Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 224 (2007) 365–388
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Convergence acceleration of Runge–Kutta schemes for solving the Navier–Stokes equations

R.C. Swanson [a,*], E. Turkel [b], C.-C. Rossow [c]

[a] NASA Langley Research Center, Computational Aerosciences Branch, Mail Stop 128, Hampton, VA 23681-2199, USA
[b] Tel-Aviv University, Department of Mathematics, Tel-Aviv, Israel
[c] DLR, Deutsches Zentrum für Luft- und Raumfahrt, Lilienthalplatz 7, D-38108 Braunschweig, Germany

Received 29 September 2006; received in revised form 26 February 2007; accepted 28 February 2007
Available online 12 March 2007

Abstract

The convergence of a Runge–Kutta (RK) scheme with multigrid is accelerated by preconditioning with a fully implicit operator. With the extended stability of the Runge–Kutta scheme, CFL numbers as high as 1000 can be used. The implicit preconditioner addresses the stiffness in the discrete equations associated with stretched meshes. This RK/implicit scheme is used as a smoother for multigrid. Fourier analysis is applied to determine damping properties. Numerical dissipation operators based on the Roe scheme, a matrix dissipation, and the CUSP scheme are considered in evaluating the RK/implicit scheme. In addition, the effect of the number of RK stages is examined. Both the numerical and computational efficiency of the scheme with the different dissipation operators are discussed. The RK/implicit scheme is used to solve the two-dimensional (2-D) and three-dimensional (3-D) compressible, Reynolds-averaged Navier–Stokes equations. Turbulent flows over an airfoil and wing at subsonic and transonic conditions are computed. The effects of the cell aspect ratio on convergence are investigated for Reynolds numbers between 5.7×10^6 and 100×10^6. It is demonstrated that the implicit preconditioner can reduce the computational time of a well-tuned standard RK scheme by a factor between 4 and 10.

MSC: 65N22; 76H05

Keywords: Navier–Stokes; Runge–Kutta; Implicit preconditioning; Fourier analysis; Multigrid

1. Introduction

Computational fluid dynamics has expanded rapidly in recent years and problems with increasing complexity are being solved. While relatively good computational efficiency has been attained for the Euler equations, there are still significant challenges remaining for the Navier–Stokes equations. As a near term objective one should seek comparable efficiency to that for the Euler equations. A major obstacle in achieving such a goal is

[*] Corresponding author. Tel.: +1 757 864 2235; fax: +1 757 864 8816.
E-mail address: r.c.swanson@nasa.gov (R.C. Swanson).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.02.028

the geometrical stiffness of the discrete Navier–Stokes equations caused by the requirement to adequately resolve viscous boundary layers with an economical distribution of grid points. We are also confronted with the dilemma of improving computational efficiency while minimizing computer storage, especially in 3-D simulations.

One powerful solution strategy for solving large scale problems in fluid dynamics is multigrid [1–4]. The multigrid approach offers the possibility of solving discrete partial differential equations with grid independent convergence rates. Although most of the theory developed for multigrid is for elliptic problems, effective multigrid solvers [5,6] have been constructed for the Euler equations, which are hyperbolic in time. Jameson and Caughey [7] demonstrated that an Euler solution for airfoil flows, converged to the level of the truncation error, could be obtained in 3–5 multigrid cycles. However, this method slowed down considerably for laminar viscous flows with moderate cell aspect ratios. Multigrid methods for hyperbolic problems depend on two elements to accelerate convergence. One element is the smoothing of high-frequency components of the solution error. The choice of an iterative scheme for smoothing is crucial, since multigrid requires a smooth solution error to approximate a fine grid problem on a coarser grid. In addition, the smoother must be effective on the coarser grids, since these grids are responsible for removing the low-frequency error modes that cause slow asymptotic convergence of iterative schemes. The second element for accelerating convergence is the expulsion of errors on the coarse grids, which occurs faster for time-like iterative methods due to the larger time steps permitted on coarser grids.

Many multigrid methods that are currently used for solving the Euler and Navier–Stokes equations rely upon an explicit multistage time stepping scheme for a smoother. Frequently this explicit scheme is augmented with a scalar implicit residual smoothing [8] to extend stability, allowing the use of larger time steps. This combination proved to be quite effective in solving inviscid flow problems. In addition, such schemes have been applied effectively to a wide variety of viscous flow problems in both two and three dimensions [9,10]. However, convergence rates slower than 0.99 are encountered when solving turbulent viscous flows.

For viscous flow problems the anisotropy due to grid cell aspect ratio reduces the effectiveness of the high-frequency damping in certain coordinate directions. There are two principal techniques that can reduce or even eliminate the dramatic slowdown that can occur due to such geometrical stiffness. One approach is semi-coarsening, where coarse grids are generated by coarsening in one direction rather than all directions. Mulder [11] generalized this type of coarsening to treat the flow alignment problem (i.e. vanishing damping in a coordinate direction normal to the flow) and also the cell aspect ratio problem. The primary difficulties with such an approach are programming complexity and increased operation count, especially in three dimensions. In order to reduce the operation count a directional coarsening was considered [12,13]. For example, in a 2-D flow the grid was coarsened only in the direction normal to a solid boundary (sometimes called *j*-line coarsening), resulting in a reduced cell aspect ratio and improved smoothing.

The second technique for reducing geometrical stiffness is to apply an implicit method in the direction of strongest coupling. In two dimensions appropriate line relaxation allows the removal of the adverse effects on convergence due to aspect ratio. Thus, efforts have been made to improve the performance of the implicit residual smoothing used in conjunction with Runge–Kutta schemes. The simple diffusion operator in this implicit process was replaced with a convection operator that includes flux Jacobians [14]. Since approximate factorization was used for the inversion of the implicit operator, there was still a strong limitation on the time step allowed. To reduce the complexity of the operator, as well as to eliminate the factorization error, a directional smoothing was developed [15], where smoothing was performed in only the wall normal direction (i.e. *j*-line smoothing). With this approach the time step was limited by the other coordinate directions. These directional coarsening and smoothing methods have not been widely adopted due to their programming complexity and limited applicability in general block-structured grid formulations. When using unstructured grids, the inherent lack of structure in the grid introduces additional challenges. Nonetheless, Mavriplis [16] successfully combined *j*-line coarsening, *j*-line smoothing, and Jacobi preconditioning with an unstructured grid method to demonstrate cell aspect ratio independent convergence rates for 2-D, turbulent, viscous airfoil flow computations.

The directional methods can significantly mitigate, and when combined appropriately even eliminate, the effects of cell aspect ratio in two dimensions; they are considerably less effective when applied to general 3-D problems [17]. Furthermore, there is still a significant stability restriction (Courant–Friedrichs–Lewy

(CFL) number generally less than 10), which reflects the explicit nature of the foundation Runge–Kutta (RK) scheme. In order to extend the generality of the implicit procedure and significantly augment the stability bound of the RK scheme, Rossow [18] introduced a fully implicit operator instead of a scalar implicit residual smoothing procedure. This RK/implicit scheme requires the computation of the flux Jacobians that appear in the flow equations. To reduce storage Rossow expressed the Jacobians in terms of the Mach number and computed them with each application of the residual smoothing. The implicit operator was approximately inverted with symmetric point Gauss–Seidel iteration. The Roe scheme was used for the dissipation in the implicit operator and the residual function. With the RK/implicit scheme CFL numbers exceeding 100 were attained in turbulent airfoil flow calculations.

In the present work we evaluate the RK/implicit scheme, both with computation and analysis, and extend it to three dimensions. The flexibility of the scheme is investigated by considering the effect of choosing alternative numerical dissipation operators. As a result, we demonstrate that the preconditioned RK scheme can be implemented with similar benefits in a variety of existing multigrid methods with a multistage time stepping scheme as a smoother. The RK/implicit scheme is applied to several airfoil flows, including a transonic case with strong shock/boundary-layer interaction. In addition, the performance of the scheme for Reynolds numbers between 5.7×10^6 and 100×10^6 is considered. At the highest Reynolds number the maximum grid cell aspect ratio exceeds 50,000. To assess the scheme in three dimensions turbulent viscous flow over an ONERA M6 wing is computed. For all the calculations the convergence behavior and computational effort for the scheme are discussed.

2. Governing equations

We consider both the 2-D and 3-D Navier–Stokes equations for compressible flow. Assuming a volume fixed in space and time, the integral form of these equations can be written as

$$\int\int\int_{\mathscr{V}} \frac{\partial \mathbf{W}}{\partial t} \mathrm{d}\mathscr{V} + \int\int_{\mathscr{S}} \mathscr{F} \cdot \mathbf{n}\, \mathrm{d}\mathscr{S} = 0, \tag{2.1}$$

where the symbol ∂ indicates partial differentiation, $\mathbf{W}$ is the state vector of conservative variables, $\mathscr{F}$ is the flux density tensor, and $\mathscr{V}$, $\mathscr{S}$, and $\mathbf{n}$ denote the volume, surface, and outward facing normal of the control volume. One can split the flux density tensor into a convective contribution $\mathscr{F}_\mathrm{c}$ and a viscous contribution $\mathscr{F}_\mathrm{v}$, which are given by

$$\mathscr{F}_\mathrm{c} = \begin{bmatrix} \rho\mathbf{q} \\ \rho u\mathbf{q} + p\mathbf{e}_x \\ \rho v\mathbf{q} + p\mathbf{e}_y \\ \rho w\mathbf{q} + p\mathbf{e}_z \\ \rho H\mathbf{q} \end{bmatrix}, \quad \mathscr{F}_\mathrm{v} = \begin{bmatrix} 0 \\ \bar{\tau} \cdot \mathbf{e}_x \\ \bar{\tau} \cdot \mathbf{e}_y \\ \bar{\tau} \cdot \mathbf{e}_z \\ \bar{\tau} \cdot \mathbf{q} - \mathbf{Q} \end{bmatrix}, \tag{2.2}$$

where $\mathbf{q}$ is the velocity vector with Cartesian components (u, v, w), and the unit vectors $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ are associated with the Cartesian coordinates (x, y, z). The variables ρ, p, H represent density, pressure, and total specific enthalpy, respectively. The stress tensor $\bar{\tau}$ and the heat flux vector $\mathbf{Q}$ are given by

$$\bar{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}, \quad \mathbf{Q} = k \begin{bmatrix} \partial T/\partial x \\ \partial T/\partial y \\ \partial T/\partial z \end{bmatrix} \tag{2.3}$$

with k denoting the coefficient of thermal conductivity and T representing the temperature.

In order to close the system given by Eq. (2.1) we use the equation of state

$$p = \rho RT, \tag{2.4}$$

where R is the specific gas constant.

3. Numerical algorithms

We first briefly describe the standard solution scheme for solving the compressible Navier–Stokes equations that will be used as a reference. Variations of this scheme are considered based on the choice of the numerical dissipation scheme. A principal component of the standard scheme is scalar implicit residual smoothing, which provides additional support for the basic iterative scheme, and thus, allows extended stability. A replacement of this component of the standard scheme is the basis for the alternative scheme that allows dramatically improved convergence rates. This alternative formulation is discussed in the second part of this section.

3.1. RK/standard scheme

There are three basic elements in the standard solution scheme: a multistage time-stepping scheme, implicit residual smoothing, multigrid acceleration. We consider, as in many existing computer codes for flow computations, a five-stage Runge–Kutta (RK) scheme. This scheme can be written as

$$\begin{aligned}
\mathbf{W}^{(0)} &= \mathbf{W}^{n},\\
\mathbf{W}^{(1)} &= \mathbf{W}^{(0)} - \alpha_1 \Delta t \mathbf{R}(\mathbf{W}^{(0)}),\\
&\vdots\\
\mathbf{W}^{(5)} &= \mathbf{W}^{(0)} - \alpha_5 \Delta t \mathbf{R}(\mathbf{W}^{(4)}),\\
\mathbf{W}^{n+1} &= \mathbf{W}^{(5)},
\end{aligned} \tag{3.1}$$

where $\mathbf{R}$ is the vector residual function, Δt is the time step, the superscript n denotes time level, the superscript enclosed in parentheses indicates the RK stage, and the RK coefficients [19] are given by

$$[\alpha_1, \ldots, \alpha_5] = [0.25, 0.1667, 0.375, 0.5, 1.0].$$

For convenience we have omitted the indices of the grid points. The residual function $\mathbf{R}^{(q)}$ is defined by

$$\mathbf{R}^{(q)} = \mathbf{R}(\mathbf{W}^{(q)}) = \frac{1}{\mathscr{V}}\left[\mathscr{L}_{\mathrm{c}}\mathbf{W}^{(q)} + \sum_{r=0}^{q}\gamma_{qr}\mathscr{L}_{\mathrm{v}}\mathbf{W}^{(r)} + \sum_{r=0}^{q}\gamma_{qr}\mathscr{L}_{\mathrm{d}}\mathbf{W}^{(r)}\right] \tag{3.2}$$

with $\sum\gamma_{qr} = 1$ for consistency. The operators $\mathscr{L}_{\mathrm{c}}$, $\mathscr{L}_{\mathrm{v}}$, and $\mathscr{L}_{\mathrm{d}}$ relate to the convection, viscous, and numerical dissipation terms. Central differencing is used to approximate the convective and viscous operators. The coefficients γ_{qr} are the weights of the viscous and dissipation terms on each stage (see Ref. [20]), which are taken to be [1, 0, 0.56, 0, 0.44]. Such a scheme is frequently designated as a RK(5,3) scheme, since it has 5 stages and the dissipation terms are evaluated only at three stages.

To extend the stability of the RK scheme we apply implicit residual smoothing, which is defined by

$$(1-\beta_\xi\delta_\xi^2)(1-\beta_\eta\delta_\eta^2)(1-\beta_\zeta\delta_\zeta^2)\overline{\mathbf{R}}^{(q)} = \mathbf{R}^{(q)}, \tag{3.3}$$

where δ^2 is the standard central difference operator for a diffusion term, and (ξ, η, ζ) are the coordinates of a uniformly spaced computational domain. The parameter β is a local function of the grid aspect ratio. There are several ways to define this function (for examples see [21,20]). After inverting the product operator in Eq. (3.3) we substitute $\overline{\mathbf{R}}^{(q)}$ for $\mathbf{R}^{(q)}$ in Eq. (3.1). For the inversion scalar tridiagonal solves are performed in each coordinate direction.

One can view the implicit residual smoothing as a preconditioner, and the multistage scheme can be viewed as a smoother for the multigrid method. As a smoother the scheme should be designed so that it has good high-frequency damping properties. A Fourier analysis shows that the five-stage RK scheme alone smooths effectively the high-frequency components of the solution error. However, with the addition of implicit residual smoothing, there is significant deterioration in the smoothing behavior of the RK scheme [20]. In evaluating the resulting scheme one must also consider the improved stability of the scheme, which allows faster error propagation in the coarse grid process of multigrid.

In the standard scheme the full approximation scheme (FAS) is applied to the nonlinear system of equations. Consider a fine grid and a sequence of successively coarser grids generated by eliminating every other mesh line in each coordinate direction. Let the index k denote the kth grid. Let I_k^{k-1} be the fine-to-coarse grid restriction operator and I_{k-1}^{k} be the coarse-to-fine grid prolongation operator. If $\overline{\mathbf{W}}_k$ is the current solution on grid k, the residual on this grid is $\mathbf{R}_k \equiv \mathbf{f}_k - \mathscr{L}_k\overline{\mathbf{W}}_k$, where $\mathbf{f}_k$ is a forcing function. This leads to the coarse-grid equation

$$\mathscr{L}_{k-1}\mathbf{W}_{k-1} = \mathbf{f}_{k-1} = -I_k^{k-1}\mathbf{R}_k + \mathscr{L}_{k-1}(I_k^{k-1}\overline{\mathbf{W}}_k). \tag{3.4}$$

After solving the coarse-grid equation for $\mathbf{W}_{k-1}$, the fine-grid solution is corrected by

$$\mathbf{W}_k \leftarrow \overline{\mathbf{W}}_k + I_{k-1}^{k}(\mathbf{W}_{k-1} - I_k^{k-1}\overline{\mathbf{W}}_k). \tag{3.5}$$

Eq. (3.4) is solved by applying the same relaxation procedure that is used to solve the fine-grid equation. On the coarse grids, the second-order approximation of the convective operator, which is used on the fine grid, is reduced to first order. Multigrid is applied recursively to the coarse-grid equation.

The restriction operators for transferring the residual and solution values from a fine grid to a coarse grid are the ones proposed by Jameson [6] for a cell-centered, finite-volume scheme. The residual and solution restriction operators are defined, respectively, by a summing of the residuals and by a volume weighting of the solution values over the fine-grid cells that comprise a coarse-grid cell. Coarse-grid corrections are transferred with a bilinear (2-D) or trilinear (3-D) interpolation operator. A conventional V-cycle or W-cycle is used to execute the multigrid process.

3.2. Discretization and dissipation

Using the finite-volume technique for spatial discretization, Eq. (2.1) can be written in semidiscrete form as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{1}{\mathscr{V}}\sum_{\text{all faces}} \mathbf{F}_\text{n} S = 0, \tag{3.6}$$

where $\mathbf{F}_\text{n}$ is the normal flux density vector at the cell face, now $\mathscr{V}$ represents the volume of a computational cell, and S is the area of a cell face. The convective part of the flux density vector $\mathbf{F}_\text{c}$ can be expressed as

$$\mathbf{F}_\text{c} = \frac{1}{2}(\mathbf{F}^\text{L} + \mathbf{F}^\text{R}) + \mathbf{D}, \tag{3.7}$$

where $\mathbf{F}^\text{L}$ and $\mathbf{F}^\text{R}$ are the left and right states of the inviscid flux density vector normal to the cell interface, and $\mathbf{D}$ is the numerical dissipation. With this flux vector we construct a central difference approximation plus numerical dissipation. Central differencing is used to approximate the physical diffusion terms.

In the present work we consider three different forms for the dissipation. One form comes from Roe's flux difference split scheme [22], and it can be written as

$$D = -\frac{1}{2}|\mathbf{A}|(\mathbf{W}^\text{R} - \mathbf{W}^\text{L}) = -\frac{1}{2}|\mathbf{A}|\Delta\mathbf{W}, \tag{3.8}$$

where $\mathbf{A}$ is the flux Jacobian at a cell face. For this form we use $|\mathbf{A}|\Delta\mathbf{W}$ expressed in terms of the cell interface Mach number M_0, according to Rossow [23]. The Mach number M_0 is given by

$$M_0 = \min(|M|, 1)\text{sign}(M). \tag{3.9}$$

The resulting form for $|\mathbf{A}|\Delta\mathbf{W}$ is given in Appendix A, and the expression for $|A|$ is given in Ref. [24]. For second-order accuracy the symmetric limited positive (SLIP) scheme of Jameson [25] is used following the implementation of Swanson et al. [26].

Another dissipation formulation considered is closely related to that of the Roe scheme. It is generally called matrix dissipation (see [27]). There is one principal difference between the Roe scheme dissipation and the matrix dissipation. The SLIP scheme is replaced by a scalar switching function (i.e. Jameson–Schmidt–Turkel switch [28]), which uses a pressure function to change from third-order dissipation in smooth regions to first-order dissipation in the neighborhood of shocks. Both dissipation forms impose entropy conditions to ensure nonvanishing convective and acoustic eigenvalues of the inviscid Jacobian matrix.

The third dissipation scheme is the convective upwind and split pressure (CUSP) scheme. Jameson [25] designed this scheme so that it can support single interior point discrete shock waves. For this scheme the dissipation flux can be written as

$$D = -\frac{1}{2}\bar{v}\Delta\mathbf{W} - \frac{1}{2}\beta\Delta\mathbf{F} \tag{3.10}$$

with $\bar{v}$ and β being parameters determined such that single interior point shocks are permitted. Discussion and analysis of the CUSP scheme and the HCUSP version are given in Ref. [26]. We note that all of these schemes are upwind type schemes.

3.3. RK/implicit scheme

Define the update for the qth stage of a RK scheme as

$$\mathbf{W}^{(q)} = \mathbf{W}^{(0)} + \delta\mathbf{W}^{(q)}, \tag{3.11}$$

where

$$\delta\mathbf{W}^{(q)} = \mathbf{W}^{(q)} - \mathbf{W}^{(0)} = -\alpha_q\frac{\Delta t}{\mathscr{V}}\mathscr{L}\mathbf{W}^{(q-1)} \tag{3.12}$$

and $\mathscr{L}$ is the complete difference operator given in Eq. (3.2). To extend the support of the difference scheme we consider implicit residual smoothing. Applying the smoothing technique of Ref. [8] we have the following:

$$\mathscr{L}_i\overline{\delta\mathbf{W}}^{(q)} = \delta\mathbf{W}^{(q)}, \tag{3.13}$$

where $\mathscr{L}_i$ is an implicit operator. By approximately inverting the operator $\mathscr{L}_i$ we obtain

$$\overline{\delta\mathbf{W}}^{(q)} = -\alpha_q\frac{\Delta t}{\mathscr{V}}\mathscr{P}\mathscr{L}\mathbf{W}^{(q-1)} = -\alpha_q\frac{\Delta t}{\mathscr{V}}\mathscr{P}\sum_{\text{all faces}}\mathbf{F}_{\mathrm{n}}^{(q-1)}S, \tag{3.14}$$

where $\mathscr{P}$ is a preconditioner defined by the approximate inverse $\widetilde{\mathscr{L}}_i^{-1}$. The change $\overline{\delta\mathbf{W}}^{(q)}$ replaces the explicit update appearing in Eq. (3.11). Thus, each stage in the RK scheme is preconditioned by an implicit operator.

Unlike the standard scheme, which uses a diffusion operator for the implicit operator $\mathscr{L}_i$, a first-order upwind approximation based on the Roe Scheme is used for the convective derivatives in the implicit operator. To derive this operator one treats the spatial discretization terms in Eq. (3.6) implicitly and applies linearization. For a detailed derivation see Rossow [18]. Substituting for the implicit operator in Eq. (3.13), we obtain for the qth stage of the RK scheme

$$\left[I + \varepsilon\frac{\Delta t}{\mathscr{V}}\sum_{\text{all faces}}\mathbf{A}_{\mathrm{n}}S\right]\overline{\delta\mathbf{W}}^{(q)} = -\alpha_q\frac{\Delta t}{\mathscr{V}}\sum_{\text{all faces}}\mathbf{F}_{\mathrm{n}}^{(q-1)}S = \widehat{\mathbf{R}}^{(q-1)}, \tag{3.15}$$

where the matrix $\mathbf{A}_{\mathrm{n}}$ is the flux Jacobian associated with the normal flux density vector $\mathbf{F}_{\mathrm{n}}$ at a cell face, $\widehat{\mathbf{R}}^{(q-1)}$ represents the residual function for the $(q-1)$th stage, and ε is a parameter to be determined. The matrix $\mathbf{A}_{\mathrm{n}}$ can be decomposed into $\mathbf{A}_{\mathrm{n}}^{+}$ and $\mathbf{A}_{\mathrm{n}}^{-}$, which are defined by

$$\mathbf{A}_{\mathrm{n}}^{+} = \frac{1}{2}(\mathbf{A}_{\mathrm{n}} + |\mathbf{A}_{\mathrm{n}}|), \quad \mathbf{A}_{\mathrm{n}}^{-} = \frac{1}{2}(\mathbf{A}_{\mathrm{n}} - |\mathbf{A}_{\mathrm{n}}|). \tag{3.16}$$

If we substitute for $\mathbf{A}_{\mathrm{n}}$ in Eq. (3.15) using the definitions of Eq. (3.16), then the implicit scheme can be written as

$$\left[I + \varepsilon\frac{\Delta t}{\mathscr{V}}\sum_{\text{all faces}}\mathbf{A}_{\mathrm{n}}^{+}S\right]\overline{\delta\mathbf{W}}_{i,j,k}^{(q)} = \widehat{\mathbf{R}}_{i,j,k}^{(q-1)} - \frac{\Delta t}{\mathscr{V}}\sum_{\text{all faces}}\mathbf{A}_{\mathrm{n}}^{-}\overline{\delta\mathbf{W}}_{\mathrm{NB}}^{(q)}S, \tag{3.17}$$

where the indices (i, j, k) indicate the cell of interest, and NB refers to all the direct neighbors of the cell being considered. As discussed by Rossow [18], the quantity $\mathbf{A}_\mathrm{n}^-\overline{\delta\mathbf{W}}$ represents the flux density change associated with waves having a negative wave speed (i.e. waves that enter the cell (i, j, k) from outside). Only the neighbor cells NB can contribute to these changes in flux density. Similarly, the quantity $\mathbf{A}_\mathrm{n}^+\overline{\delta\mathbf{W}}$ represents flux density changes associated with positive wave speeds (i.e. waves that leave the cell (i, j, k)). These flux density changes are determined only by information from within the cell (i, j, k).

To solve Eq. (3.17) for the changes in conservative variables $\overline{\delta\mathbf{W}}_{i,j,k}^{(q)}$, the 5×5 matrix (a 4×4 matrix in two dimensions) on the left-hand side of Eq. (3.17) must be inverted. It is sufficient to approximate the inverse of the implicit operator. An adequate approximate inverse is obtained with three symmetric Gauss–Seidel sweeps. To initialize the iterative process the unknowns are set to zero. Alternative iterative methods such as red-black Gauss–Seidel could also be used (see Ref. [29]), which would allow the RK/implicit scheme to be applied on unstructured grids.

To efficiently evaluate the Jacobian matrices $\mathbf{A}_\mathrm{n}^+$ and $\mathbf{A}_\mathrm{n}^-$ we rely upon their forms when expressed in terms of the cell interface Mach number M_0. For simplification, we transform Eq. (3.15) to the set of primitive variables $[\rho \quad p \quad u \quad v \quad w]^\mathrm{T}$. Thus,

$$\left[I + \varepsilon\frac{\Delta t}{\mathscr{V}}\sum_{\text{all faces}}\mathbf{P}_\mathrm{n}S\right]\overline{\delta\mathbf{U}}^{(q)} = -\frac{\partial\mathbf{U}}{\partial\mathbf{W}}\alpha_q\frac{\Delta t}{\mathscr{V}}\sum_{\text{all faces}}\mathbf{F}_\mathrm{n}^{(q-1)}S, \tag{3.18}$$

where the matrix $\mathbf{P}_\mathrm{n}$, which is the analog of the normal flux Jacobian expressed in primitive variables, is given by

$$\mathbf{P}_\mathrm{n} = \frac{\partial\mathbf{U}}{\partial\mathbf{W}}\mathbf{A}_\mathrm{n}\frac{\partial\mathbf{W}}{\partial\mathbf{U}} = \frac{\partial\mathbf{U}}{\partial\mathbf{W}}(\mathbf{A}_\mathrm{n}^+ + \mathbf{A}_\mathrm{n}^-)\frac{\partial\mathbf{W}}{\partial\mathbf{U}} = \mathbf{P}_\mathrm{n}^+ + \mathbf{P}_\mathrm{n}^-. \tag{3.19}$$

The Jacobian $\partial\mathbf{U}/\partial\mathbf{W}$ on the right-hand side of Eq. (3.18) must multiply the conservative flux balance in order to ensure conservation. Using the definitions of Eq. (3.16) and the dissipation matrix, which is defined in Appendix A, one can determine the matrices $\mathbf{P}_\mathrm{n}^+$ and $\mathbf{P}_\mathrm{n}^-$, which are also given in Appendix A. The resulting matrices can easily be recomputed, only requiring storage for the normal velocity magnitude and the Mach number M_0. The contributions of the viscous flux Jacobians can be included in a straightforward manner using primitive variables (see Ref. [20]). We present the viscous Jacobian for the thin-layer form of the Navier–Stokes equations in Appendix A.

Due to the upwind approximation used for the implicit operator, the coefficients for the RK scheme are also based on an upwind scheme. Now the numerical dissipation is evaluated at every stage. For three-stage and five-stage schemes, we use respectively the RK coefficients from Ref. [30] of

$$[\alpha_1, \ldots, \alpha_3] = [0.15, 0.4, 1.0],$$
$$[\alpha_1, \ldots, \alpha_5] = [0.0695, 0.1602, 0.2898, 0.5060, 1.0].$$

We summarize the implementation of the RK/implicit scheme as follows. In the first step, the explicitly evaluated residuals of a RK stage are transformed to residuals in primitive variables to form the right-hand side of Eq. (3.18). Next, we approximately invert the implicit operator with symmetric Gauss–Seidel. This yields new residuals in primitive variables, which are transformed to conservative variables. As the final step, the new residuals (i.e. new changes) are used in the RK stage to update the conservative variables.

4. Fourier analysis

In designing a rapidly converging scheme for solving the Euler and Navier–Stokes equations there are several factors one must consider. First, if the scheme is to be used as a smoother for multigrid, then it must have good damping of high-frequency error components. In addition, one should design the scheme to cluster the residual eigenvalues corresponding to the high-frequency modes away from the origin in the complex plane. Another important factor is the magnitude of the CFL number. The scheme should be constructed so that the CFL number is sufficiently large to produce significant reduction (if not elimination) of the convergence slowdown effects that are associated with high-aspect ratio mesh cells. A large CFL number also facilitates the expulsion of error components. At the same time the capability for large CFL numbers must not compromise the high-frequency damping property of the scheme. The RK/implicit scheme can satisfy both of these criteria. In constructing these types of schemes there are several factors to consider.

An important consideration in designing RK/implicit schemes is the selection of the RK coefficients. We have selected coefficients that have been determined so as to give optimal damping of high frequencies for a given spatial differencing operator. However, the damping behavior of the RK scheme is changed due to the introduction of the implicit preconditioner. In order to ensure good h-ellipticity (high-frequency damping) of the scheme, a parameter is introduced into the implicit operator. This parameter, which we designate by ε, multiplies the implicit spatial operator (see Eq. (4.15)) and takes on a role similar to that of β in the scalar preconditioner of the RK/standard scheme. The magnitude of ε for best damping depends on the number of RK stages. When reducing the number of RK stages, the lower limit is a RK$(1,1)$ implicit scheme, which essentially represents the classical backward Euler implicit scheme. Fourier analysis shows that such a scheme is unconditionally stable and exhibits good damping properties if the implicit (LHS) and explicit (RHS) discretizations match. However, for practical applications usually only a first-order discretization is employed for the LHS operator, whereas second-order accuracy is required for the RHS operator. This still provides unconditional stability, but the damping properties of the scheme are impaired. Therefore, we consider only schemes with at least two stages. As we will demonstrate shortly one can determine a sufficiently small ε so as to have good damping over a broad range of frequencies and maintain stability.

In order to use the preconditioner one must compute the inverse of the implicit operator at each RK stage. How well this inverse approximates the implicit operator can have a significant impact on the performance of the RK/implicit scheme. For example, if lexicographic pointwise symmetric Gauss–Seidel (SGS) is used to approximately invert the implicit operator, then one must determine the number of symmetric sweeps that is appropriate, keeping in mind computational effort as well as convergence rate. Another important factor for the RK/implicit scheme is the number of stages. Choosing a small number of RK stages is beneficial for a low computational effort. In reducing the number of stages one must make sure that the eigenvalue clustering property of the scheme is not seriously compromised. We apply Fourier analysis to evaluate the properties of different RK/implicit schemes.

For the Fourier analysis we consider a finite domain with periodic boundary conditions. We take a Fourier transform of the discretized form of the linearized (constant coefficient), time-dependent Euler equations when solved with a RK scheme combined with an implicit preconditioner. Initially, as a reference, we consider a standard RK$(5,3)$ scheme preconditioned with a scalar implicit residual smoothing. Then we compare the damping properties of this scheme with those of several RK schemes with a fully implicit preconditioner. A principal objective is to evaluate these schemes, which we call RK/implicit schemes, as smoothers for a full-coarsened multigrid method. An additional objective is to provide guidance in determining the implicit parameter ε.

The Fourier analysis is applied to the 2-D, nonconservative Euler equations with the solution vector $[s \quad u \quad v \quad p]^{\mathrm{T}}$. Consider the domain $\Omega = \{(x,y) : 0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1\}$. Define a Cartesian grid with $m \times n$ cells and spacings h_x and h_y. Let $\mathbf{W}_{j_1,j_2}$ denote the discrete solution vector that resides at the mesh point $(j_1 h_x, j_2 h_y)$. Now consider the semi-discrete form of the flow equations, which can be written as

$$\Delta t \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{W}_{j_1,j_2} = -\frac{\Delta t}{\mathscr{V}}\mathscr{L}^h \mathbf{W}_{j_1,j_2}, \tag{4.1}$$

where $\mathscr{L}^h$ is the linearized discrete residual operator defined by

$$\mathscr{L}^h \equiv \mathbf{A}\delta_x + \mathbf{B}\delta_y \tag{4.2}$$

with $\mathbf{A}$ and $\mathbf{B}$ being flux Jacobian matrices and δ representing a difference operator. The spatial discretization is carried out by upwind biased differencing. Using the left and right eigenvectors of the Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$ to generate similarity transformations we obtain

$$|\mathbf{A}| = S|\Lambda_A|S^{-1}, \quad |\mathbf{B}| = T|\Lambda_B|T^{-1}. \tag{4.3}$$

Expressing the second-order upwind difference approximation as a sum of a central difference and numerical dissipation, the discrete linear residual operator is written as

$$\mathscr{L}^h = \mathscr{L}^h_{\mathrm{c}} + \mathscr{L}^h_{\mathrm{d}}, \tag{4.4}$$

where

$$\mathscr{L}^h_{\mathrm{c}} = \frac{h_y}{2}[\mathbf{A}(E_x^{+1} - E_x^{-1})] + \frac{h_x}{2}[\mathbf{B}(E_y^{+1} - E_y^{-1})], \tag{4.5}$$

$$\mathscr{L}^h_{\mathrm{d}} = \frac{h_y}{4}[|\mathbf{A}|(E_x^{-2} - 4E_x^{-1} + 6 - 4E_x^{+1} + E_x^{+2})] + \frac{h_x}{4}[|\mathbf{B}|(E_y^{-2} - 4E_y^{-1} + 6 - 4E_y^{+1} + E_y^{+2})]. \tag{4.6}$$

The shift operators $E_x^{\pm l}$ and $E_y^{\pm l}$ are defined by

$$E_x^{\pm l}\mathbf{W}_{j_1,j_2} = \mathbf{W}_{j_1\pm l,j_2}, \quad E_y^{\pm l}\mathbf{W}_{j_1,j_2} = \mathbf{W}_{j_1,j_2\pm l}. \tag{4.7}$$

Taking the discrete Fourier transform of Eq. (4.1) we obtain

$$\Delta t\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{W}}_{k_1,k_2} = \Delta\widehat{\mathbf{W}}_{k_1,k_2} = \widehat{\mathbf{W}}^{n+1}_{k_1,k_2} - \widehat{\mathbf{W}}^{n}_{k_1,k_2} = -\frac{\Delta t}{\mathscr{V}}\widehat{\mathscr{L}}^h\widehat{\mathbf{W}}_{k_1,k_2}, \tag{4.8}$$

where the superscript n refers to time step. The transformed discrete vector function is given by

$$\widehat{\mathbf{W}}_{k_1,k_2} = \frac{1}{m_f n_f}\sum_{j_1=0}^{m_f-1}\sum_{j_2=0}^{n_f-1}\mathbf{W}_{j_1,j_2}\exp[-\mathrm{i}(j_1\theta_x + j_2\theta_y)] \tag{4.9}$$

and the phase angles θ_x and θ_y are defined by

$$\theta_x = 2\pi\frac{k_1}{m_f}, \quad \theta_y = 2\pi\frac{k_2}{n_f}. \tag{4.10}$$

The wave numbers are given by

$$k_1 = -\left(\frac{1}{2}m_f - 1\right),\ldots,\frac{1}{2}m_f, \quad k_2 = -\left(\frac{1}{2}n_f - 1\right),\ldots,\frac{1}{2}n_f. \tag{4.11}$$

The transformed residual operator $\widehat{\mathscr{L}}^h$ is a function of the transformed shift operators, which are defined by

$$\widehat{E}_x \equiv \exp(\mathrm{i}\theta_x), \quad \widehat{E}_y \equiv \exp(\mathrm{i}\theta_y), \quad -\pi < \theta_x \leqslant \pi, \ -\pi < \theta_y \leqslant \pi. \tag{4.12}$$

If we apply a q-stage RK scheme (with the numerical dissipation computed on each stage) to integrate in time Eq. (4.8), then

$$\widehat{\mathbf{W}}^{n+1}_{k_1,k_2} = \mathbf{G}_{rk}\widehat{\mathbf{W}}^{n}_{k_1,k_2}, \tag{4.13}$$

where the amplification matrix $\boldsymbol{G}_{rk}$ is given by

$$\mathbf{G}_{rk} = I - \bar{\alpha}_q\widehat{\mathscr{L}}^h + \bar{\alpha}_q\bar{\alpha}_{q-1}(\widehat{\mathscr{L}}^h)^2 - \bar{\alpha}_q\bar{\alpha}_{q-1}\bar{\alpha}_{q-2}(\widehat{\mathscr{L}}^h)^3 + \cdots - (\bar{\alpha}_q\bar{\alpha}_{q-1}\cdots\bar{\alpha}_1)(\widehat{\mathscr{L}}^h)^q \tag{4.14}$$

with I denoting the identity matrix and the $\bar{\alpha}_q$ representing the product of the RK coefficient and the time step divided by the volume of the mesh cell ($\Delta t/\mathscr{V}$). For the RK/implicit scheme we introduce an implicit preconditioner

$$\mathscr{P}^{-1} = L_i = I + \varepsilon\frac{\Delta t}{\mathscr{V}}[h_y(\mathbf{A}^+(I - E_x^{-1}) - \mathbf{A}^-(I - E_x^{+1}))] + \varepsilon\frac{\Delta t}{\mathscr{V}}[h_x(\mathbf{B}^+(I - E_y^{-1}) - \mathbf{B}^-(I - E_y^{+1}))], \tag{4.15}$$

where the matrices $\mathbf{A}^+$, $\mathbf{B}^+$ and $\mathbf{A}^-$, $\mathbf{B}^-$ associated with the positive and negative eigenvalues, respectively, of the matrices $\mathbf{A}$ and $\mathbf{B}$ are defined according to Eq. (3.16). The implicit parameter ε allows the RK/implicit scheme to be designed to effectively damp high-frequency error modes. Then, for the qth stage of the RK scheme we have

$$\widehat{\mathbf{W}}^{(q)}_{k_1,k_2} = \widehat{\mathbf{W}}^{(0)}_{k_1,k_2} - \alpha_q\frac{\Delta t}{\mathscr{V}}\widehat{\mathscr{P}}\widehat{\mathscr{L}}^h\widehat{\mathbf{W}}^{(q-1)}_{k_1,k_2}. \tag{4.16}$$

The amplification matrix of the RK/implicit scheme is obtained by replacing $\widehat{\mathscr{L}}$ with $\widehat{\mathscr{P}}\widehat{\mathscr{L}}$ in Eq. (4.14), giving

$$\mathbf{G}_{rki} = I - \bar{\alpha}_q\widehat{\mathscr{P}}\widehat{\mathscr{L}}^h + \bar{\alpha}_q\bar{\alpha}_{q-1}(\widehat{\mathscr{P}}\widehat{\mathscr{L}}^h)^2 - \bar{\alpha}_q\bar{\alpha}_{q-1}\bar{\alpha}_{q-2}(\widehat{\mathscr{P}}\widehat{\mathscr{L}}^h)^3 + \cdots - (\bar{\alpha}_q\bar{\alpha}_{q-1}\cdots\bar{\alpha}_1)(\widehat{\mathscr{P}}\widehat{\mathscr{L}}^h)^q. \tag{4.17}$$

We approximate the inverse of the implicit operator with pointwise symmetric Gauss–Seidel iterations. For one symmetric sweep the approximate inverse is given by

$$\mathscr{P} = (L_i^-)^{-1} + [I - (L_i^-)^{-1} L_i](L_i^+)^{-1}, \tag{4.18}$$

where L_i is the implicit operator defined in Eq. (4.15), and L_i^+ and L_i^- are the implicit operators defined by

$$L_i^+ = I + \varepsilon \frac{\Delta t}{\mathscr{V}} \{ h_y [\mathbf{A}^+ (I - E_x^{-1}) - \mathbf{A}^-] + h_x [\mathbf{B}^+ (I - E_y^{-1}) - \mathbf{B}^-] \}, \tag{4.19}$$

$$L_i^- = I - \varepsilon \frac{\Delta t}{\mathscr{V}} \{ h_y [\mathbf{A}^- (I - E_x^{+1}) - \mathbf{A}^+] + h_x [\mathbf{B}^- (I - E_y^{+1}) - \mathbf{B}^+] \}. \tag{4.20}$$

For additional iterations to approximate the inverse [29], we have

```
$P_1 = 0$
for iter = 1 : niter
$P_1 = \mathscr{P} + (I - \mathscr{P} * L_i) * P_1$
end
$\mathscr{P} = P_1$
```

where niter is the number of symmetric sweeps in the iterative process.

The standard scheme, which is the RK(5,3) scheme described in Section 3.1, has five stages with three weighted evaluations of the numerical dissipation. Thus, the amplification matrix is no longer a simple polynomial in terms of the transformed operator. It is given by

$$\mathbf{G}_{rk} = I - \bar{\alpha}_5 \{ I - \bar{\alpha}_4 \varGamma_5 [I - \bar{\alpha}_3 \widehat{\mathscr{Q}}_\mathrm{c} + (\bar{\alpha}_3 \bar{\alpha}_2 \widehat{\mathscr{Q}}_\mathrm{c} \varGamma_3 - \bar{\alpha}_2 \gamma_3 \widehat{\mathscr{Q}}_\mathrm{d}) \varGamma_1] \} \widehat{\mathscr{Q}}^h + \bar{\alpha}_5 \bar{\alpha}_2 \gamma_3 (1 - \gamma_5) \widehat{\mathscr{Q}}_\mathrm{d} \varGamma_1 \widehat{\mathscr{Q}}^h \tag{4.21}$$

with

$$\varGamma_1 = I - \bar{\alpha}_1 \widehat{\mathscr{Q}}_\mathrm{c}, \quad \varGamma_3 = \widehat{\mathscr{Q}}_\mathrm{c} + \gamma_3 \widehat{\mathscr{Q}}_\mathrm{d}, \quad \varGamma_5 = \widehat{\mathscr{Q}}_\mathrm{c} + \gamma_5 \widehat{\mathscr{Q}}_\mathrm{d},$$
$$\mathscr{Q}^h = \mathscr{P}_s \mathscr{L}^h, \quad \mathscr{Q}_\mathrm{c} = \mathscr{P}_s \mathscr{L}_\mathrm{c}, \quad \mathscr{Q}_\mathrm{d} = \mathscr{P}_s \mathscr{L}_\mathrm{d},$$

$\mathscr{P}_s$ being the two-dimensional form of the scalar implicit preconditioner given in Eq. (3.3), and γ_3 and γ_5 denoting the weights of the numerical dissipation (i.e. $\gamma_3 = 0.56$, $\gamma_5 = 0.44$), on the third and fifth stages, respectively. The variable coefficients of the scalar preconditioner [20] are as follows:

$$\beta_\xi = \max \left\{ \frac{1}{4} \left[\left(\frac{N}{N^*(1 + \psi \mathrm{AR})} \right)^2 - 1 \right], 0 \right\}, \tag{4.22}$$

$$\beta_\eta = \max \left\{ \frac{1}{4} \left[\left(\frac{N}{N^*(1 + \psi \mathrm{AR}^{-1})} \right)^2 - 1 \right], 0 \right\}, \tag{4.23}$$

where N is the CFL number for the preconditioned scheme, N^* is the CFL number for the basic RK scheme, AR denotes aspect ratio (h_x/h_y), and ψ is a parameter (taken to be 0.11). Usually, the CFL number ratio N/N^* is 2 and $N = 7.5$.

As a reference, we examine the damping characteristics of the standard scheme. Let λ be an eigenvalue of the amplification matrix $\boldsymbol{G}_{rk}$, which is defined by Eq. (4.21) and is a function of the phase angles (θ_x, θ_y) associated with the Cartesian coordinates (x, y). Also, let $g(\theta_x, \theta_y) = \max[\lambda(\mathbf{G}_{rk})]$ for a given (θ_x, θ_y). For the analysis we consider a flow Mach number of 0.5 and a flow angle of 45°. Fig. 1(a) shows the contours of g when $\mathrm{AR} = 1$. The damping of the high-high, high-low, and low-high frequency modes is similar. The smoothing factor, which is the maximum g outside the dashed line square (i.e. $\frac{\pi}{2} \leqslant |\theta_x|, |\theta_y| \leqslant \pi$) of Fig. 1(a), is approximately 0.75. To have a good smoother for full-coarsened multigrid, the difference operator must have good h-ellipticity. That is, all of the modes with high-frequency components must be well damped. A good smoother is also one that clusters the eigenvalues away from the unit circle (stability boundary), with most, if not all, of them lying within the circle with radius of 1/2 (i.e. a smoothing factor of 1/2). In Fig. 1(b), which shows the

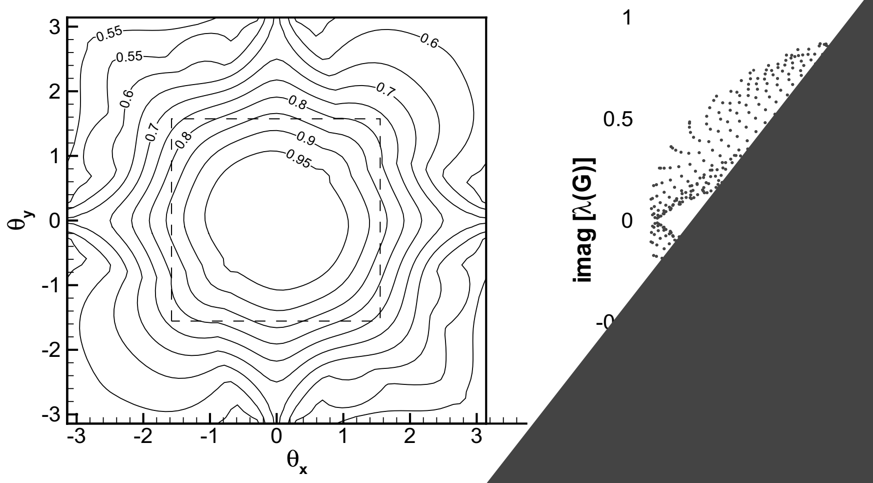
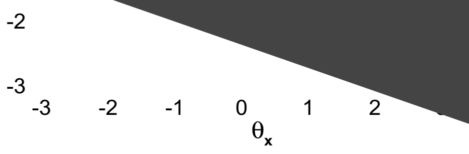


Fig. 1. RK/standard scheme applied to 2-D Euler equatio
and (b) eigenvalue spectrum.

eigenvalue spectrum $\lambda(\mathbf{G})$ when $\mathbf{G} =$
a large number of eigenvalues lie o
AR = 100. For this case the high
damping behavior supports th
smoother and the mesh AR i
once again poor clustering

For the RK/implicit s
ior. In Fig. 3 the varia
one-dimensional (1-
importance of ch
the desired valu
damping is at
dissipation



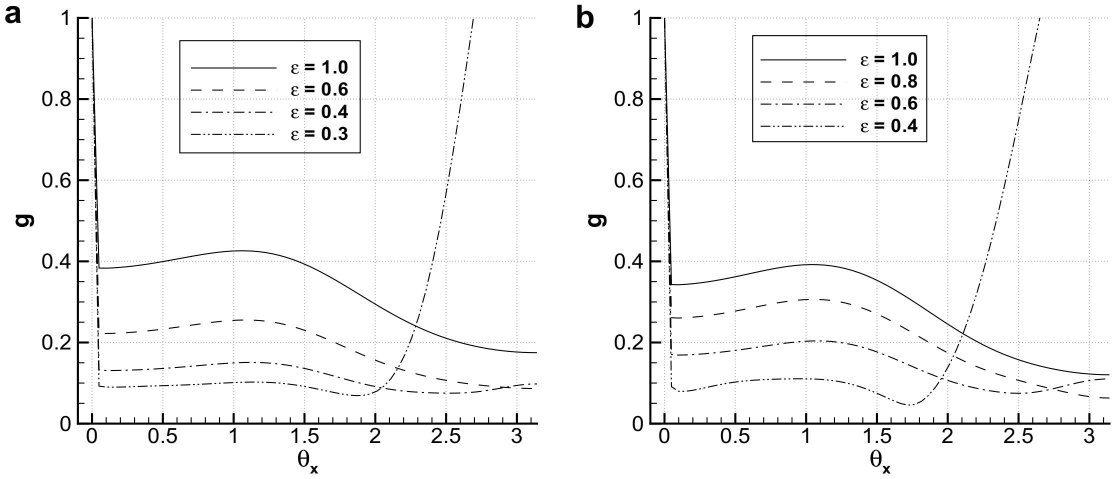
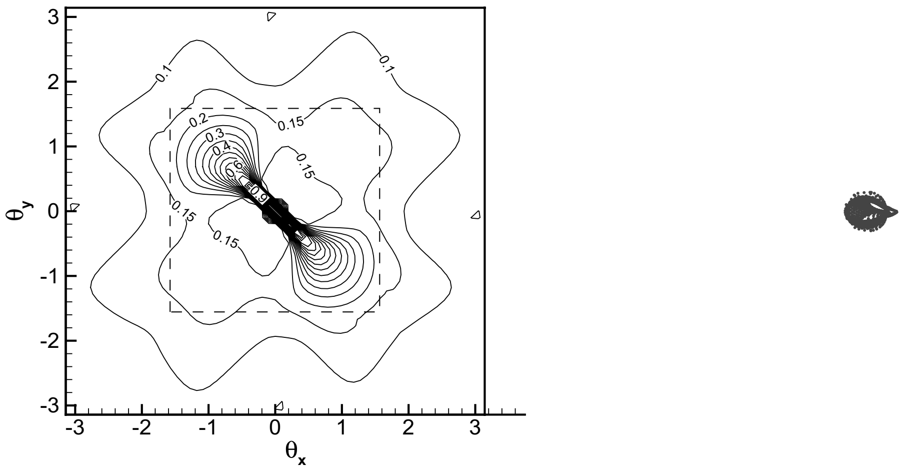


Fig. 3. Effect on amplification factor of RK/implicit scheme (applied to 1-D Euler equations) due to variation of implicit parameter (ε): (a) 5-stage scheme; (b) 3-stage scheme.

several 2-D flows have demonstrated that $\varepsilon = 0.5$ works well with matrix dissipation for both the 5-stage and 3-stage schemes. With Roe dissipation $\varepsilon = 0.6$ has proven to be a good choice for both schemes.

The contours of g and eigenvalue distribution for the 5-stage RK/implicit scheme are displayed in Fig. 4. In Fig. 4(b) and in subsequent figures the amplification matrix $\mathbf{G} = \mathbf{G}_{rki}$. One can clearly see the improved damping and eigenvalue clustering of the 5-stage RK/implicit scheme relative to the RK/standard scheme. Although not shown, the 3-stage scheme and a 2-stage scheme exhibit similar damping. For these results three symmetric Gauss–Seidel (SGS) sweeps were performed to approximately invert the implicit operator. The effect on the damping of the scheme due to a reduction in the number of SGS sweeps is seen in Fig. 5. While two SGS sweeps produce similar damping behavior, there is a deterioration in the damping of some high-frequency modes; in particular, those modes that have a high-frequency component in one mesh direction and a low-frequency component in the other. When considering a less accurate approximation for the inverse of the implicit operator, one must not only consider the possible adverse effect on the damping properties but also the reduced computational effort. The reduction in the operation count may sufficiently reduce the overall computational time to compensate for the slower convergence rate. In Fig. 6 the damping and eigenvalue dis-



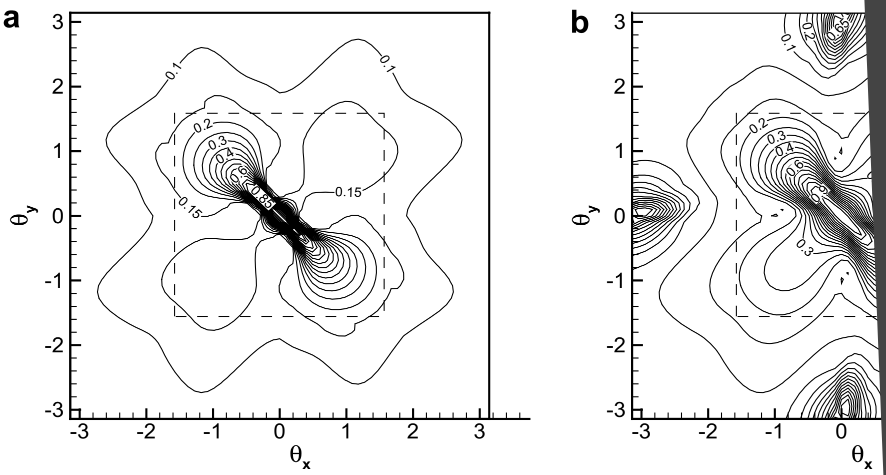


Fig. 5. Effect of SGS sweeps on the amplification factor of RK5/implicit scheme applied to 2-D Euler eq cells, CFL = 1000, ε = 0.4: (a) 2 SGS sweeps; (b) 1 SGS sweep.

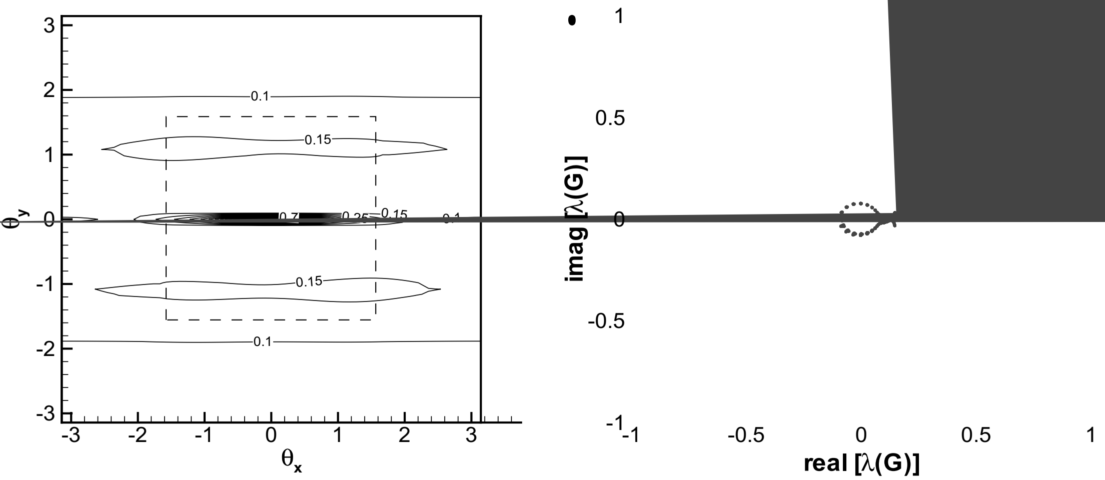


Fig. 6. RK5/implicit scheme applied to 2-D Euler equations: AR = 100, grid: 64 × 64 cells, CFL = 1000, ε = 0.4. (a) Contours of amplification factor and (b) eigenvalue spectrum.

tribution of the 5-stage scheme when AR = 100 are shown. Here again there is a dramatic improvement in damping behavior relative to the standard scheme.

Based on the analysis of the RK/implicit scheme for the linear system there is no limitation on the CFL number provided ε is sufficiently large. This strong stability property provides flexibility in the scheme which can be important as the mesh cell AR increases with increasing Reynolds number. For example, one may consider the possibility of increasing the CFL number in a strip region near a solid boundary, since an unlimited CFL number cannot be used throughout the domain when solving the nonlinear flow equations. Thus, one could compensate for the decrease in time step as the AR increases, retaining good damping of the high-frequency modes.

5. Numerical results

The RK/implicit scheme was used to compute turbulent, viscous flow over the RAE 2822 airfoil and the ONERA M6 wing. The effects of turbulence were included by applying the Baldwin-Lomax model [31].

Table 1
Flow conditions for RAE 2822 airfoil

Cases	M_∞	α (in degrees)	Re_c	x_{tr}/c
Case 1	0.676	1.93	5.7×10^6	0.11
Case 9	0.730	2.79	6.5×10^6	0.03
Case 10	0.750	2.81	6.2×10^6	0.03

The airfoil solutions were calculated with the Case 1, Case 9 and Case 10 flow conditions (see Table 1) from the experimental investigation of Cook et al. [32]. For Case 1 the flow is primarily subsonic with a relatively small region of supersonic flow. In Case 9, one of the most frequently used cases in evaluating computational methods, there is a fairly strong shock occurring on the upper surface of the airfoil. For Case 10 a stronger shock occurs on the upper surface, resulting in substantial separation behind the shock that nearly merges with the trailing edge separation. This case often causes numerical oscillations that result in a significant deterioration in the convergence rate. For the wing flow, solutions were computed with flow conditions from the experiment of Schmitt and Charpin [33]. This is a supercritical flow with free-stream Mach number $M_\infty = 0.84$, angle of attack $\alpha = 3.06^\circ$ and Reynolds number $Re = 11.7 \times 10^6$. In this case a λ shock occurs on the upper surface of the wing, due to the double shock at the inboard stations merging with a much stronger single shock at the outboard stations.

For computing solutions of the three RAE 2822 cases a structured C-type mesh with 64 cells in the normal direction and 320 cells around the airfoil (256 cells on the airfoil) was used. The normal spacing of this mesh at the airfoil surface is 1.0×10^{-5}, and the maximum cell aspect ratio occurring at the surface is 2413. In order to investigate the performance of the RK/implicit scheme for a range of Reynolds numbers (between 5.7×10^6 and 100×10^6) a family of C-type meshes was generated [34]. These meshes are adapted to the Reynolds number of the flow, and the resulting cell aspect ratios vary from about 3000 to over 50,000. Each mesh has 368 cells around the airfoil (312 cells on the airfoil) and 88 cells normal to the airfoil. For these meshes the normal spacing varies from 3.7×10^{-6} to 2.3×10^{-7}. For the multigrid algorithm coarse meshes were created by eliminating every other mesh line in each coordinate direction (i.e. full coarsening). A four-level W-cycle (2-D) and a three-level V-cycle (3-D) were employed to execute the multigrid. All 2-D calculations were performed on a Linux workstation with a Pentium 4 and a dual 3.0 GHz processor.

In all the 2-D applications the same boundary conditions were imposed. On the surface the no-slip condition was applied. At the outer boundary Riemann invariants were used. A far-field vortex effect was included to specify the velocity for an inflow condition at the outer boundary. A detailed discussion of the boundary conditions is given in Ref. [20]. For the 3-D cases the far-field vortex effect was not included; but otherwise, the boundary conditions were the same. The calculations were started on the solution grid with the initial solution given as the free-stream conditions. When a full multigrid process was applied, the initial solution on a given level of refinement was obtained from a coarser level.

5.1. Two-dimensional airfoil flows

In Fig. 7, the convergence histories for Case 9 of the RK/standard and the 5-stage RK/implicit schemes are compared. For all results the residuals are normalized by the corresponding residual of the first iteration. The number after RK indicates the number of stages. These histories show the behavior of the L_2 norm of the residual of the continuity equation with multigrid cycles. Unless indicated otherwise, the calculations were started on the finest grid for the 2-D results. For the RK/standard scheme the numerical dissipation operator is given by matrix dissipation [27], and for the RK5/implicit scheme the dissipation operator is based on the Roe scheme [22]. In the calculations with the RK5/implicit scheme the CFL number was 16 during the first 8 multigrid cycles; and then, it was increased to 1000. From the figures, one can see that the RK5/implicit scheme requires a factor of about 17 fewer multigrid cycles than the standard scheme to reduce the residual 13 orders of magnitude. The average rate of reduction of the residual (i.e. convergence rate) for the standard scheme exceeds 0.98, while for the RK5/implicit scheme the rate is 0.76. With respect to computational effort, the RK5/implicit scheme is about 2.5 times faster than the standard scheme. As shown in Ref. [36] the

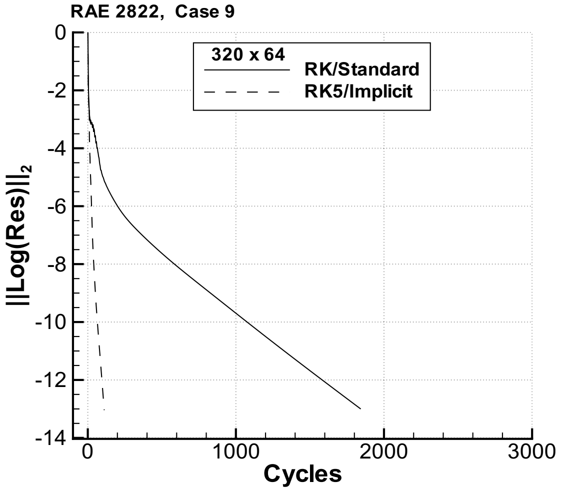


Fig. 7. Convergence histories of RK/standard scheme and RK5/implicit scheme with Roe dissipation: Case 9, 320 × 64 grid.

computed pressure distribution agrees fairly well with the experimental data. The computed and experimental lift and drag coefficients are given in Table 2.

Similar convergence histories were obtained for Cases 1 and 10. In Tables 3–5 the convergence rates, number of multigrid cycles (to reduce residual thirteen orders of magnitude) and computational times are given for both the RK/standard and RK5/implicit schemes applied to all cases. The effect on convergence of the RK5/implicit due to alternative dissipation forms is also included in the tables. With matrix and HCUSP dissipation the RK/implicit scheme is roughly between 2 and 2.7 times faster, in computer time, than the standard scheme for all cases. The computational savings with the matrix dissipation is about the same as it is with the Roe scheme dissipation even though the convergence rates are faster with the Roe scheme. This is because matrix dissipation does not have the additional expense of evaluating a limiter for each characteristic field.

5.1.1. Implicit parameter

In Section 4, we examined in one dimension the damping behavior of the RK/implicit scheme with variation in the implicit parameter ε. Using this analysis as guidance, we performed numerical experiments for a

Table 2
Computed lift and drag coefficients for RAE 2822 airfoil

Cases	C_L	C_D	$(C_D)_p$	$(C_D)_v$
Case 1	0.6101	0.008315	0.002528	0.005787
Case 9	0.8530	0.01783	0.01232	0.005506
Case 10	0.8480	0.02885	0.02342	0.005409

Numerical dissipation from Roe scheme. Weak grid clustering in neighborhood of shock wave.

Table 3
Computational effort required for Case 1, 320 × 64 grid

Scheme	Dissipation	CPU time (s)	MG cycles	Convergence rate
RK/standard	Matrix	481	1792	0.983
RK5/implicit	Roe	180	98	0.736
RK3/implicit	Roe	111	97	0.733
RK5/implicit	Matrix	202	128	0.791
RK3/implicit	Matrix	126	127	0.789
RK5/implicit	HCUSP	243	146	0.815
RK3/implicit	HCUSP	152	145	0.813

Table 4
Computational effort required for Case 9, 320 × 64 grid

Scheme	Dissipation	CPU time (s)	MG cycles	Convergence rate
RK/standard	Matrix	509	1891	0.984
RK5/implicit	Roe	201	110	0.761
RK3/implicit	Roe	126	110	0.761
RK5/implicit	Matrix	203	128	0.791
RK3/implicit	Matrix	124	125	0.787
RK5/implicit	HCUSP	242	145	0.813
RK3/implicit	HCUSP	150	144	0.812

Table 5
Computational effort required for Case 10, 320 × 64 grid

Scheme	Dissipation	CPU time (s)	MG cycles	Convergence rate
RK/standard	Matrix	680	2519	0.988
RK5/implicit	Roe	260	143	0.811
RK3/implicit	Roe	161	141	0.808
RK5/implicit	Matrix	252	159	0.828
RK3/implicit	Matrix	152	153	0.822
RK5/implicit	HCUSP	261	157	0.826
RK3/implicit	HCUSP	163	155	0.824

range of flow conditions to determine an appropriate ε for a given type of dissipation. The values of ε selected for the schemes with Roe and matrix dissipation are 0.6 and 0.5, respectively. On the coarse meshes of the multigrid method ε is 0.4. Fig. 8 shows the effect of varying ε on the convergence of two RK5/implicit schemes for Case 9. With Roe dissipation there is a relatively small variation in convergence (i.e. convergence rate between 0.75 and 0.77) when ε is changed by ±0.1. The variation with matrix dissipation is somewhat larger, as the convergence rate is between 0.76 and 0.81. This larger variation in convergence rate may be due to differences between the explicit operator (matrix dissipation) and the implicit operator (Roe dissipation).

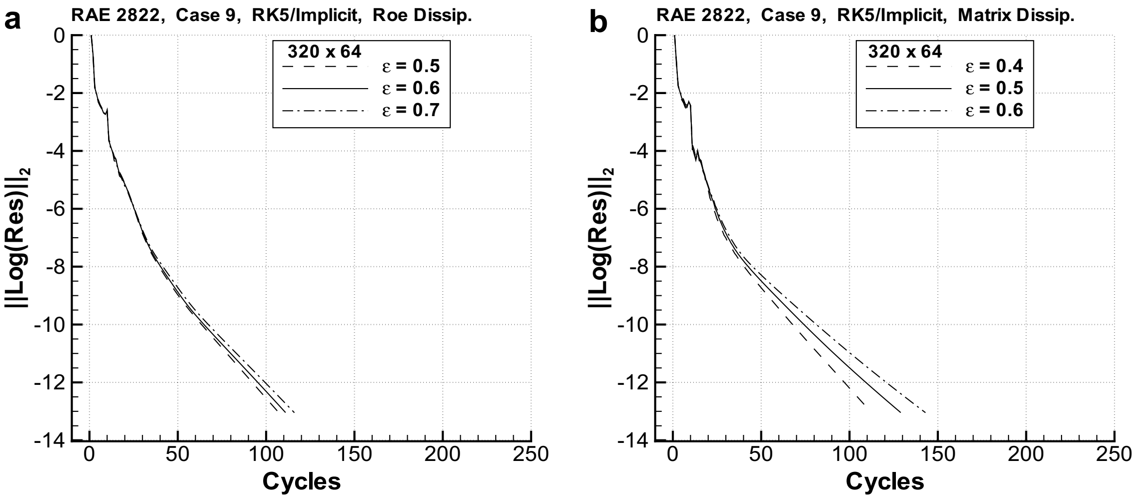


Fig. 8. Effect of implicit parameter ε on convergence of the RK5/implicit scheme with two types of dissipation: Case 9, 320 × 64 grid. (a) Roe dissipation, (b) matrix dissipation.

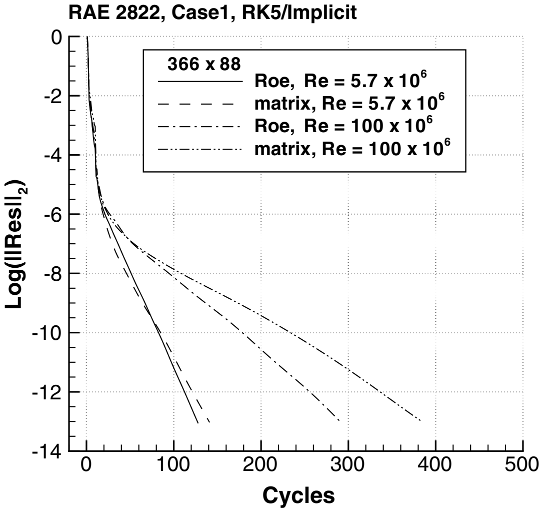


Fig. 9. Convergence histories of RK5/implicit schemes with Roe and matrix dissipation showing effect of Reynolds number variation: Case 1, 366 × 88 grid.

5.1.2. Scheme behavior for high AR grids

The results presented so far are for a grid with moderately high aspect ratio cells. To investigate how the RK/implicit scheme alleviates stiffness associated with high aspect ratio cells, calculations were performed with the Reynolds number varying by more than an order of magnitude. The computational meshes are the same as the ones used by Faßbender [34] to examine the effects of Reynolds number variation on turbulence modeling. To avoid difficulties, such as convergence stall, that can occur due to limiter functions, the flow conditions (M_∞, α) for an essentially subsonic case (Case 1) were used for the calculations. In Fig. 9, convergence histories are presented for the RK5/implicit scheme with Roe and matrix dissipation forms when $Re = 5.7 \times 10^6$ and $Re = 100 \times 10^6$. The maximum surface grid cell aspect ratios for the two Re values are 3949 and 50,260. From the two sets of curves in the figure we see that over the Re range the number of multigrid cycles only increases by a factors of 2.3 and 2.7 with Roe and matrix dissipation, respectively. In Tables 6 and 7 the convergence rates and computing times for $Re = 5.7 \times 10^6$ and $Re = 100 \times 10^6$ are displayed. Convergence quantities for other Reynolds numbers are given in Ref. [36]. For higher Re values the convergence

Table 6
Computational effort required for Case 1, 368 × 88 grid, $Re = 5.7 \times 10^6$, AR = 3949

Scheme	Dissipation	CPU time (s)	MG cycles	Convergence rate
RK/standard	Matrix	1105	2516	0.988
RK5/implicit	Roe	371	128	0.791
RK3/implicit	Roe	230	126	0.788
RK5/implicit	Matrix	368	140	0.807
RK3/implicit	Matrix	217	136	0.802

Table 7
Computational effort required for Case 1, 368 × 88 grid, $Re = 100 \times 10^6$, AR = 50, 260

Scheme	Dissipation	CPU time (s)	MG cycles	Convergence rate
RK/standard	Matrix	3458	7865	0.996
RK5/implicit	Roe	841	291	0.902
RK3/implicit	Roe	521	286	0.901
RK5/implicit	Matrix	980	384	0.925
RK3/implicit	Matrix	600	378	0.924

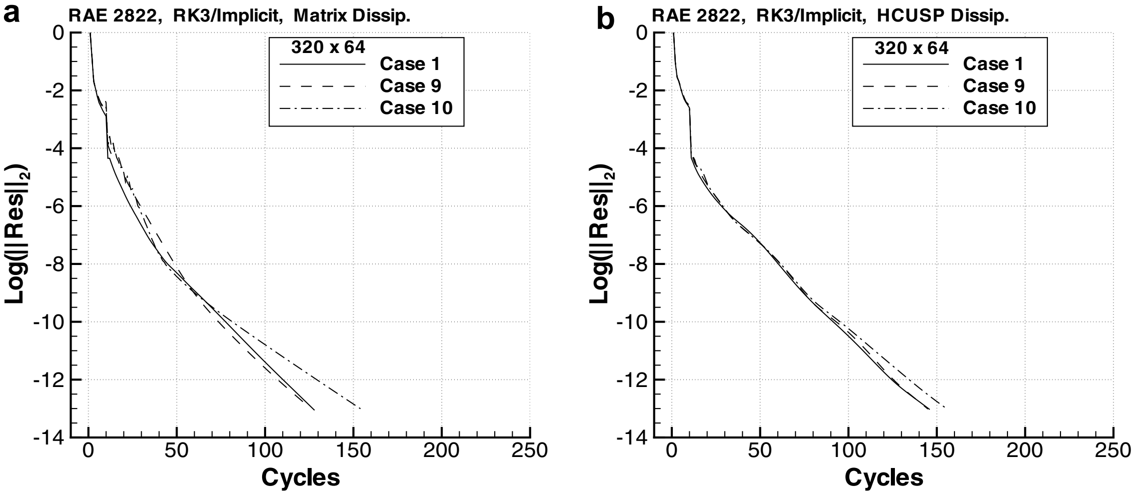


Fig. 10. Convergence histories of RK3/implicit scheme for Cases 1, 9 and 10: 320 × 64 grid. (a) Matrix dissipation, (b) HCUSP dissipation.

rate with the standard scheme exceeds 0.995. Using the Roe scheme dissipation, the RK5/implicit scheme converges at rates between 0.88 and 0.90 for the higher *Re* values (i.e. at higher cell aspect ratios). The corresponding reduction in computational time is about a factor of 4 relative to the standard scheme.

5.1.3. Reducing number of RK stages

There are several alternative ways to increase the computational efficiency. One alternative is to reduce the number of stages in the RK scheme. With such an approach one would expect almost no loss in numerical efficiency, since we use a CFL number of 1000. Calculations were performed for Cases 1, 9, and 10 with the RK/implicit scheme using 3 stages and CFL = 1000. The solutions were obtained with the same (moderately high aspect ratio) 320 × 64 grid used for the results of the RK/implicit scheme with 5 stages. Fig. 10 shows the convergence histories with the matrix and HCUSP dissipation forms, respectively. As indicated in Tables 3–7 the convergence rates with the 5-stage and 3-stage schemes for each dissipation form are nearly the same. The RK3/implicit scheme with Roe and matrix dissipation is about 4–6.5 and 3.8–5.5 times faster, respectively, (depending on the Reynolds number) than the RK/standard scheme. Another approach for reducing the computing time is to lower the number of RK stage evaluations of the numerical dissipation and/or to decrease the number of SGS sweeps for approximately inverting the implicit operator. For example, Rossow [35] demonstrates that even with the convergence rate penalty produced by performing only one SGS sweep rather than three sweeps there is more than a 25% reduction in the computational time.

5.1.4. Effect of full multigrid

Convergence of the solution, to the approximate level of the truncation error, can be accelerated by implementing full multigrid (FMG). The residual and lift coefficient (C_L) convergence histories for the 3-stage RK/implicit scheme with Roe dissipation and FMG are shown in Fig. 11. The calculation was done for Case 9 with the 320 × 64 grid using 4 levels of refinement, which contain 1, 2, 3, and 4 grids. After just 10 iterations on the single grid, multigrid was executed on each successive level for 100 cycles. This allows a CFL number of 1000 for finer levels. With 4 cycles on the final level the C_L is obtained to within 1% of the converged value. Only 10 cycles are required to get the lift and drag coefficients to 3 significant figures. As seen in Fig. 12 the surface pressure distribution at 10 cycles is nearly identical to the corresponding one at 100 cycles.

5.2. Three-dimensional wing flows

For the 3-D computations of flow around the ONERA M6 wing a single block C–O mesh topology containing a total of 192 × 48 × 32 cells (streamwise, normal and spanwise directions) was used. Matrix dissipation

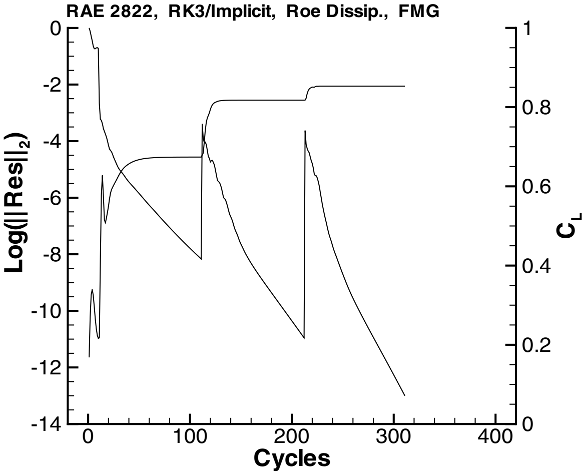


Fig. 11. Convergence history of RK3/implicit scheme with Roe dissipation and FMG: Case 9, 320 × 64 grid.

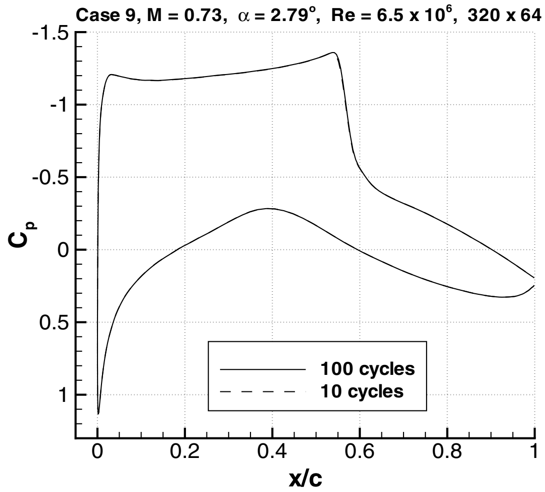


Fig. 12. Surface pressure distribution at 10 and 100 cycles for RK3/implicit scheme with Roe dissipation and FMG: Case 9, 320 × 64 grid.

was applied with the RK/implicit schemes. In Table 8, we compare the convergence rates of the standard and 5-stage RK/implicit schemes for a range of Mach numbers. The approximate computer time in minutes required (on a DEC UNIX single processor workstation) to reduce the residual 6 orders of magnitude is also

Table 8
Comparison of convergence rates of the RK/standard scheme and RK5/implicit scheme (with matrix dissipation) for the ONERA M6 wing at several Mach numbers

Mach no.	RK/Standard convergence rate	MG cycles	CPU time (min)	RK/implicit convergence rate	MG cycles	CPU time (min)
0.30	.988	950	491	.896	108	111
0.84	.982	734	380	.758	46	46
0.95	.981	698	363	.800	58	59
1.05	.978	609	315	.677	34	35
1.10	.978	604	312	.677	34	34

V-cycle with residual reduced 6 orders.

Table 9
Effect of number of RK stages and SGS sweeps on convergence of the RK/implicit scheme (with matrix dissipation) for the ONERA M6 wing

Scheme	SGS sweeps	ε	MG cycles	Convergence rate	CPU time (min)
RK/standard	–	–	4444	.995	1178
RK5/implicit	3	.45	166	.889	163
RK5/implicit	2	.45	168	.891	138
RK5/implicit	1	.45	243	.923	146
RK3/implicit	3	.45	180	.898	118
RK3/implicit	2	.45	167	.890	88
RK2/implicit	3	.65	195	.904	88
RK2/implicit	2	.5	189	.902	68

FMG with V-cycle and residual reduced 9 orders on fine grid.

given. Multigrid without FMG was used to accelerate the convergence. For the RK/implicit scheme the CFL number was gradually increased from 10 to 40 over the first 9 cycles, and then it was increased to 1000. In all cases, even when the oncoming flow is mildly supersonic, the RK5/implicit scheme converges significantly faster. At the lowest Mach number ($M_\infty = 0.3$) the RK/implicit scheme converges somewhat slower than it does at transonic speeds (when $M_\infty < 1.0$). However, as demonstrated by Rossow [35], this scheme with Roe dissipation exhibits the same convergence for low M_∞ (i.e. $\leqslant 0.3$) as it does for transonic Mach numbers, when the implicit and residual operators are preconditioned by modifying the sound speed (see Appendix A) in the numerical dissipation operators. Thus, one would anticipate a similar behavior with the matrix dissipation if the dissipation is modified appropriately.

If we consider all the cases of Table 8, the computational efficiency relative to the standard scheme is increased by factors between 4 and 9. In addition, the present RK5/implicit scheme exhibits better 3-D performance than observed previously [36]. This can be seen by considering the $M_\infty = 0.84$ case. For this case the residual is reduced 6 orders at a rate of 0.758, whereas the rate for previous results is 0.791. The improvement in convergence is due to the reduction of the entropy fix cutoff for the implicit operator.

As noted previously, the computational efficiency of the RK/implicit scheme is affected by both the number of RK stages and the number of SGS sweeps for approximately inverting the implicit operator. In Table 9, we present for the baseline ONERA M6 wing case ($M = 0.84$) the performance of the RK/implicit scheme with matrix dissipation as the number of RK stages and SGS sweeps is varied. The solutions were obtained using a

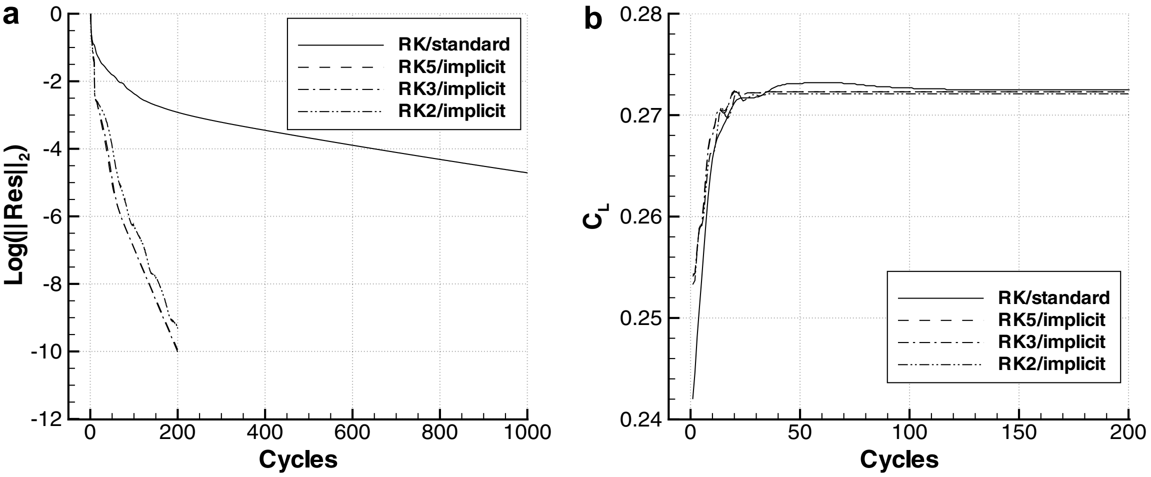


Fig. 13. Convergence histories, on the finest grid, of the RK/standard scheme and several RK/implicit schemes with FMG, matrix dissipation, and 2 SGS sweeps: ONERA M6 wing, 192 × 48 × 32 grid. (a) Residual and (b) lift coefficient.

V-cycle and FMG with 20, 20, and 200 multigrid cycles on 3 levels of refinement. For each calculation the L_2 norm of the residual was reduced at least 9 orders of magnitude, measured from the start of the level with the finest grid. Table 9 also includes the convergence rate (when a 9 order reduction on the finest mesh was reached) and the implicit parameter ε used on the finest mesh of each refinement level. On the coarse meshes an ε of 0.4 was used. The 2-stage scheme [30] with 2 SGS sweeps is the most efficient scheme with respect to computer time. However, it should be pointed out that in various 2-D applications the 3-stage scheme with 2 SGS sweeps has proven to be more robust. This scheme is more than 10 times faster than the standard scheme.

In Fig. 13, the residual and C_L histories on the solution grid of the 5-stage, 3-stage, and 2-stage RK/implicit schemes with 2 SGS sweeps and FMG are shown. The residual histories of the 5-stage and 3-stage schemes are essentially the same. Both the lift and drag coefficients are converged to within 0.1% in just 24 cycles on the fine grid.

6. Concluding remarks

A Runge–Kutta scheme preconditioned with a fully implicit operator has been implemented as a smoother for multigrid. The implicit operator extends the stability limit of the RK scheme, allowing the problem of geometric stiffness to be addressed. Fourier analysis has been performed to compare the smoothing properties of the RK/implicit scheme with those of a RK/standard scheme, which is used in many existing computer codes for solving the Euler and Navier–Stokes equations. The analysis has demonstrated that the RK scheme with a fully implicit operator exhibits much better damping behavior than the standard scheme.

In all applications considered a CFL number of 1000 has been used. The RK/implicit scheme has been applied with different dissipation operators, such as the Roe scheme, matrix dissipation, and the CUSP scheme. The amenability of the scheme to different forms of dissipation is quite important due to the wide usage of RK schemes accelerated by implicit residual smoothing and multigrid. The RK/implicit scheme can be easily implemented in these codes by replacing the scalar implicit operator with the fully implicit operator.

The performance of the RK/implicit scheme with different numerical dissipation formulations has been evaluated by solving the 2-D, Reynolds-Averaged Navier–Stokes (RANS) equations for three turbulent airfoil flow test cases, including a difficult transonic case with significant separation. Both 5-stage and 3-stage schemes have been considered. In addition, the effect of mesh aspect ratio on convergence has been investigated. With Roe dissipation the 3-stage RK/implicit scheme is 4–6.5 times faster (depending on the Reynolds number) than a RK/standard scheme, which is a well tuned 5-stage RK scheme with multigrid and scalar implicit residual smoothing. It should be emphasized that the RK/standard scheme has only 3 evaluations of the dissipation, all the characteristic fields are limited in the same way, and the residual smoothing is a scalar procedure. The RK3/implicit scheme with matrix dissipation is 3.8–5.5 times faster than the RK/standard scheme.

The RK/implicit scheme has also been used to solve the 3-D RANS equations for turbulent transonic flow over the ONERA M6 wing. Using matrix dissipation and 3 SGS sweeps to approximately invert the fully implicit operator, the RK5/implicit scheme is about 7 times faster than the RK/standard scheme. Additional improvement in computational efficiency has been demonstrated by reducing the number of stages and/or SGS sweeps. The three-stage RK/implicit with 2 SGS sweeps is roughly 10 times faster in reducing the residual by 9 orders of magnitude. We have not attempted to reduce the computational time at the expense of additional storage.

Acknowledgment

The authors would like to express their thanks to Dr. Veer Vatsa for helpful discussions related to the 3-D Navier–Stokes code TLNS3D and for some numerical tests.

Appendix A

The flux difference splitting dissipation expressed in terms of Mach number can be written as

$$\begin{bmatrix} \Delta F_{\rho} \\ \Delta F_{\rho E} \\ \Delta F_{\rho u} \\ \Delta F_{\rho v} \\ \Delta F_{\rho w} \end{bmatrix} = \beta \mathbf{D} \begin{bmatrix} \Delta \rho \\ \Delta p \\ \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} \tag{7.1}$$

with Δ indicating the difference between left and right states, and

$$\mathbf{D} = \begin{bmatrix}
|q_n| & \frac{a_1}{c} & n_x \rho M_0 & n_y \rho M_0 & n_z \rho M_0 \\
\frac{1}{2} q^2 |q_n| & \begin{array}{c} \frac{1}{\gamma-1}|q_n| \\ +\frac{H}{c} a_1 \\ +q_n M_0 \end{array} & \begin{array}{c} \rho u |q_n| \\ +n_x q_n \rho c a_1 \\ +n_x \rho H M_0 \end{array} & \begin{array}{c} \rho v |q_n| \\ +n_y q_n \rho c a_1 \\ +n_y \rho H M_0 \end{array} & \begin{array}{c} \rho w |q_n| \\ +n_z q_n \rho c a_1 \\ +n_z \rho H M_0 \end{array} \\
u|q_n| & \begin{array}{c} \frac{u}{c} a_1 \\ +n_x M_0 \end{array} & \begin{array}{c} n_x \rho u M_0 \\ +n_x n_x \rho c a_1 \\ +\rho |q_n| \end{array} & \begin{array}{c} n_y \rho u M_0 \\ +n_x n_y \rho c a_1 \end{array} & \begin{array}{c} n_z \rho u M_0 \\ +n_x n_z \rho c a_1 \end{array} \\
v|q_n| & \begin{array}{c} \frac{v}{c} a_1 \\ +n_y M_0 \end{array} & \begin{array}{c} n_x \rho v M_0 \\ +n_y n_x \rho c a_1 \end{array} & \begin{array}{c} n_y \rho v M_0 \\ +n_y n_y \rho c a_1 \\ +\rho |q_n| \end{array} & \begin{array}{c} n_z \rho v M_0 \\ +n_y n_z \rho c a_1 \end{array} \\
w|q_n| & \begin{array}{c} \frac{w}{c} a_1 \\ +n_z M_0 \end{array} & \begin{array}{c} n_x \rho w M_0 \\ +n_z n_x \rho c a_1 \end{array} & \begin{array}{c} n_y \rho w M_0 \\ +n_z n_y \rho c a_1 \end{array} & \begin{array}{c} n_z \rho w M_0 \\ +n_z n_z \rho c a_1 \\ +\rho |q_n| \end{array}
\end{bmatrix}, \tag{7.2}$$

where γ is the specific heat ratio, H is the specific total enthalpy, c is the speed of sound, and $a_1 = 1 - |M_0|$. The normal velocity $q_n = n_x u + n_y v + n_z w$, where the (n_x, n_y, n_z) are the components of the outward facing unit normal at a cell face. The cell face normal is scaled by β.

The positive and negative contributions to the matrix $\boldsymbol{P}_n$ are given by

$$\mathbf{P}_n^{+} = \frac{\beta}{2}(q_n + |q_n|)\mathbf{I} + \frac{\beta}{2} \begin{bmatrix} 0 & \frac{a_1}{c} & n_x \rho a_3 & n_y \rho a_3 & n_z \rho a_3 \\ 0 & (\gamma - 1)\frac{h}{c} a_1 & n_x \gamma p a_3 & n_y \gamma p a_3 & \gamma n_z p a_3 \\ 0 & n_x \rho^{-1} a_3 & n_x n_x c a_1 & n_x n_y c a_1 & n_x n_z c a_1 \\ 0 & n_y \rho^{-1} a_3 & n_y n_x c a_1 & n_y n_y c a_1 & n_y n_z c a_1 \\ 0 & n_z \rho^{-1} a_3 & n_z n_x c a_1 & n_z n_y c a_1 & n_z n_z c a_1 \end{bmatrix}, \tag{7.3}$$

$$\mathbf{P}_n^{-} = \frac{\beta}{2}(q_n - |q_n|)\mathbf{I} + \frac{\beta}{2} \begin{bmatrix} 0 & -\frac{a_1}{c} & n_x \rho a_2 & n_y \rho a_2 & n_z \rho a_2 \\ 0 & -(\gamma - 1)\frac{h}{c} a_1 & n_x \gamma p a_2 & n_y \gamma p a_2 & n_z \gamma p a_2 \\ 0 & n_x \rho^{-1} a_2 & -n_x n_x c a_1 & -n_x n_y c a_1 & -n_x n_z c a_1 \\ 0 & n_y \rho^{-1} a_2 & -n_y n_x c a_1 & -n_y n_y c a_1 & -n_y n_z c a_1 \\ 0 & n_z \rho^{-1} a_2 & -n_z n_x c a_1 & -n_z n_y c a_1 & -n_z n_z c a_1 \end{bmatrix}, \tag{7.4}$$

where $a_2 = 1 - M_0$, and $a_3 = 1 + M_0$. When scaling the numerical dissipation for low-speed flows, the speed of sound c appearing in the matrices $\mathbf{P}_n^{+}$ and $\mathbf{P}_n^{-}$ is replaced by the low-speed preconditioning speed of sound c', where

$$c' = \sqrt{\alpha^2 q^2 + M_r^2 c^2}, \tag{7.5}$$

with q denoting the flow speed, and M_r representing the reference Mach number, which is defined as

$$\alpha = \frac{1}{2}(1 - M_r^2), \quad M_r^2 = \min\left[\max\left(\frac{q^2}{c^2}, k\frac{q_\infty^2}{c_\infty^2}\right), 1\right]. \tag{7.6}$$

The parameter k is taken to be unity, and the subscript ∞ denotes free-stream condition. Although we use the modified sound speed c' in the computation of $\mathbf{P}_{\mathrm{n}}^{+}$ and $\mathbf{P}_{\mathrm{n}}^{-}$ in the present paper, this is not a requirement for effective performance of the RK/implicit algorithm [35].

In this paper we apply the thin-layer assumption to the Navier–Stokes equations. For 2-D applications this assumption means that only the viscous terms associated with the direction normal to a solid surface are retained, while for 3-D problems all viscous terms except the cross-derivative terms are retained. Thus, the viscous matrix for the implicit preconditioner is given by

$$\mathbf{B}_{\mathrm{n}} = \frac{f_{\mathrm{nd}}\beta^2}{\rho\mathscr{V}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\gamma\frac{\mu}{Pr}\frac{p}{\rho} & \gamma\frac{\mu}{Pr} & 0 & 0 & 0 \\ 0 & 0 & n_x n_x \mu^* + \mu & n_x n_y \mu^* & n_x n_z \mu^* \\ 0 & 0 & n_y n_x \mu^* & n_y n_y \mu^* + \mu & n_y n_z \mu^* \\ 0 & 0 & n_z n_x \mu^* & n_z n_y \mu^* & n_z n_z \mu^* + \mu \end{bmatrix}, \qquad (7.7)$$

where f_{nd} is a factor due to nondimensionalization of the governing flow equations and $\mu^* = \lambda + \mu$. By the Stokes' hypothesis $\lambda = -\frac{2}{3}\mu$, and the coefficient of viscosity $\mu = \mu_{\mathrm{l}} + \mu_{\mathrm{t}}$, where the subscripts l and t refer to laminar and turbulent values. The quantity Pr is the Prandtl number, which is determined by the sum $Pr_{\mathrm{l}} + Pr_{\mathrm{t}}$.

References

[1] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comput. 31 (1977) 333–390.

[2] A. Brandt, Multigrid techniques: 1984 guide with applications to fluid dynamics, GMD-Studie 85, GMD-FIT, 1985.

[3] W. Hackbush, Multi-grid Methods and Applications, Springer-Verlag, 1985.

[4] U. Trottenberg, C.W. Oosterlee, A. Schüller, Multigrid, Academic Press, 2001.

[5] A. Jameson, Solution of the Euler equations for two-dimensional, transonic flow by a multigrid method, Appl. Math. Comput. 13 (1983) 327–356.

[6] A. Jameson, Multigrid algorithms for compressible flow calculations, in: W. Hackbush, U. Trottenberg (Eds.), Second European Conference on Multigrid Methods, Cologne, 1985, Lecture Notes in Mathematics, vol. 1228, Springer-Verlag, Berlin, 1986.

[7] A. Jameson, D.A. Caughey, How many steps are required to solve the Euler equations of steady, compressible flow: in search of a fast solution algorithm, AIAA Paper 2001-2673, 2001.

[8] A. Jameson, The evolution of computational methods in aerodynamics, J. Appl. Mech. 50 (4b) (1983) 1052–1070.

[9] V.N. Vatsa, B.W. Wedan, Development of an efficient multigrid code for 3-d Navier–Stokes equations, AIAA Paper 89-1791, 1989.

[10] R. Radespiel, C.-C. Rossow, R.C. Swanson, An efficient cell-vertex multigrid scheme for the three-dimensional Navier–Stokes equations, AIAA J. 28 (8) (1990) 423–459.

[11] W. Mulder, A new multigrid approach to convection problems, J. Comput. Phys. 83 (1989) 303–323.

[12] R. Radespiel, R.C. Swanson, Progress with multigrid schemes for hypersonic flow problems, J. Comput. Phys. 116 (1995) 103–122.

[13] N.A. Pierce, M.B. Giles, Preconditioned multigrid methods for compressible flow calculations on stretched meshes, J. Comput. Phys. 136 (1997) 425–445.

[14] J. Blazek, Verfahren zur Beschleunigung der Lösung der Euler- und Navier–Stokes-Gleichungen bei stationären Über- und Hyperschallströmungen, Ph.D. thesis, Technical University of Braunschweig, Braunschweig, Germany, 1995.

[15] E. Turkel, V. Vatsa, V. Venkatakrishnan, Uni-directional implicit acceleration techniques for compressible Navier–Stokes solvers, AIAA Papser 99-3265, 1999.

[16] D. Mavriplis, Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes, J. Comput. Phys. 145 (1998) 141–165.

[17] D. Mavriplis, On convergence acceleration techniques for unstructured meshes, AIAA Paper 98-2966, 1998.

[18] C.-C. Rossow, Convergence acceleration for solving the compressible Navier–Stokes equations, AIAA J. 44 (2006) 345–352.

[19] P.J. Van der Houwen, Construction of Integration Formulas for Initial Value Problems, North Holland, 1977.

[20] R.C. Swanson, E. Turkel, Multistage schemes with multigrid for Euler and Navier–Stokes equations, NASA TP 3631, 1997.

[21] L. Martinelli, Calculations of viscous flow with a multigrid method, Ph.D. thesis, Princeton University, 1987.

[22] P.L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, J. Comput. Phys. 43 (1981) 357–372.

[23] C.-C. Rossow, A flux splitting scheme for compressible and incompressible flows, J. Comput. Phys. 164 (2000) 104–122.

[24] E. Turkel, V.N. Vatsa, R.C. Swanson, C.-C. Rossow, Convergence acceleration for the three dimensional compressible Navier–Stokes equations, in: Proceedings of European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands, September 5–8, 2006.

[25] A. Jameson, Analysis and design of numerical schemes for gas dynamics I: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence, Int. J. Comput. Fluid Dyn. 4 (1995) 171–218.

[26] R.C. Swanson, R. Radespiel, E. Turkel, On some numerical dissipation schemes, J. Comput. Phys. 147 (1998) 518–544.

[27] R.C. Swanson, E. Turkel, On central difference and upwind schemes, J. Comput. Phys. 101 (1992) 292–306.

[28] A. Jameson, W. Schmidt, E. Turkel, Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes, AIAA Paper 8l-1259, 1981.

[29] T.W. Roberts, R.C. Swanson, A study of multigrid preconditioners using eigensystem analysis, AIAA Paper 2005-5229, 2005.

[30] B. Van Leer, C.-H. Tai, K.G. Powell, Design of optimally smoothing multi-stage schemes for the Euler equations, AIAA Paper 89-1933, 1989.

[31] B.S. Baldwin, H. Lomax, Thin layer approximation and algebraic model for separated flows, AIAA Paper 78-257, 1978.

[32] P.H. Cook, M.A. McDonald, M.C.P. Firmin, Aerofoil RAE 2822 pressure distributions and boundary layer and wake measurements, AGARD-AR-138, 1979.

[33] V. Schmitt, F. Charpin, Pressure distributions on the ONERA-M6-wing at transonic Mach numbers, AGARD-AR-138, 1979 (Chapter B-1).

[34] J. Faßbender, Improved robustness of numerical simulation of turbulent flows around civil transport aircraft at flight Reynolds numbers, Ph.D. thesis, Technical University of Braunschweig, Braunschweig, Germany, 2004.

[35] C.-C. Rossow, Efficient computation of compressible and incompressible flows, J. Comput. Phys. 220 (2007) 879–899.

[36] R.C. Swanson, E. Turkel, C.-C. Rossow, V.N. Vatsa, Convergence acceleration for multistage time-stepping schemes, AIAA Paper 2006-3525, 2006.