIAA Paper Series*, 2006. AIAA-2006-3523.
10. Swanson RC, Turkel E, Rossow CC. Convergence acceleration of Runge–Kutta schemes for solving the Navier–Stokes equations. *Journal of Computational Physics* 2007; **224**(1):365–388.
11. Pierce NA, Giles MB, Jameson A, Martinelli L. Accelerating three-dimensional Navier–Stokes calculations. *AIAA Paper Series*, 1997. AIAA-97-1953.
12. Turkel E, Vasta V, Venkatakrishnan V. Uni-directional implicit acceleration techniques for compressible Navier–Stokes solvers. *AIAA Paper Series*, 1999. AIAA-99-3265.

DOI: 10.1002/nla

13. Raw M. Robustness of coupled algebraic multigrid for the Navier–Stokes equations. *AIAA Paper Series*, 1996. AIAA-1996-0297.
14. Weiss JM, Maruszewski JP, Smith WA. Implicit solution of preconditioned Navier–Stokes equations using algebraic multigrid. *AIAA Journal* 1999; **37**(1):29–36.
15. Giles MB. On the iterative solution of adjoint equations. In *Automatic Differentiation*: *From Simulation to Optimization*, Corliss G, Faure C, Griewank A, Hascoet L, Naumann U (eds). Springer-Verlag, 2001; 145–152.
16. Dwight R, Brezillon J. Efficient and robust algorithms for solution of the adjoint compressible Navier–Stokes equations with applications. *International Journal of Numerical Methods for Heat and Fluid Flow* 2008; **60**(4):365–389.
17. Dwight RP. Goal-oriented mesh adaptation using a dissipation-based error indicator. *International Journal of Numerical Methods for Heat and Fluid Flow* 2007; **56**(8):1193–1200.
18. Dwight RP. Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *International Journal of Numerical Methods for Heat and Fluid Flow* 2008; **227**(5):2845–2863.
19. Stueben K, Clees T. *SAMG Users Manual.* Fraunhofer Institute SCAI. Available from: http://www.scai.fraunhofer.de/samg.
20. Roe PL. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics* 1986; **18**: 337–365.
21. Van Leer B. Flux-vector splitting for the Euler equations. *Proceedings of 8th International Conference on Numerical Methods in Fluid Dynamics*, vol. 170. Springer: Aachen, 1982; 507–512.
22. Boehmer K, Hemker P, Stetter H. The defect correction approach. *Computing Supplement* 1984; **5**:1–32.
23. Koren B. Multigridand defect correction for the steady Navier–Stokes equations. *Journal of Computational Physics* 1990; **87**:25–46.
24. Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations. *Institute for Computational Studies*, POB 1852, Fort Collins, CO, 1982.
25. Stüben K. Algebraic multigrid (AMG): experiences and comparisons. *Applied Mathematics and Computation* 1983; **13**:419–452.
26. Brandt A. Algebraic multigrid theory: the symmetric case. *Applied Mathematics and Computation* 1986; **19**: 23–56.
27. Ruge J, Stüben K. Algebraic multigrid (AMG). In *Multigrid Methods*, McCormick SF (ed.), Frontiers in Applied Mathematics, vol. 3. SIAM: Philadelphia, PA, 1987; 73–130.
28. Clees T. AMG strategies for PDE systems with applications in industrial semiconductor simulation. *Ph.D. Thesis*, Universität zu Köln, 2004.
29. Stüben K. An introduction to algebraic multigrid. In *Multigrid*, Trottenberg U, Oosterlee CW, Schüller A (eds). Academic Press: London, 2001. ISBN: 0-12-701070-X.
30. Vanek P, Mandel J, Brezina M. Algebraic multigrid on unstructured meshes. UCD/CCM Report 34, University of Colorado at Denver, 1994.
31. Van der Vorst HA. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**:631–644.
32. Saad Y. *Iterative Methods for Sparse Linear Systems.* SIAM: Philadelphia, PA, U.S.A., 2003.

DOI: 10.1002/nla