

Convergence Acceleration for Multistage Time-Stepping Schemes

R. C. Swanson*, E. Turkel†, C.-C. Rossow‡, V. N. Vatsa§

*,[§] *NASA Langley Research Center, Hampton, VA
Hampton, VA 23681-2199, USA*

† *Tel-Aviv University, Israel*

‡ *DLR, Deutsches Zentrum für Luft- und Raumfahrt
Lilienthalplatz 7 D-38108 Braunschweig, Germany*

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 could be used. The implicit preconditioner addresses the stiffness in the discrete equations associated with stretched meshes. Numerical dissipation operators (based on the Roe scheme, a matrix formulation, and the CUSP scheme) as well as the number of RK stages are considered in evaluating the RK/implicit scheme. 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. In two dimensions, turbulent flows over an airfoil at subsonic and transonic conditions are computed. The effects of mesh cell aspect ratio on convergence are investigated for Reynolds numbers between 5.7×10^6 and 100.0×10^6 . Results are also obtained for a transonic wing flow. For both 2-D and 3-D problems, it is demonstrated that the computational time of a well-tuned standard RK scheme can be reduced at least a factor of four.

Introduction

The field of computational fluid dynamics has expanded rapidly over the last 25 to 30 years, and fluid dynamics 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 minimum near term objective one should seek comparable efficiency to that for the Euler equations. A major obstacle in achieving such a goal is 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. In addition, we are also confronted with the dilemma of how to improve computational efficiency and at the same time minimize computer storage required, especially in 3-D simulations.

One powerful solution strategy for solving large scale problems in fluid dynamics is multigrid.¹⁻³ The multigrid approach offers the possibility of solving discrete partial differential equations with grid independent

*Senior Research Scientist, Computational AeroSciences Branch, Mail Stop 128, Senior Member AIAA.

†Professor, School of Mathematical Sciences.

‡Director, Institut für Aerodynamik und Strömungstechnik, Senior Member AIAA.

§Senior Research Scientist, Computational AeroSciences Branch, Mail Stop 128, Senior Member AIAA.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.2006

convergence rates. Although most of the theory developed for multigrid is for elliptic problems, effective multigrid solvers^{4,5} have been constructed for the Euler equations, which are hyperbolic in time. In fact, Jameson and Caughey⁶ demonstrated that an Euler solution for airfoil flows could be obtained in 3–5 multigrid cycles (converged to the level of the truncation error). Such multigrid methods 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. Moreover, it is the coarser grids that 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 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 implicit residual smoothing⁷ 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.^{8,9} However, convergence rates exceeding 0.99 are encountered when solving turbulent viscous flows.

For viscous flow problems the anisotropy due to grid cell aspect ratio causes a low effectiveness in 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 technique involves semi-coarsening, where a family of coarse grids is generated by coarsening in one direction at a time rather than all directions. Mulder¹⁰ introduced this type of coarsening in order 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.^{11,12} 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. With this in mind efforts were made to improve the performance of the implicit residual smoothing being used in conjunction with Runge-Kutta schemes. As an initial step the simple diffusion operator in this implicit process was replaced with a convection operator that includes flux Jacobians.¹³ Since approximate factorization was used to facilitate the inversion of the implicit operator, there was still a strong limitation on the actual time step allowed due to the factorization error. To reduce the complexity of the operator, as well as to eliminate the factorization error, a directional smoothing was developed,¹⁴ where smoothing was performed in only the wall normal direction (i.e., j-line smoothing). With this approach the time step still was limited due to 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. However, using unstructured grids where there is a flexible database structure, Mavriplis¹⁵ combined j-line coarsening and j-line smoothing along with Jacobi preconditioning to demonstrate cell aspect ratio independent convergence rates for two dimensional, turbulent, viscous airfoil flow computations.

The directional methods just described 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.¹⁶ Furthermore, there is still a significant Courant-Friedrichs-Lewy (CFL) restriction (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 of bound of the RK scheme, Rossow¹⁷ introduced a fully implicit operator into the implicit residual smoothing procedure. This RK/implicit scheme requires 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 a symmetric Gauss-Seidel iteration. The Roe scheme was used to approximate 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 and extend it to three dimensions. Initially,

the flexibility of the scheme is investigated by considering the effect of choosing an alternative numerical dissipation operator for the residual function. 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 also considered. At the highest Reynolds number the maximum grid cell aspect ratio exceeds 50,000. To assess the scheme in three dimensions viscous transonic flow over the ONERA M6 wing is computed. For all calculations the convergence behavior and computational effort with the scheme are discussed.

Governing Equations

In this paper 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

$$\iiint_{\mathcal{V}} \frac{\partial \mathbf{W}}{\partial t} d\mathcal{V} + \iint_{\mathcal{S}} \mathcal{F} \cdot \mathbf{n} d\mathcal{S} = 0, \quad (1)$$

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

$$\mathcal{F}_c = \begin{bmatrix} \rho \mathbf{q} \\ \rho u \mathbf{q} + p \mathbf{e}_x \\ \rho v \mathbf{q} + p \mathbf{e}_y \\ \rho w \mathbf{q} + p \mathbf{e}_z \\ \rho H \mathbf{q} \end{bmatrix}, \quad \mathcal{F}_v = \begin{bmatrix} 0 \\ \bar{\tau} \cdot \mathbf{e}_x \\ \bar{\tau} \cdot \mathbf{e}_y \\ \bar{\tau} \cdot \mathbf{e}_z \\ \bar{\tau} \cdot \mathbf{q} - \mathbf{Q} \end{bmatrix} \quad (2)$$

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

$$\bar{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix}, \quad \mathbf{Q} = k \begin{bmatrix} \partial T / \partial x \\ \partial T / \partial y \\ \partial T / \partial z \end{bmatrix} \quad (3)$$

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

In order to close the the system given by Eq. (1) we use the equation of state

$$p = \rho R T \quad (4)$$

where R is the specific gas constant.

Numerical Algorithms

In this section we first briefly describe the standard solution scheme for solving the compressible Navier-Stokes equations. Variations of this scheme are considered based on the choice of the numerical dissipation scheme. A principal component of the standard scheme is implicit residual smoothing, which provides additional support for the basic iterative scheme, and thus, allows extended stability. A modification of this component of the standard scheme is the basis for an alternative scheme that allows dramatically improved convergence rates. This alternative formulation is discussed in the second part of this section.

RK/Standard Scheme

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

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

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 RK stage, and the RK coefficients are given by

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

For convenience we have omitted the indices referring to the grid points. The residual function \mathbf{R} is defined by

$$\mathbf{R} = \mathbf{R}(\mathbf{W}^{(q)}) = \frac{1}{\mathcal{V}} \left[\mathcal{L}_c \mathbf{W}^{(q)} + \sum_{r=0}^q \gamma_{qr} \mathcal{L}_v \mathbf{W}^{(r)} + \sum_{r=0}^q \gamma_{qr} \mathcal{L}_d \mathbf{W}^{(r)} \right], \tag{6}$$

and the operators \mathcal{L}_c , \mathcal{L}_v , and \mathcal{L}_d relate to the convection, viscous, and numerical dissipation terms. Terms in the governing equations are approximated with central differencing. The coefficients γ_{qr} are the weights of the viscous and dissipation terms on each stage (see Ref. 19), which are taken to be [1, 0, 0.56, 0, 0.44]. Such a scheme is frequently designated as a (5, 3) scheme, since the dissipation terms are evaluated only on three stages.

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

$$(1 - \beta_\xi \delta_\xi^2)(1 - \beta_\eta \delta_\eta^2)(1 - \beta_\zeta \delta_\zeta^2) \bar{\mathbf{R}} = \mathbf{R}, \tag{7}$$

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 Martinelli,¹⁸ Swanson and Turkel¹⁹). After inverting the product operator in Eq. 7 we substitute $\bar{\mathbf{R}}$ for \mathbf{R} in Eq. 5. For the inversion tridiagonal solves are performed in each coordinate direction.

One can view the implicit residual smoothing as a preconditioner. 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.¹⁹ In evaluating the resulting scheme one must also consider the improved stability of the scheme, which allows faster error propagation in the course grid process of multigrid.

In the standard scheme the full approximation scheme (FAS) is applied to the 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 k -th grid. Also, 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 $\bar{\mathbf{W}}_k$ is the current solution on grid k , the residual on this grid is $\mathbf{R}_k \equiv \mathbf{f}_k - \mathcal{L}_k \bar{\mathbf{W}}_k$. This leads to the coarse-grid equation

$$\mathcal{L}_{k-1} \mathbf{W}_{k-1} = \mathbf{f}_{k-1} = -I_k^{k-1} \mathbf{W}_k + \mathcal{L}_{k-1} (I_{k-1}^{k-1} \bar{\mathbf{W}}_k). \tag{8}$$

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

$$\mathbf{W}_k \leftarrow \bar{\mathbf{W}}_k + I_{k-1}^k (\mathbf{W}_{k-1} - I_k^{k-1} \bar{\mathbf{W}}_k). \tag{9}$$

Equation (8) is solved by applying the same relaxation procedure that is used to solve the fine-grid equation. 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 Jameson⁵ for a cell-centered, finite-volume scheme. The residual and solution restriction operators are defined 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 interpolation operator. A conventional V - or W -cycle is used to execute the multigrid process.

Discretization and Dissipation

Using the finite volume technique for spatial discretization, Eq. (1) can be written in semidiscrete form as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{1}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{F}_n S = 0, \quad (10)$$

where \mathbf{F}_n is the normal flux density vector at the cell face, now \mathcal{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}_c can be expressed as

$$\mathbf{F}_c = \frac{1}{2}(\mathbf{F}^L + \mathbf{F}^R) + \mathbf{D}, \quad (11)$$

where \mathbf{F}^L and \mathbf{F}^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. This particular form results in a central difference approximation plus numerical dissipation. A 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,²¹ and it can be written as

$$D = -\frac{1}{2}|\mathbf{A}|(\mathbf{W}^R - \mathbf{W}^L) \quad (12)$$

$$= -\frac{1}{2}|\mathbf{A}|\Delta\mathbf{W}, \quad (13)$$

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.²² The Mach number M_0 is given by

$$M_0 = \min(|M|, 1) \text{ sign}(M). \quad (14)$$

The resulting form for $|\mathbf{A}|\Delta\mathbf{W}$ is given in the Appendix. For second order accuracy the symmetric limited positive (SLIP) scheme of Jameson²³ is used following the implementation of Swanson, Radespiel, and Turkel.²⁴

Another dissipation formulation being considered is closely related to that of the Roe scheme. It is generally called matrix dissipation (see Swanson and Turkel²⁰). There are two principal differences between the Roe scheme dissipation and the matrix dissipation. First, a switching function based on pressure is used to change from third order dissipation (second order scheme) and first order dissipation (e.g., first order scheme in neighborhood of shock). Second, there are lower bounds imposed on both the convective and acoustic eigenvalues of the Jacobian matrix.

The third dissipation scheme is the convective upwind and split pressure (CUSP) scheme. Jameson²³ designed this scheme such that it can support single interior point discrete shock waves. For this scheme the dissipation flux can be written as

$$D = -\frac{1}{2}\bar{\nu}\Delta\mathbf{W} - \frac{1}{2}\beta\Delta\mathbf{F}, \quad (15)$$

with $\bar{\nu}$ and β being parameters determined such that single interior point shocks are permitted. Discussion and analysis of the CUSP scheme are given in Ref. 24. We note that these are all upwind type schemes. The original JST algorithm²⁵ used a scalar dissipation which behaves more as a central difference scheme.

RK/Implicit Scheme

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

$$\mathbf{W}^{(q)} = \mathbf{W}^{(0)} + \delta\mathbf{W}^{(q)}, \quad (16)$$

where

$$\delta\mathbf{W}^{(q)} = \mathbf{W}^{(q)} - \mathbf{W}^{(0)} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \mathcal{L} \mathbf{W}^{(q-1)} = \widehat{\mathbf{R}}(\mathbf{W}^{(q-1)}) \quad (17)$$

and \mathcal{L} is the complete difference operator given in Eq. (6). To extend the support of the difference scheme we consider implicit residual smoothing. Then, we have the following:

$$\mathcal{L}_i \overline{\delta \mathbf{W}} = \delta \mathbf{W}^{(q)}, \quad (18)$$

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

$$\overline{\delta \mathbf{W}} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \mathcal{P} \mathcal{L} \mathbf{W}^{(q-1)}, \quad (19)$$

where \mathcal{P} is a preconditioner defined by the approximate inverse $\tilde{\mathcal{L}}_i^{-1}$. The change $\overline{\delta \mathbf{W}}$ is used to replace the explicit update appearing in Eq. (16). Thus each stage in the RK scheme is preconditioned by an implicit operator.

Unlike the standard scheme, which uses a diffusion operator for \mathcal{L}_i , we consider a fully implicit operator. A first order upwind approximation based on the Roe Scheme is used for the spatial derivatives in the implicit operator. To derive this operator one treats the spatial discretization terms in Eq. (10) implicitly and applies linearization. For a detailed derivation see Rossow.¹⁷ Substituting for the implicit operator in Eq. (18), we obtain for the q th stage of the RK scheme

$$\left[I + \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{A}_n S \right] \overline{\delta \mathbf{W}} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{F}_n^{(q-1)} S = \widehat{\mathbf{R}}^{(q-1)}, \quad (20)$$

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

$$\mathbf{A}_n^+ = \frac{1}{2} (\mathbf{A}_n + |\mathbf{A}_n|), \quad \mathbf{A}_n^- = \frac{1}{2} (\mathbf{A}_n - |\mathbf{A}_n|). \quad (21)$$

If we substitute for \mathbf{A}_n in Eq. (20) using the definitions of Eq. (21), then (including the grid indices for the purpose of discussion) the implicit scheme can be written as

$$\left[I + \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{A}_n^+ S \right] \overline{\delta \mathbf{W}}_{i,j,k} = \widehat{\mathbf{R}}_{i,j,k}^{(q-1)} - \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{A}_n^- \overline{\delta \mathbf{W}}_{NB} S, \quad (22)$$

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,¹⁷ the quantity $\mathbf{A}_n^- \overline{\delta \mathbf{W}}$ 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 $\mathbf{A}_n^+ \overline{\delta \mathbf{W}}$ 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. (22) for the changes in conservative variables $\overline{\delta \mathbf{W}}_{i,j,k}$, the 5×5 matrix (a 4×4 matrix in two dimensions) on the left-hand side of Eq. (22) 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. Alternative iterative methods such as red-black Gauss-Seidel could also be used (see Ref. 26), which would allow the RK/implicit scheme to be applied on unstructured grids.

To efficiently evaluate the Jacobian matrices \mathbf{A}_n^+ and \mathbf{A}_n^- we rely upon their forms when expressed in terms of the cell interface Mach number M_0 . For simplification, we transform Eq. (20) to the set of primitive variables $[\rho \ p \ u \ v \ w]^T$. Thus,

$$\left[I + \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{P}_n S \right] \overline{\delta \mathbf{U}} = -\frac{\partial \mathbf{U}}{\partial \mathbf{W}} \alpha_q \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{F}_n^{(q-1)} S, \quad (23)$$

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

$$\mathbf{P}_n = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \mathbf{A}_n \frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} (\mathbf{A}_n^+ + \mathbf{A}_n^-) \frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \mathbf{P}_n^+ + \mathbf{P}_n^-. \quad (24)$$

Note that the Jacobian $\partial \mathbf{U} / \partial \mathbf{W}$ on the right-hand side of Eq. (23) must multiply the conservative flux balance in order to ensure conservation. Using the definitions of Eq. (21) and the dissipation matrix in terms of the Mach number defined in the Appendix, one can determine the matrices \mathbf{P}_n^+ and \mathbf{P}_n^- , which are

also given in the Appendix. 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 the definitions presented in Ref. 27 for primitive variables.

Due to the upwind approximation used for the implicit operator, the coefficients for the RK scheme are also based on an upwind scheme. Unless indicated otherwise, a (5,5) RK scheme (numerical dissipation evaluated on all stages) is employed for all dissipation formulations considered for the residual function, and the RK coefficients are given by

$$[\alpha_1, \dots, \alpha_5] = [0.0695, 0.1602, 0.2898, 0.5060, 1.0].$$

These coefficients were obtained from Ref. 31.

A summary of the implementation of the RK/implicit is as follows. In the first step, the explicitly evaluated residuals for a RK stage are transformed to residuals in primitive variables to form the right-hand side of Eq. (23). Next, we approximately invert the implicit operator with symmetric Gauss-Seidel. This approximate inversion yields new residuals in primitive variables, which must be 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.

Numerical Results

The RK/implicit scheme described previously 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.³² In this section the airfoil solutions are compared with the experimental data of Cook, McDonald and Firmin.²⁸ The cases considered are as follows: Case 1, Case 9, Case 10. The flow conditions for these cases are given in Table 1. For Case 1 the flow is primarily subsonic with a relatively small region of supersonic flow. With 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 convergence rate. For the wing flow the calculated solution is compared with the experimental data of Schmitt and Charpin.²⁹ This particular case is a supercritical flow, and the free-stream Mach number M_∞ is 0.84, the angle of attack α is 3.06 degrees, and the Reynolds number Re is 11.7×10^6 . For 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 to 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 2,413. In order to investigate the performance of the RK/implicit scheme for a range of Reynolds numbers (between 5.7×10^6 and 100.0×10^6) a family of C-type meshes was generated.³⁰ 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 being considered the same boundary conditions were imposed. On the surface the no-slip condition was applied. At the outer boundary Riemann invariants were used to apply the boundary conditions. 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. 19. For the 3-D case the boundary conditions were essentially the same except the far-field vortex effect was not included. In all the calculations that start on the desired solution grid, the initial solution was 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.

Two-Dimensional Results

In Figs. 1–3 the surface pressure distributions obtained for the three test cases are displayed. The computed lift and drag coefficients are presented in Table 2. These results were calculated using the Roe scheme dissipation for both the implicit operator and the residual function (right-hand side). The pressure distributions of Case 1 and Case 9 agree fairly well with the experimental data. For Case 10 the well-known inadequacy of the Baldwin-Lomax turbulence model in predicting the data is evident. Solutions for these cases obtained using the matrix dissipation formulation and also the HCUSP scheme (a form of the CUSP scheme) for evaluating the residual function are similar to those shown (for a direct comparison see Ref. 24). In the subsequent discussion on the RAE cases we focus on the performance of the RK/implicit scheme with the three dissipation forms.

Figures 4–6 compare the convergence histories of the standard RK and RK/implicit schemes for the Roe scheme dissipation. In each case the residuals are normalized by the corresponding residual of the first iteration. Throughout the paper the number after RK indicates the number of stages. These histories indicate 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 for the residual function is defined by matrix dissipation, and for the RK/implicit scheme the dissipation operator is based on the Roe Scheme. In the calculations with the RK/implicit scheme the CFL number is 16 during the first 8 multigrid cycles; and then, it is increased to 160. From the figures one can see that the RK/implicit scheme requires more than a factor of 13 fewer multigrid cycles than the standard scheme to reduce the residual 13 orders of magnitude. The convergence rates for the standard scheme exceed 0.98 (see Tables 3–5). Here, and subsequently, convergence rate means average rate of reduction of the residual. Even for the more difficult Case 10 the convergence rate for the RK/implicit scheme is below 0.83. With respect to computational effort, the RK/implicit scheme is about 2 times faster than the standard scheme.

In Figs. 7–8 the convergence histories for the RK/implicit scheme with matrix and HCUSP dissipation are shown for all the cases. For these forms of dissipation the cycle reduction factor is between 11 and 12, and the convergence rates are similar. As indicated in Tables 3–5, with the two dissipation forms the RK/implicit is roughly between 1.6 and 1.9 times faster in computer time than the standard scheme. The computational savings with the Roe scheme dissipation is only somewhat better than it is with the matrix dissipation. This is because the matrix dissipation does not have the additional expense of evaluating a limiter for each characteristic field. Nevertheless, the convergence rates exhibited with the RK/implicit scheme using the dissipation of the Roe scheme are noticeably faster.

The results presented thus far were obtained on a grid with moderately high aspect ratio cells. In order to investigate how the RK/implicit scheme alleviates stiffness associated with high aspect ratio cells, we now consider calculations that were performed when the Reynolds number was varied by more than an order of magnitude. The computational meshes, which were described previously, are the same as those used by Faßbender³⁰ 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 Figs. 9 and 10 convergence histories are presented for $Re = 5.7 \times 10^6$ to $Re = 100.0 \times 10^6$. The maximum surface grid cell aspect ratio varies from 3,949 to 50,260. From the dashed line convergence curves in Fig. 9 we see that the number of multigrid cycles only increases by a factor less than 2.5 over the Re range. In addition, we see the improved convergence indicated by the solid lines when the low-speed preconditioning speed of sound (denoted by c') is introduced into the numerical dissipation of the implicit operator. If the same c' is used in determining the dissipation in the residual function, then the scheme with Roe dissipation can be applied with comparable efficiency to low Mach number ($M \leq 0.1$) flows. For further discussion of this modification (i.e., scaling) of the dissipation see Appendix and Ref. 33. This scaling of the dissipation allowed the CFL number to be increased from 160 to 1000. The convergence rates and computing times for these calculations are displayed in Tables 6–9. Results annotated with prec indicate that the numerical dissipation is scaled appropriately by c' . For the higher Re values the convergence rate with the standard scheme exceeds 0.995. Using the Roe scheme dissipation, the RK/implicit scheme converges at a rates between 0.88 and 0.9 for the higher Re values (i.e., at higher cell aspect ratios). The corresponding reduction in computational expense is about a factor of 4 relative to the standard scheme.

Before proceeding we need to discuss further the use of c' in the implicit operator, which allowed a much larger CFL number. For most of the present work we could not explain why such a modification in the

implicit operator allowed an increase in the CFL number, especially since the c' and the standard speed of sound are nearly the same for transonic flows. Thus, there should not have been any special benefit from introducing c' in the implicit operator. Only more recently, through the work of Rossow,³³ did we discover that this behavior is a consequence of using a different initialization for calculating the approximate inverse of the implicit operator when the c' scaling is used. In particular, we set the initial guess in updating the residual function to zero when using this scaling. Otherwise, we initialized with the explicit residual function. If, instead, we always apply the zero initialization, then the scheme without c' works at CFL = 1000. However, by using the formulation with c' , and modifying the numerical dissipation of the explicit (right-hand side) residual function of Eq. (22) in the same manner, we derive the additional benefit of being able to also compute low-speed flows accurately and efficiently when Roe dissipation is used.³³ If other forms of dissipation are used, appropriate modifications of the right-hand side should be possible, allowing those forms to also be used for efficiently computing low-speed flows.

There are several alternative ways to make the gains in computational savings significantly greater. One approach is to reduce the number of stages in the RK scheme. With such an approach one would expect almost no loss in numerical efficiency (even though reducing the number of stages reduces the allowable time step), since it should be possible to use a CFL number of 1000. Calculations were performed for Cases 1, 9, and 10 with the RK/implicit scheme using 3 stages, CFL = 1000, and the c' scaling of the dissipation. The RK coefficients for the 3-stage scheme are [0.15, 0.40, 1.0]. Solutions were determined with the same (moderately high aspect ratio) 320×64 grid used for the results presented in Figs. 4–6. Figures 11 and 12 show the convergence histories with Roe and matrix dissipation, respectively. As indicated in Tables 3–9 the convergence rates with the 5-stage and 3-stage schemes are nearly the same. In addition, the RK/implicit scheme with Roe and matrix dissipation is about 4–6.5 and 3.5–5.5 times faster, respectively, (depending on the Reynolds number) than the RK/standard scheme. Further savings in computational expense are anticipated by reducing the number of RK stage evaluations of the numerical dissipation and/or by reducing the number of symmetric G-S sweeps for approximately inverting the implicit operator. For example, Rossow³³ demonstrates that even with the convergence rate penalty produced by performing only one G-S sweep rather than three sweeps there is more than a 25% reduction in the computational time.

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 convergence histories for the 3-stage RK/implicit scheme with Roe dissipation and FMG are shown in Figs. 13 and 14. 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 performing just 10 iterations on the single grid, multigrid was executed on each successive level for 100 cycles. This procedure allows the CFL number to be 1000 for levels 2–4. With 4 cycles on the final level the lift coefficient 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 Figs. 15 and 16 the surface pressure and skin-friction distributions are nearly identical to the corresponding distributions at 100 cycles.

Three-Dimensional Results

For the 3-D computation a C-O mesh topology was used. Computations were performed on a single block mesh containing a total of $192 \times 48 \times 32$ cells (streamwise, normal, and spanwise directions). Matrix dissipation was used with the RK/implicit schemes. In Figs. 17 and 18 the computed pressure distributions at four spanwise locations are compared with experimental data. There is fairly good agreement with the data using the Baldwin-Lomax turbulence model. Figs. 19 and 20 show the residual and lift coefficient (C_L) histories for calculations with the standard and both 5-stage and 3-stage RK/implicit schemes. All results were obtained without FMG. There is a dramatic improvement in the residual convergence using the RK/implicit scheme. For the RK5/implicit and RK3/implicit schemes the residual is reduced 12 orders of magnitude in 284 and 280 cycles, respectively (convergence rate of about 0.92), whereas the standard scheme requires 5,448 cycles (convergence rate of 0.995). With the RK3/implicit scheme the lift and drag coefficients are converged to 3 significant digits in 30 cycles, but this can be significantly improved by applying FMG. The resulting computational savings for the 5-stage RK/implicit scheme is a factor of 2.5, and for the 3-stage scheme it is a factor of 4.2.

Solutions were also obtained with FMG, which contained 3 levels of refinement, with the first level being a single grid. On each level 100 iterations (or cycles) were done. Figures 21 and 22 show the residual and lift coefficient histories. The RK5/standard scheme requires 324 cycles on the fine grid to reduce the residual 4 orders of magnitude, and 607 cycles for a 5 orders reduction. With the RK/implicit schemes the residual is

decreased 4 orders in only 22 cycles and 5 orders in 37 cycles. In addition, both the lift and drag coefficients are converged to within 0.1% in just 20 cycles on the fine grid.

Concluding Remarks

A Runge-Kutta scheme preconditioned with a fully implicit operator has been implemented as a smoother for multigrid. The implicit operator allows the problem of geometric stiffness to be addressed and extends the stability limit of the RK scheme. By appropriate initialization of the Gauss-Seidel iterative process for approximating the inverse of the implicit operator, the allowable CFL number has been increased to 1000. This RK/implicit scheme has been applied with different dissipation operators, such as the Roe scheme, matrix formulation, and the CUSP scheme. The amenability of the scheme to different forms of dissipation is quite important since many existing computer codes for solving the Euler and Navier-Stokes equations use 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 AGARD 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 the 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 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 3.5–5.5 times faster than the RK5/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 the RK3/implicit scheme is 4.2 times faster than the RK5/standard scheme.

At this point, in an effort to minimize the additional storage for the flux Jacobians of the RK/implicit scheme, especially for 3-D applications, we have not attempted to reduce the computational time at the expense of additional storage.

Appendix

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

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

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

$$\mathbf{D} = \begin{bmatrix} |q_n| & c^{-1}a_1 & \rho M_0 n_x & \rho M_0 n_y & \rho M_0 n_z \\ \frac{1}{2}q^2|q_n| & \frac{1}{\gamma-1}|q_n| + c^{-1}Ha_1 & \rho|q_n| + q_n\rho ca_1 n_x & \rho|q_n| + q_n\rho ca_1 n_y & \rho|q_n| + q_n\rho ca_1 n_z \\ & +q_n M_0 & +\rho H M_0 n_x & +\rho H M_0 n_y & +\rho H M_0 n_z \\ u|q_n| & c^{-1}ua_1 + M_0 n_x & \rho|q_n| + \rho u M_0 n_x & \rho u M_0 n_y & \rho u M_0 n_z \\ & & +\rho ca_1 n_x n_x & +\rho ca_1 n_x n_y & +\rho ca_1 n_x n_z \\ v|q_n| & c^{-1}va_1 + M_0 n_y & \rho v M_0 n_x & \rho|q_n| + \rho v M_0 n_y & \rho v M_0 n_z \\ & & +\rho ca_1 n_y n_x & +\rho ca_1 n_y n_y & +\rho ca_1 n_y n_z \\ w|q_n| & c^{-1}wa_1 + M_0 n_z & \rho w M_0 n_x & \rho w M_0 n_y & \rho|q_n| + \rho w M_0 n_z \\ & & +\rho ca_1 n_z n_x & +\rho ca_1 n_z n_y & +\rho ca_1 n_z n_z \end{bmatrix},$$

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 \mathbf{P} are given by

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

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

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

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

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

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

The parameter k is taken to be unity, and the subscript ∞ denotes free-stream condition.

References

- ¹Brandt, A., "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," GMD-Studie 85, GMD-FIT, 1985.
- ²Hackbush, W., "Multi-grid Methods and Applications," Springer-Verlag, 1985.
- ³Trottenberg, U., Oosterlee, C. W., Schüller, A., "Multigrid," Academic Press, 2001.
- ⁴Jameson, A., "Solution of the Euler Equations for Two-Dimensional, Transonic Flow by a Multigrid Method," *Appl. Math. Comput.*, Vol. 13, 1983, pp. 327–356.
- ⁵Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," in *Second European Conference on Multigrid Methods, Cologne, 1985*, editors: W. Hackbush and U. Trottenberg, [Lecture Notes in Mathematics, Vol. 1228], Springer-Verlag, Berlin, 1986.
- ⁶Jameson, A. and Caughey, D. A., "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, June 2001.
- ⁷Jameson, A., "The Evolution of Computational Methods in Aerodynamics," *J. Appl. Mech.*, Vol. 50, No. 4b, 1983, pp. 1052–1070.
- ⁸Vatsa, V. N. and Wedan, B. W., "Development of an Efficient Multigrid Code for 3-D Navier-Stokes Equations," *AIAA Paper* 89-1791, June 1989.
- ⁹Radespiel, R., Rossow, C.-C., and Swanson, R. C., "An Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations," *AIAA J.*, Vol. 28, No. 8, 1990, pp. 423–459.
- ¹⁰Mulder, W., "A New Multigrid Approach to Convection Problems," *J. Comput. Phys.*, Vol. 83, 1989.
- ¹¹Radespiel, R. and Swanson, R. C., "Progress with Multigrid Schemes for Hypersonic Flow Problems," *J. Comput. Phys.* Vol. 116, 1995 pp. 103–122.
- ¹²Pierce, N. A. and Giles, M. B., "Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes," *J. Comput. Phys.*, Vol. 136, 1997, pp. 425–445.
- ¹³Blazek J., "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.
- ¹⁴Turkel, E., Vatsa, V., and Venkatakrishnan, V., "Uni-Directional Implicit Acceleration Techniques for Compressible Navier-Stokes Solvers," *AIAA Paper* 99-3265, 1999.
- ¹⁵Mavriplis, D., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes", *J. Comput. Phys.*, Vol. 145, 1998, pp. 141–165.
- ¹⁶Mavriplis, D., "On Convergence Acceleration Techniques for Unstructured Meshes", *AIAA Paper* 98-2966, 1998.
- ¹⁷Rossow, C.-C., "Convergence Acceleration for Solving the Compressible Navier-Stokes Equations." *AIAA J.*, Vol. 44, 2006, pp. 345-352.
- ¹⁸Martinelli, L., "Calculations of Viscous Flow with a Multigrid Method," Ph. D. Thesis, Princeton University, 1987.
- ¹⁹Swanson, R. C. and Turkel, E., "Multistage Schemes with Multigrid for Euler and Navier-Stokes Equations," NASA TP 3631, 1997.
- ²⁰Swanson, R. C. and Turkel, E., "On central difference and upwind schemes," *J. Comput. Phys.*, Vol. 101, 1992, pp. 292–306.
- ²¹Roe, P. L., "Approximate Riemann solvers, parameter vectors and difference schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.
- ²²Rossow, C.-C., "A Flux Splitting Scheme for Compressible and Incompressible Flows," *J. Comput. Phys.*, Vol. 164, 2000, pp. 104–122.
- ²³A. 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 Dyna.*, Vol. 4, 1995, pp. 171–218.
- ²⁴Swanson, R. C., Radespiel, R., and Turkel, E., "On Some Numerical Dissipation Schemes," *J. Comput. Phys.*, Vol. 147, 1998, pp. 518-544.
- ²⁵Jameson, A., Schmidt, W. and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," *AIAA Paper* 81-1259, June 1981.
- ²⁶Roberts, T. W. and Swanson, R. C., "A Study of Multigrid Preconditioners Using Eigensystem Analysis," *AIAA Paper* 2005-5229, June 2005.
- ²⁷Abarbanel, S. and Gottlieb, D., "Optimal Time Splitting for Two- and Three-Dimensional Navier-Stokes Equations with Mixed Derivatives," *J. Comput. Phys.*, Vol. 41, 1981, pp. 1–33.
- ²⁸Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil RAE 2822 Pressure Distributions and Boundary Layer and Wake Measurements," AGARD-AR-138, 1979.
- ²⁹Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," AGARD-AR-138, Chap. B-1, May 1979.
- ³⁰Faßbender, J., "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.
- ³¹Van Leer, B., Tai, C.-H., Powell, K. G., "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations," *AIAA Paper* 89-1933, June 1989.
- ³²Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Flows," *AIAA Paper* 78-257, 1978.
- ³³Rossow, C.-C., "Toward Efficient Computation of Compressible and Incompressible Flows." *AIAA Paper* 2006-3522, June, 2006 (submitted to *J. Comput. Phys.*, 2006).

Cases	M_∞	α (deg.)	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

Table 1. Free-stream conditions 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

Table 2. Computed lift and drag coefficients for RAE 2822 airfoil. Numerical dissipation from Roe scheme. Weak grid clustering in neighborhood of shock wave.

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	481	1792	0.983
RK5/implicit	Roe	232	128	0.791
RK5/implicit (prec)	Roe	180	98	0.736
RK3/implicit	Roe	147	130	0.794
RK3/implicit (prec)	Roe	111	97	0.733
RK5/implicit	matrix	271	165	0.834
RK5/implicit (prec)	matrix	204	124	0.785
RK3/implicit	matrix	179	175	0.843
RK3/implicit (prec)	matrix	136	132	0.797
RK5/implicit	HCUSP	299	167	0.836

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

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	509	1891	0.984
RK5/implicit	Roe	257	142	0.810
RK5/implicit (prec)	Roe	201	110	0.761
RK3/implicit	Roe	161	141	0.809
RK3/implicit (prec)	Roe	126	110	0.761
RK5/implicit	matrix	286	175	0.843
RK5/implicit (prec)	matrix	210	128	0.791
RK3/implicit	matrix	182	178	0.845
RK3/implicit (prec)	matrix	144	140	0.807
RK5/implicit	HCUSP	312	174	0.842

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

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	680	2519	0.988
RK5/implicit	Roe	336	186	0.851
RK5/implicit (prec)	Roe	260	143	0.811
RK3/implicit	Roe	209	184	0.850
RK3/implicit (prec)	Roe	161	141	0.808
RK5/implicit	matrix	351	215	0.870
RK5/implicit (prec)	matrix	266	162	0.831
RK3/implicit	matrix	221	217	0.871
RK3/implicit (prec)	matrix	168	164	0.833
RK5/implicit	HCUSP	367	205	0.864

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

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	1105	2516	0.988
RK5/implicit (prec)	Roe	371	128	0.791
RK3/implicit (prec)	Roe	230	126	0.788
RK5/implicit (prec)	matrix	368	140	0.807
RK3/implicit (prec)	matrix	230	141	0.809

Table 6. Computational effort required for Case 1, 368×88 grid, $Re = 5.7 \times 10^6$, $AR = 3,949$.

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	1817	4123	0.993
RK5/implicit (prec)	Roe	511	176	0.844
RK3/implicit (prec)	Roe	317	174	0.842
RK5/implicit (prec)	matrix	562	213	0.869
RK3/implicit (prec)	matrix	354	216	0.871

Table 7. Computational effort required for Case 1, 368×88 grid, $Re = 20.0 \times 10^6$, $AR = 11,057$.

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	2739	6238	0.995
RK5/implicit (prec)	Roe	701	242	0.884
RK3/implicit (prec)	Roe	435	239	0.882
RK5/implicit (prec)	matrix	834	318	0.910
RK3/implicit (prec)	matrix	517	317	0.910

Table 8. Computational effort required for Case 1, 368×88 grid, $Re = 57.0 \times 10^6$, $AR = 30,714$.

Scheme	Dissipation	CPU Time (sec.)	MG Cycles	Reduction/Cycle
RK5/standard	matrix	3458	7865	0.996
RK5/implicit (prec)	Roe	841	291	0.902
RK3/implicit (prec)	Roe	521	286	0.901
RK5/implicit (prec)	matrix	1014	386	0.925
RK3/implicit (prec)	matrix	633	387	0.926

Table 9. Computational effort required for Case 1, 368×88 grid, $Re = 100.0 \times 10^6$, $AR = 50,260$.

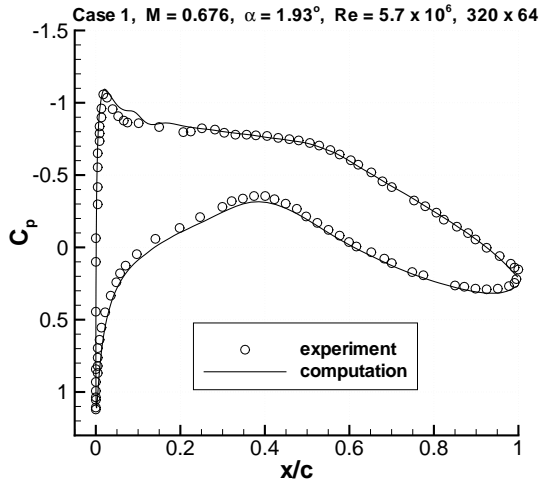


Figure 1. Surface pressure distribution for Case 1.

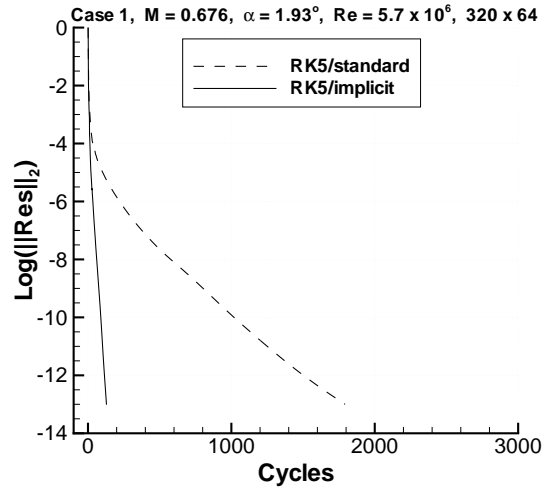


Figure 4. Convergence histories for Case 1.

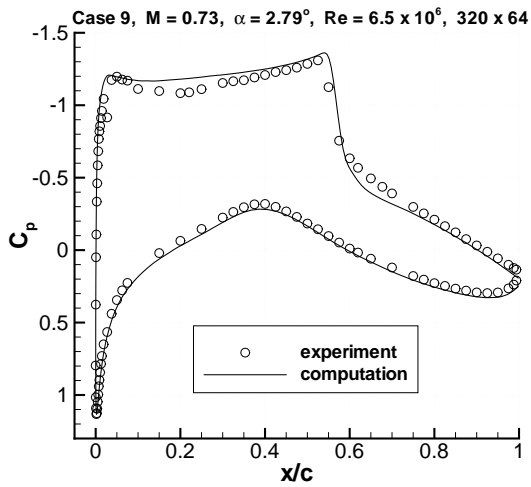


Figure 2. Surface pressure distribution for Case 9.

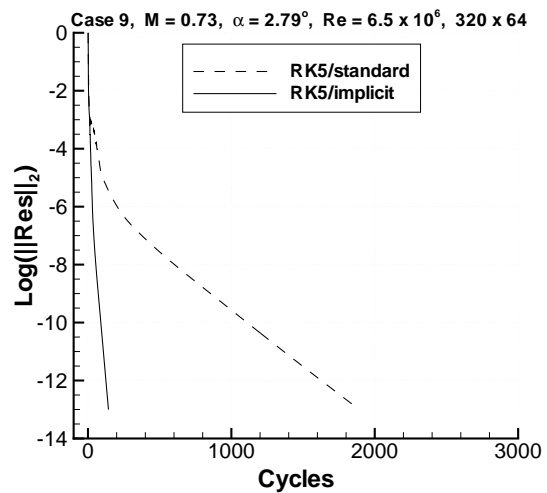


Figure 5. Convergence histories for Case 9.

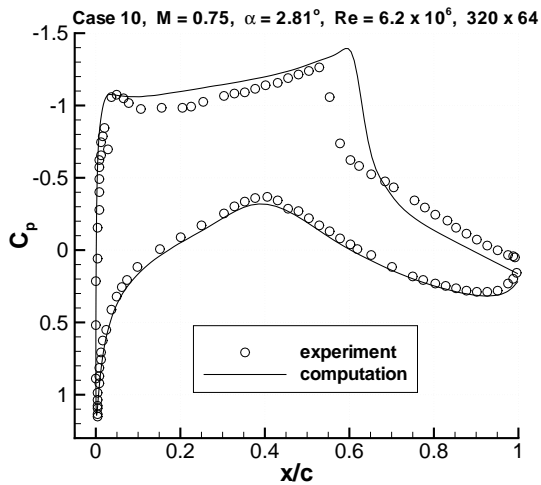


Figure 3. Surface pressure distribution for Case 10.

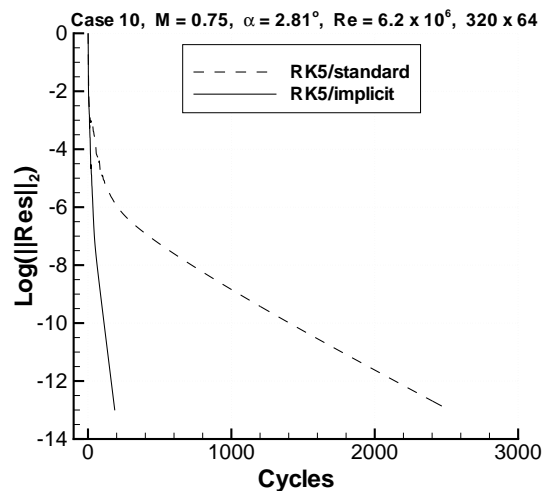


Figure 6. Convergence histories for Case 10.

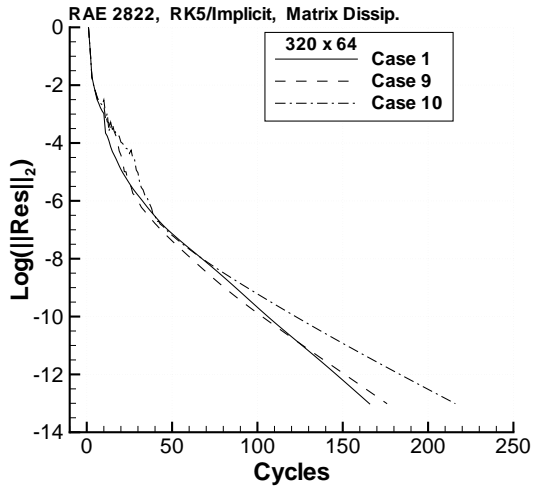


Figure 7. Convergence histories, matrix dissipation.

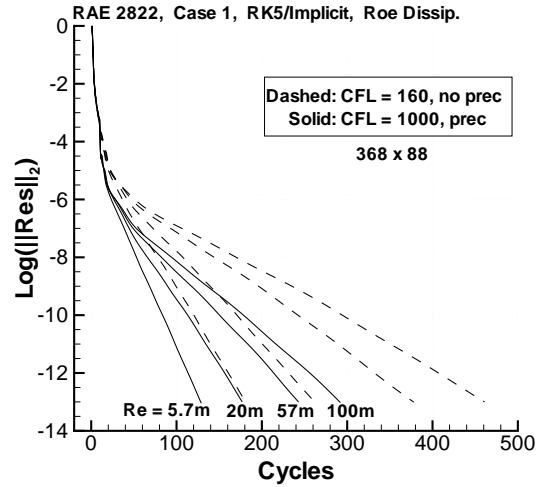


Figure 9. Convergence histories for Reynolds number variation (m means million), Roe dissipation.

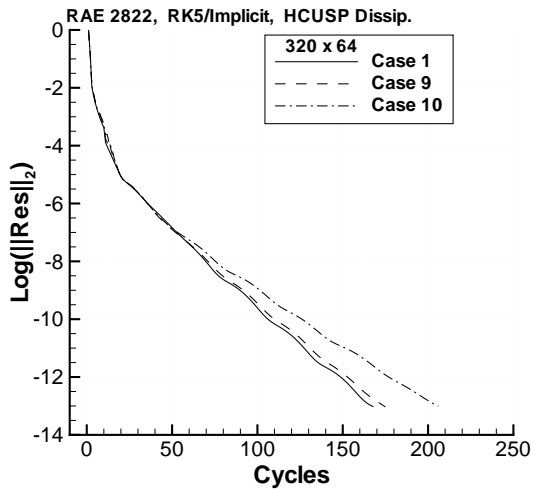


Figure 8. Convergence histories, HCUSP dissipation.

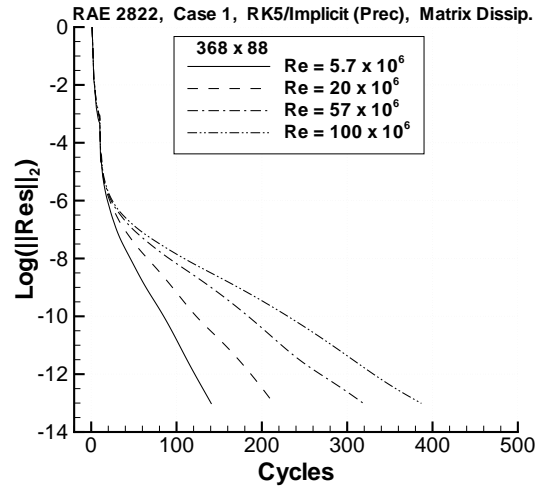


Figure 10. Convergence histories for Reynolds number variation, matrix dissipation.

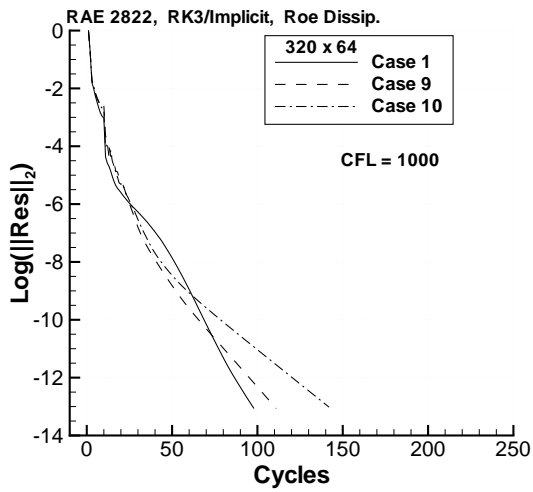


Figure 11. Convergence histories, Roe dissipation, 3 stages, CFL = 1000.

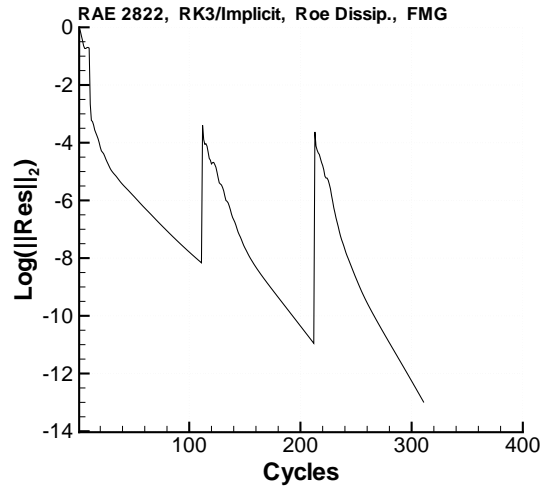


Figure 13. Convergence history for RK3/implicit scheme with FMG (Case 9, 320 x 64).

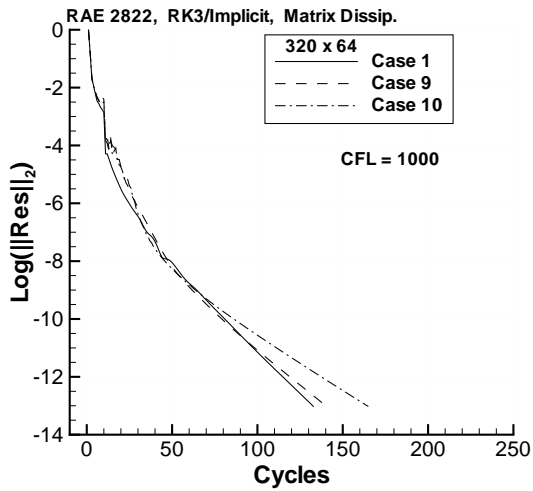


Figure 12. Convergence histories, matrix dissipation, 3 stages, CFL = 1000.

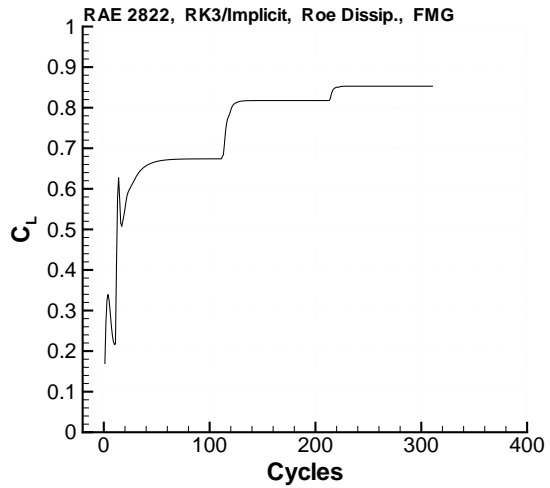


Figure 14. Lift history for RK3/implicit scheme with FMG (Case 9, 320 x 64).

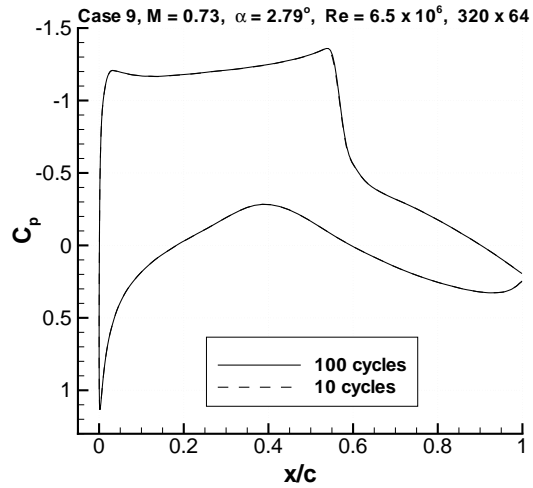


Figure 15. Surface pressure distribution at 10 and 100 cycles (Case 9, 320×64).

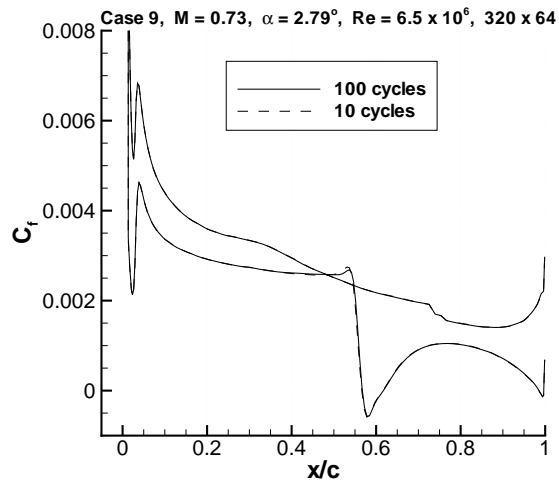


Figure 16. Surface skin-friction distribution at 10 and 100 cycles (Case 9, 320×64).

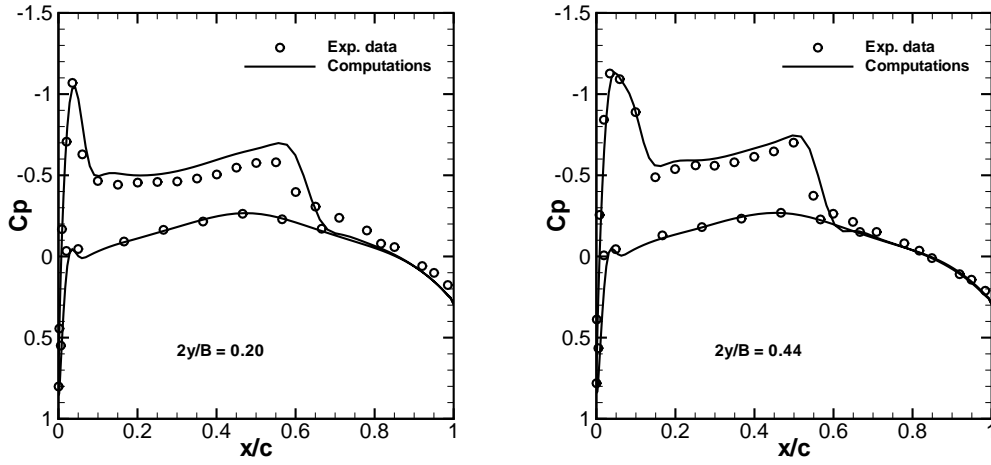


Figure 17. Surface pressure distribution for ONERA M6 wing, $2y/B = 0.20, 0.44$.

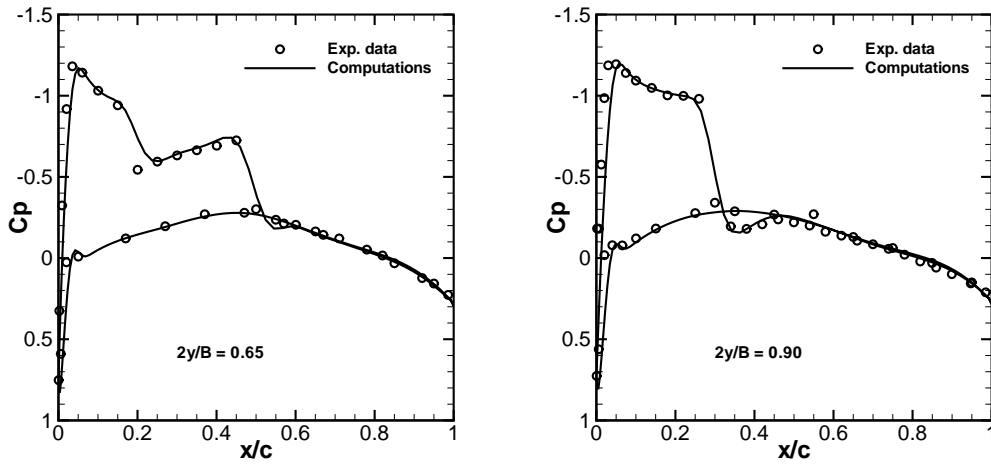


Figure 18. Surface pressure distribution for ONERA M6 wing, $2y/B = 0.65, 0.90$.

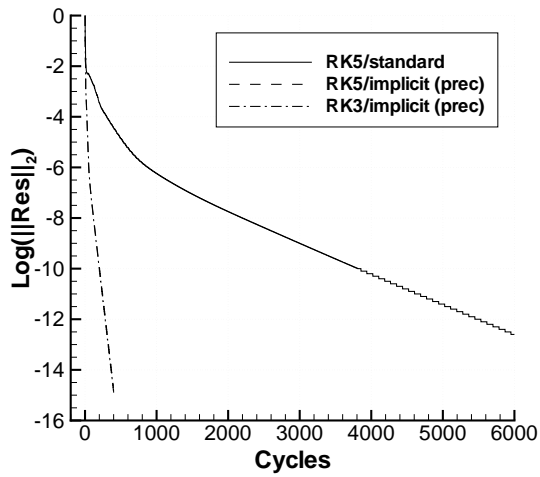


Figure 19. Residual histories for ONERA M6 wing, matrix dissipation.

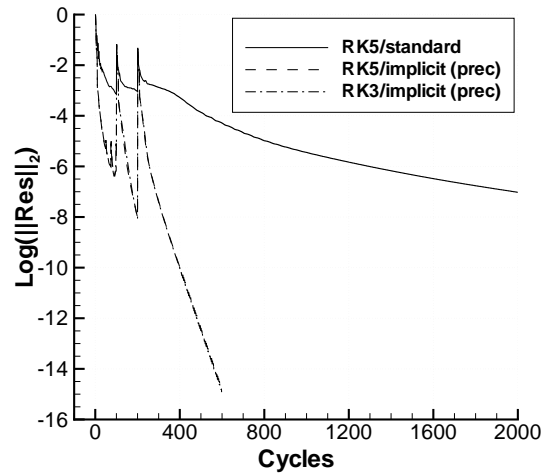


Figure 21. Residual histories with FMG for ONERA M6 wing, matrix dissipation.

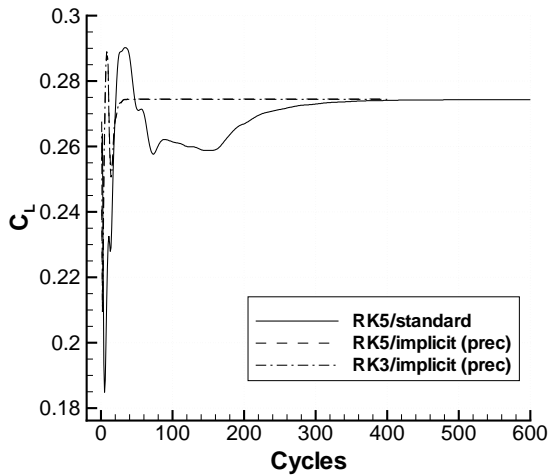


Figure 20. Lift coefficient histories for ONERA M6 wing.

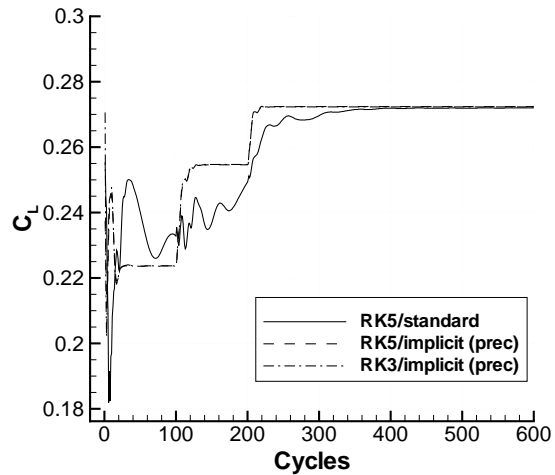


Figure 22. Lift coefficient histories with FMG for ONERA M6 wing.