

Comparison of Several Spatial Discretizations for the Navier–Stokes Equations

D. W. Zingg,* S. De Rango,* M. Nemeč,* and T. H. Pulliam†

**Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, Canada;* †*NASA Ames Research Center, Moffett Field, California*

Received October 22, 1999; revised February 28, 2000

Grid convergence studies for subsonic and transonic flows over airfoils are presented in order to compare the accuracy of several spatial discretizations for the compressible Navier–Stokes equations. The discretizations include the following schemes for the inviscid fluxes: (1) second-order-accurate centered differences with third-order matrix numerical dissipation, (2) the second-order convective upstream split pressure scheme (CUSP), (3) third-order upwind-biased differencing with Roe’s flux-difference splitting, and (4) fourth-order centered differences with third-order matrix numerical dissipation. The first three are combined with second-order differencing for the grid metrics and viscous terms. The fourth discretization uses fourth-order differencing for the grid metrics and viscous terms, as well as higher-order approximations near boundaries and for the numerical integration used to calculate forces and moments. The results indicate that the discretization using higher-order approximations for all terms is substantially more accurate than the others, producing less than two percent numerical error in lift and drag components on grids with less than 13,000 nodes for subsonic cases and less than 18,000 nodes for transonic cases. Since the cost per grid node of all of the discretizations studied is comparable, the higher-order discretization produces solutions of a given accuracy much more efficiently than the others. © 2000 Academic Press

Key Words: aerodynamics; Navier–Stokes equations; finite-difference methods; higher-order methods.

1. INTRODUCTION

It is essential that algorithms for the numerical solution of the compressible Navier–Stokes equations be both reliable and efficient, where efficiency is measured in terms of the computational effort required to achieve a given level of accuracy. Most algorithms in use for steady flows can be divided into two distinct components, a discretization of the spatial derivatives in the governing partial differential equations and an iterative method for

driving the discretized equations to steady state. For this class of algorithms, the accuracy of a converged steady solution depends only on the spatial discretization and is independent of the iterative method. The nature of the spatial discretization affects the efficiency of the overall algorithm through (1) its computational cost per node per iteration, (2) its accuracy, i.e., its effect on the number of grid nodes required to achieve a given level of numerical accuracy (independent of physical model error), and (3) its effect on the convergence rate of the iterative method and thus the total number of iterations needed to reach steady state.

The discretization of the inviscid flux terms has received considerable attention. It often consists of the following components:

1. A discrete approximation of the flux derivatives applicable to scalar flux functions, including a nondissipative skew-symmetric component and a dissipative symmetric component.
2. A splitting technique permitting extension of the dissipative component of the approximation to hyperbolic systems of equations. This includes both upwind schemes with inherent dissipation based on flux-vector or flux-difference splitting and central schemes with explicitly added dissipation.
3. A limiting technique which enforces monotonicity or positivity of specific elements of the solution through appropriate use of a first-order discretization in the vicinity of shock waves and other unresolved regions of high gradient.

The strategy chosen for adding numerical dissipation in the discretization of the inviscid fluxes can have a major impact on the accuracy of the overall spatial discretization. Flux-difference splitting [22] has become very popular, often in combination with a third-order upwind-biased operator and a flux limiter [15, 16]. Allmaras [1] used grid convergence studies to demonstrate the improvement in accuracy associated with the use of a flux-difference-split upwind scheme over the scalar artificial dissipation scheme of Jameson *et al.* [12] in the computation of laminar boundary layers. Similarly, Frew *et al.* [7] and others have shown that the matrix artificial dissipation scheme of Swanson and Turkel [23] is considerably more accurate than the scalar scheme in computing turbulent flows over airfoils. Swanson *et al.* [24] provide a comparison of several numerical dissipation schemes with a focus on the convective upstream split pressure (CUSP) [10, 11, 25] scheme. The results of Nemec and Zingg [18] show that the CUSP scheme is competitive with matrix dissipation in terms of accuracy at a reduced cost.

Modified wavenumber analysis shows that a second-order centered difference approximation to a second derivative is roughly twice as accurate as a second-order centered difference approximation to a first derivative. Hence it is reasonable to assume that a second-order treatment of the inviscid flux terms might be the most important source of error in a second-order spatial discretization. This motivates the common practice of combining a third-order upwind-biased approximation for the inviscid flux terms with a second-order approximation for the viscous fluxes. However, in computations of turbulent flows over airfoils, accurate prediction of drag requires low numerical errors in boundary layers, which are dominated by a balance between the viscous and inviscid fluxes in the streamwise momentum equation. Hence it may be worthwhile to raise the accuracy of the viscous terms as well.

The purpose of this paper is to compare the accuracy of four different spatial discretizations for a range of aerodynamic flows. The first uses matrix artificial dissipation

together with second-order centered differences. The second uses the CUSP scheme, again in combination with second-order centered differences. The third discretization is an upwind-biased scheme using flux-difference splitting based on the Roe average [22]. These discretizations use second-order approximations for the grid metrics and the viscous derivatives. The fourth discretization under study is the algorithm of De Rango and Zingg [6], which uses fourth-order centered differences with matrix artificial dissipation for the convective terms and fourth-order differencing for the grid metrics and viscous terms. Grid convergence studies are used to determine the numerical error produced by each discretization. Such studies provide a reliable means for assessing numerical error, as demonstrated by Zingg [28], and are an important component of the code verification process [21]. In the present context, the grid convergence studies provide a thorough and systematic comparison of three popular modern discretizations and a higher-order approach.

2. GOVERNING EQUATIONS AND NUMERICAL METHODS

The spatial discretizations under study are implemented in ARC2D [19], developed at the NASA Ames Research Center, which is the basis for the widely used code OVERFLOW [13]. ARC2D uses a generalized curvilinear coordinate system and the diagonal form [20] of the Beam–Warming approximate factorization algorithm [4] with local time stepping to solve the thin-layer Navier–Stokes equations. With the upwind-biased spatial discretization, convergence to steady state is achieved using the diagonally dominant approximate factorization algorithm in diagonal form of Klopfer *et al.* [14] with three sub-iterations per time step. The effects of turbulence are modelled using the Baldwin–Lomax model [2], and a far-field circulation correction is included at the outer boundary [19]. Laminar-turbulent transition points are specified, and the transition is ramped over two grid cells.

Governing equations. In two-dimensional generalized coordinates, the thin-layer Navier–Stokes equations are given by [19]

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \frac{\partial \hat{S}}{\partial \eta}, \quad (1)$$

where $\hat{Q} = J^{-1} Q = J^{-1}[\rho, \rho u, \rho v, e]^T$ is the vector of conservative dependent variables, \hat{E} and \hat{F} are the inviscid flux vectors, \hat{S} is the viscous flux vector, ξ and η are the streamwise and normal generalized coordinates, respectively, and J is the Jacobian of the coordinate transformation.

Matrix dissipation scheme. The matrix dissipation scheme [23] is implemented in the following manner,¹

$$\left(\frac{\partial \hat{E}}{\partial \xi} \right)_{j,k} \approx \delta_{\xi} \hat{E}_{j,k} - \nabla_{\xi} d_{j+1/2,k} \quad (2)$$

¹ An analogous term appears in the η direction.

with

$$\begin{aligned} d_{j+1/2,k} &= |\hat{A}|_{j+1/2,k} J_{j+1/2,k}^{-1} (\epsilon_{j+1/2,k}^{(2)} \Delta_\xi Q_{j,k} - \epsilon_{j+1/2,k}^{(4)} \nabla_\xi \Delta_\xi \nabla_\xi Q_{j,k}) \\ \epsilon_{j,k}^{(2)} &= \kappa_2 \max(\Upsilon_{j+1,k}, \Upsilon_{j,k}, \Upsilon_{j-1,k}) \\ \epsilon_{j,k}^{(4)} &= \max(0, \kappa_4 - \epsilon_{j,k}^{(2)}) \\ \Upsilon_{j,k} &= \frac{|p_{j+1,k} - 2p_{j,k} + p_{j-1,k}|}{|p_{j+1,k} + 2p_{j,k} + p_{j-1,k}|}, \end{aligned}$$

where δ_ξ is a centered difference operator, Δ_ξ and ∇_ξ are first-order forward and backward difference operators, and $\kappa_4 = 0.02$. We use $\kappa_2 = 0$ for subsonic flows and $\kappa_2 = 1.0$ for transonic flows. The term $\Upsilon_{j,k}$ is a pressure switch to control the use of first-order dissipation near shock waves. The matrix $|\hat{A}|$ is given by

$$|\hat{A}| = T_\xi |\Lambda_\xi| T_\xi^{-1}. \quad (3)$$

Here $|\Lambda_\xi|$ contains the eigenvalues of the flux Jacobian matrix $\hat{A} = \partial \hat{E} / \partial \hat{Q}$ as

$$|\Lambda_\xi| = \begin{bmatrix} |U| & 0 & 0 & 0 \\ 0 & |U| & 0 & 0 \\ 0 & 0 & |U + c\theta| & 0 \\ 0 & 0 & 0 & |U - c\theta| \end{bmatrix}, \quad (4)$$

where U is the contravariant velocity component in the ξ direction, c is the speed of sound, $\theta = \sqrt{\xi_x^2 + \xi_y^2}$, and ξ_x and ξ_y are metrics of the curvilinear coordinate transformation. The matrix T_ξ contains the right eigenvectors of \hat{A} . In evaluating $|\hat{A}|_{j+1/2,k}$ we have used the simple average; the Roe average is recommended for flows containing very strong shock waves. To avoid zero eigenvalues, the elements of $|\Lambda|_\xi$ are modified as

$$\begin{aligned} \tilde{\lambda}_1, \tilde{\lambda}_2 &= \max(\lambda_{1,2}, V_l \sigma) \\ \tilde{\lambda}_3 &= \max(\lambda_3, V_n \sigma) \\ \tilde{\lambda}_4 &= \max(\lambda_4, V_n \sigma), \end{aligned} \quad (5)$$

where σ is the spectral radius of the flux Jacobian. We use $V_l = V_n = 0$ for subsonic flows, and $V_l = 0.025$, $V_n = 0.25$ for transonic flows.

CUSP scheme. The CUSP scheme [10, 11] is formulated by a combination of differences of the state and flux vectors. We consider the ECUSP version only, and refer to it as the CUSP scheme for the remainder of the paper.

For the first-order CUSP scheme, the dissipative flux for the ξ coordinate direction is added as in Eq. (2) with

$$d_{j+1/2,k} = \frac{1}{2} J_{j+1/2,k}^{-1} \alpha_{j+1/2,k}^* c (Q_{j+1,k} - Q_{j,k}) + \frac{1}{2} J_{j+1/2,k}^{-1} \beta_{j+1/2,k} \hat{E}_{j+1/2,k}, \quad (6)$$

where

$$\Delta \hat{E}_{j+1/2,k} = \xi_{x_{j+1/2,k}} (E_{j+1,k} - E_{j,k}) + \xi_{y_{j+1/2,k}} (F_{j+1,k} - F_{j,k}).$$

The vectors E and F are the flux vectors written in Cartesian coordinates. The parameters α^* and β are given by

$$\alpha^* c = \alpha c - \beta \bar{U} \quad (7)$$

$$\beta = \begin{cases} \max(0, \frac{\bar{U} + \lambda^-}{\bar{U} - \lambda^-}) & \text{if } 0 \leq M \leq 1 \\ -\max(0, \frac{\bar{U} + \lambda^+}{\bar{U} - \lambda^+}) & \text{if } -1 \leq M < 0 \\ \text{sign}(M) & \text{if } |M| > 1, \end{cases} \quad (8)$$

where

$$\lambda^\pm = U \pm c \sqrt{\xi_x^2 + \xi_y^2}$$

and α is chosen to equal $|\bar{U}|/c$. The symbol \bar{U} denotes the arithmetic mean contravariant velocity, while the eigenvalues λ^\pm are determined at the Roe state.

To construct a higher-order CUSP scheme, limiters are added which activate near flow discontinuities. We use the limiter function of Jameson [10], as modified by Nemec and Zingg [18],

$$R(u, v) = 1 - \left| \frac{u - v}{|u| + |v| + \varepsilon/(|u| + |v| + 10^{-12})} \right|^2, \quad (9)$$

where all quantities are nondimensional, with $\varepsilon = 10^{-3}$. A higher-order CUSP scheme is obtained by defining the limited average,

$$L(u, v) = \frac{1}{2} R(u, v)(u + v) \quad (10)$$

and then constructing the appropriate left and right states for each variable (where the second subscript has been dropped for simplicity),

$$\begin{aligned} q_{j+1/2}^L &= q_j + \frac{1}{2} L(\Delta q_{j+3/2}, \Delta q_{j-1/2}) \\ q_{j+1/2}^R &= q_{j+1} - \frac{1}{2} L(\Delta q_{j+3/2}, \Delta q_{j-1/2}), \end{aligned} \quad (11)$$

where q indicates an element of the vector of conservative variables Q . The dissipative flux becomes

$$d_{j+1/2} = \frac{1}{2} J_{j+1/2}^{-1} \alpha_{j+1/2}^* (Q_{j+1/2}^R - Q_{j+1/2}^L) + \frac{1}{2} J_{j+1/2}^{-1} \beta_{j+1/2} \Delta \hat{E}(Q_{j+1/2}^R, Q_{j+1/2}^L), \quad (12)$$

where

$$\begin{aligned} &\Delta \hat{E}(Q_{j+1/2}^R, Q_{j+1/2}^L) \\ &= \xi_{x_{j+1/2}} (E(Q_{j+1/2}^R) - E(Q_{j+1/2}^L)) + \xi_{y_{j+1/2}} (F(Q_{j+1/2}^R) - F(Q_{j+1/2}^L)). \end{aligned}$$

Analogous terms appear in the η direction.

Near domain boundaries, it is necessary to modify the left and right states. At the first interior node the left state becomes

$$q_{j+1/2}^L = q_j + \frac{1}{2}L(\Delta q_{j+3/2}, \Delta q_{j+1/2}) \quad (13)$$

while the right state remains unmodified. At the last interior node, the left state does not require modification, but the right state becomes

$$q_{j+1/2}^R = q_{j+1} - \frac{1}{2}L(\Delta q_{j+1/2}, \Delta q_{j-1/2}). \quad (14)$$

Compared to the simple pressure switch, applying the limiter to each variable adds considerable computational expense to the algorithm. The limiter value has to be computed for each state variable at each node for each direction (i.e., eight evaluations of the limiter function per node) compared to just two evaluations per node of the pressure switch function. Further, the inversion of the left-hand side of the implicit algorithm becomes less efficient due to the fact that the limiter may apply different values to each conservation equation. To increase the efficiency of the algorithm we use the same limiter value in each equation, which is based on the pressure. This reduces the computational cost of the limiter to roughly the same level as that of the pressure switch. Nemeč and Zingg [18] have shown that this simplification adds very little error for the class of flows considered here.

Upwind-biased scheme. The third-order upwind-biased operator for the inviscid fluxes has the following form when applied to a scalar function,

$$\delta_x q_j = \frac{1}{6\Delta x} (q_{j-2} - 6q_{j-1} + 3q_j + 2q_{j+1}). \quad (15)$$

This can be written as the sum of a fourth-order skew-symmetric component and a third-order symmetric component as

$$\begin{aligned} \delta_x q_j = \frac{1}{12\Delta x} [& (-q_{j+2} + 8q_{j+1} - 8q_{j-1} + q_{j-2}) \\ & + (q_{j+2} - 4q_{j+1} + 6q_j - 4q_{j-1} + q_{j-2})]. \end{aligned} \quad (16)$$

Hence it is equivalent to a fourth-order centered scheme with a third-order dissipative component. This is implemented using Roe's flux difference splitting with an entropy fix. A limited extrapolation of the primitive variables is used to calculate the necessary left and right states. For example,

$$\begin{aligned} \rho_{j+1/2}^L &= \rho_j + \frac{1}{2}\Psi_j \left(\frac{1}{3}\nabla\rho_j + \frac{2}{3}\Delta\rho_j \right) \\ \rho_{j-1/2}^R &= \rho_j + \frac{1}{2}\Psi_j \left(\frac{2}{3}\nabla\rho_j + \frac{1}{3}\Delta\rho_j \right), \end{aligned} \quad (17)$$

where $\nabla\rho_j = \rho_j - \rho_{j-1}$, $\Delta\rho_j = \rho_{j+1} - \rho_j$, and Ψ is a limiter function. Note that second-order approximations of the grid metric terms are used in the present implementation of this scheme.

We use the limiter function of Koren [15, 16] with the modification proposed by Venkatakrishnan [27], which can be written for the ξ direction as

$$\begin{aligned} \tilde{\Psi}_j &= \frac{3\nabla\rho_j\Delta\rho_j + (K\theta)^3}{2(\Delta\rho_j - \nabla\rho_j)^2 + 3\nabla\rho_j\Delta\rho_j + (K\theta)^3} \\ \Psi_j &= \max(0, \tilde{\Psi}_j), \end{aligned} \tag{18}$$

where $\theta = \sqrt{\xi_x^2 + \xi_y^2}$, and we have used $K = 5$. This value of K was selected as a compromise between conflicting requirements of accuracy, convergence, and monotonicity. Larger values of K lead to oscillations near shocks, while smaller values result in excessive use of the first-order scheme, causing large numerical error and poor convergence. Similar results are obtained using the limiter function of Van Albada *et al.* [26].

Higher-order algorithm. The higher-order algorithm uses the following operators to approximate first derivatives [6]:

Interior (4th order),

$$\delta_x q_j = \frac{1}{12\Delta x}(-q_{j+2} + 8q_{j+1} - 8q_{j-1} + q_{j-2}); \tag{19}$$

First interior node (3rd order),

$$\delta_x q_j = \frac{1}{6\Delta x}(-2q_{j-1} - 3q_j + 6q_{j+1} - q_{j+2}); \tag{20}$$

Boundary (3rd order),

$$\delta_x q_j = \frac{1}{24\Delta x}(-11q_j + 18q_{j+1} - 9q_{j+2} + 2q_{j+3}). \tag{21}$$

The last equation is required only for the calculation of grid metrics. Elsewhere, the grid metrics are evaluated using the same operators as the inviscid fluxes without numerical dissipation.

The numerical dissipation is added as described in the earlier section on matrix dissipation. It requires the same five-point stencil as the fourth-order centered-difference operator and, in the absence of discontinuities, is third-order accurate. At near-boundary nodes, the following operator is used,

$$\frac{1}{\Delta x}(-q_{j-1} + 3q_j - 3q_{j+1} + q_{j+2}). \tag{22}$$

The viscous terms are in the general form

$$\partial_x(\alpha_j \partial_x \beta_j). \tag{23}$$

The following fourth-order expression is used to calculate the $\partial_x \beta_j$ term at half nodes,

$$(\delta_x \beta)_{j+1/2} = \frac{1}{24\Delta x}(\beta_{j-1} - 27\beta_j + 27\beta_{j+1} - \beta_{j+2}). \tag{24}$$

Near boundaries, the following third-order expression is used,

$$(\delta_x \beta)_{j+1/2} = \frac{1}{24\Delta x}(-23\beta_j + 21\beta_{j+1} + 3\beta_{j+2} - \beta_{j+3}). \tag{25}$$

The value of $\alpha_{j+1/2}$ in Eq. (23) is determined using the fourth-order interpolation formula

$$\alpha_{j+1/2} = \frac{1}{16}(-\alpha_{j-1} + 9\alpha_j + 9\alpha_{j+1} - \alpha_{j+2}). \quad (26)$$

Near boundaries, a third-order formula is used,

$$\alpha_{j+1/2} = \frac{1}{8}(3\alpha_j + 6\alpha_{j+1} - \alpha_{j+2}). \quad (27)$$

The complete operator is then

$$\begin{aligned} \delta_x(\alpha_j \delta_x \beta_j) = \frac{1}{24\Delta x} & [\alpha_{j-3/2}(\delta_x \beta)_{j-3/2} - 27\alpha_{j-1/2}(\delta_x \beta)_{j-1/2} \\ & + 27\alpha_{j+1/2}(\delta_x \beta)_{j+1/2} - \alpha_{j+3/2}(\delta_x \beta)_{j+3/2}] \end{aligned} \quad (28)$$

in the interior, and

$$\begin{aligned} \delta_x(\alpha_j \delta_x \beta_j) = \frac{1}{24\Delta x} & [-23\alpha_{j-1/2}(\delta_x \beta)_{j-1/2} + 21\alpha_{j+1/2}(\delta_x \beta)_{j+1/2} \\ & + 3\alpha_{j+3/2}(\delta_x \beta)_{j+3/2} - \alpha_{j+5/2}(\delta_x \beta)_{j+5/2}] \end{aligned} \quad (29)$$

near boundaries. The implementation of the Baldwin–Lomax turbulence model requires the calculation of the vorticity at half nodes. This is accomplished using the operators given in Eqs. (24) and (25).

The value of \hat{Q} at a far-field boundary node is calculated as [3]

$$\hat{Q}_{bc} = \frac{1}{2}(\hat{Q}_{\infty} + \hat{Q}_{ext}) - \frac{1}{2} \text{sign}(\hat{A})(\hat{Q}_{\infty} - \hat{Q}_{ext}) \quad (30)$$

$$\text{sign}(\hat{A}) = T_{\kappa} \text{sign}(\Lambda_{\kappa}) T_{\kappa}^{-1}, \quad (31)$$

where κ is chosen in the direction normal to the boundary, bc indicates the boundary value, ∞ indicates values obtained from free-stream conditions, and ext indicates values extrapolated from the interior nodes of the mesh. The eigenvalues and eigenvectors are calculated from the mean state, $\hat{Q}_{avg} = \frac{1}{2}(\hat{Q}_{\infty} + \hat{Q}_{ext})$. The following second-order extrapolation operator is used at the far-field boundary,

$$q_j = 3q_{j+1} - 3q_{j+2} + q_{j+3}. \quad (32)$$

The pressure at the airfoil surface is determined from a third-order approximation to $\partial p / \partial n = 0$, which gives

$$p_1 = \frac{1}{11}(18p_2 - 9p_3 + 2p_4). \quad (33)$$

For an adiabatic wall, $\partial T / \partial n = 0$, and, when coupled with the assumption of zero pressure gradient and the perfect gas law, this implies $\partial \rho / \partial n = 0$. Hence the density at the airfoil surface is determined from an expression analogous to Eq. (33). Note that the assumption that $\partial p / \partial n = 0$ is not strictly correct. However, for aerodynamic flows at high Reynolds numbers, the error introduced is very small. We have experimented with extrapolation of pressure with no significant change in the solution. Furthermore, the error introduced is non-ordered² [21], and hence does not affect the conclusions from grid convergence studies. The velocity is zero on the body surface.

² Meaning that the error introduced does not vanish as the grid spacing tends to zero. Such an error cannot be evaluated through grid convergence studies.

For grids with a “C” topology, which are used exclusively in this study, the interpolation at the wake-cut (wc) is computed to fourth-order using the data above and below the wake-cut as

$$q_{k_{wc}} = \frac{1}{6}(-q_{k_{wc}+2} + 4q_{k_{wc}+1} + 4q_{k_{wc}-1} - q_{k_{wc}-2}). \quad (34)$$

In order to preserve the accuracy provided by a higher-order solution, it is necessary to use a higher-order integration technique in calculating forces and moments. This is accomplished by fitting a cubic spline to the airfoil coordinates and the surface pressure and skin friction distributions. An adaptive quadrature routine based on the two-point Gauss–Legendre rule with a global error control strategy is then used to perform the integration [6].

Overall, other than the first-order numerical dissipation added near shock waves, these approximations are all consistent with third-order global accuracy in space. However, it is not clear that this can actually be achieved in the presence of grid and flow singularities.

3. TEST CASES AND GRIDS

We consider the following four test cases:

1. NACA 0012 airfoil, $M_\infty = 0.16$, $\alpha = 6^\circ$, $Re = 2.88 \times 10^6$, laminar-turbulent transition at 0.05 and 0.8 chords on the upper and lower surfaces, respectively.
2. NACA 0012 airfoil, $M_\infty = 0.16$, $\alpha = 12^\circ$, $Re = 2.88 \times 10^6$, laminar-turbulent transition at 0.01 and 0.95 chords on the upper and lower surfaces, respectively.
3. NACA 0012 airfoil, $M_\infty = 0.7$, $\alpha = 3^\circ$, $Re = 9.0 \times 10^6$, laminar-turbulent transition at 0.05 chords on both surfaces.
4. RAE 2822 airfoil, $M_\infty = 0.729$, $\alpha = 2.31^\circ$, $Re = 6.5 \times 10^6$, laminar-turbulent transition at 0.03 chords on both surfaces.

These cases span a range of typical aerodynamic flows. Case 1 is a fully attached subsonic flow, while Case 2 has a region of separated flow near the trailing edge on the upper surface. Cases 3 and 4 are transonic flows with shock waves of moderate strength on the upper surface. The transonic cases are from Holst [9]. Experimental data for Case 4 can be found in Cook *et al.* [5] The measured coordinates for the RAE 2822 airfoil are used, as in Maksymiuk *et al.* [17], rather than the standard coordinates. Experimental data for Cases 1 and 2 can be found in Gregory and O’Reilly [8].

Table I is a summary of the grids used. All of the grids have a “C” topology. The distance to the far-field boundary is 12 chords for all grids. While this is clearly a source of error³ [28] it is not a *discretization* error, and hence will not affect our conclusions.⁴ All grids were generated using an elliptic grid generator, such that the largest distance between nodes on grid A is one chord (near the far-field boundary). Grid B was generated by removing every second node in both coordinate directions from grid A, and grid C was similarly generated from grid B. This technique produces a sequence of grids suitable for a grid convergence study. We also show results for grid C2, which has additional grid nodes clustered near the upper-surface shock wave for Cases 3 and 4. Grids C and C2 have a practical node density, with under 14,000 nodes. Grids A and B are primarily for estimation of solution error.

³ The error incurred varies with the inverse of the distance to the outer boundary [21].

⁴ For accurate prediction of drag at high-lift conditions, a substantially larger distance to the far-field boundary is recommended [28].

TABLE I
Summary of Grids

Grid	Dimensions	Points on airfoil	Off-wall spacing ($\times 10^{-6}$)	Leading edge clustering ($\times 10^{-3}$)	Trailing edge clustering ($\times 10^{-3}$)
A	1057×193	801	0.23	0.1	0.5
B	529×97	401	0.53	0.2	1.0
C	265×49	201	1.2	0.4	2.0
C2	277×49	213	1.2	0.4	2.0

4. RESULTS AND DISCUSSION

The lift, pressure drag, and friction drag coefficients for the four test cases computed on grids A, B, and C are shown in Figs. 1–4. They are plotted versus $1/N$, where N is the number of grid nodes. The discretizations are labelled as follows:

1. *Matrix*, matrix artificial dissipation with second-order centered differences;
2. *CUSP*, CUSP scheme;
3. *Roe*, third-order upwind-biased scheme for the inviscid terms with second-order centered differences for the viscous terms;
4. *Higher-order*, matrix artificial dissipation with fourth-order centered differences for the inviscid and viscous terms.

It is important to note that the higher-order discretization uses the same numerical dissipation scheme as the second-order discretization labelled “Matrix.” For the second-order discretization, the truncation error from the second-order centered difference approximations to the inviscid and viscous flux terms is of lower order than the third-order numerical dissipation. For the higher-order scheme, the third-order numerical dissipation is the leading error term in the interior discretization. Finally, the first-order numerical dissipation is not used for the subsonic cases. With the matrix dissipation scheme, this is accomplished by setting $\kappa_2 = 0$; with the CUSP and Roe schemes, the limiters are turned off.

All four discretizations produce very similar results on grid A, as expected. Hence the grid A results provide a reference solution for estimating the numerical error on grid C. A reasonable goal is to achieve a numerical error in lift and drag components of less than two percent on grid C. One might argue that this is unnecessarily stringent, given that errors in drag associated with modelling of laminar-turbulent transition and turbulence can be significantly greater than two percent. However, we would like the numerical error to be substantially less than the physical model error, both to permit accurate assessment of physical model error and to avoid compounding these errors.

With a few exceptions, the numerical errors in the lift coefficient are below two percent on grid C. All of the methods appear to be roughly equally accurate. Closer inspection of the pressure distributions shows that this is not the case, as we shall see later. The errors in the drag components computed on grid C are much larger. In order to get a true picture of solution accuracy, it is important to examine the individual drag components, pressure, and friction drag, since the errors in these components are often of opposite sign. For the two subsonic cases, shown in Figs. 1 and 2, the errors in pressure, and friction drag produced by

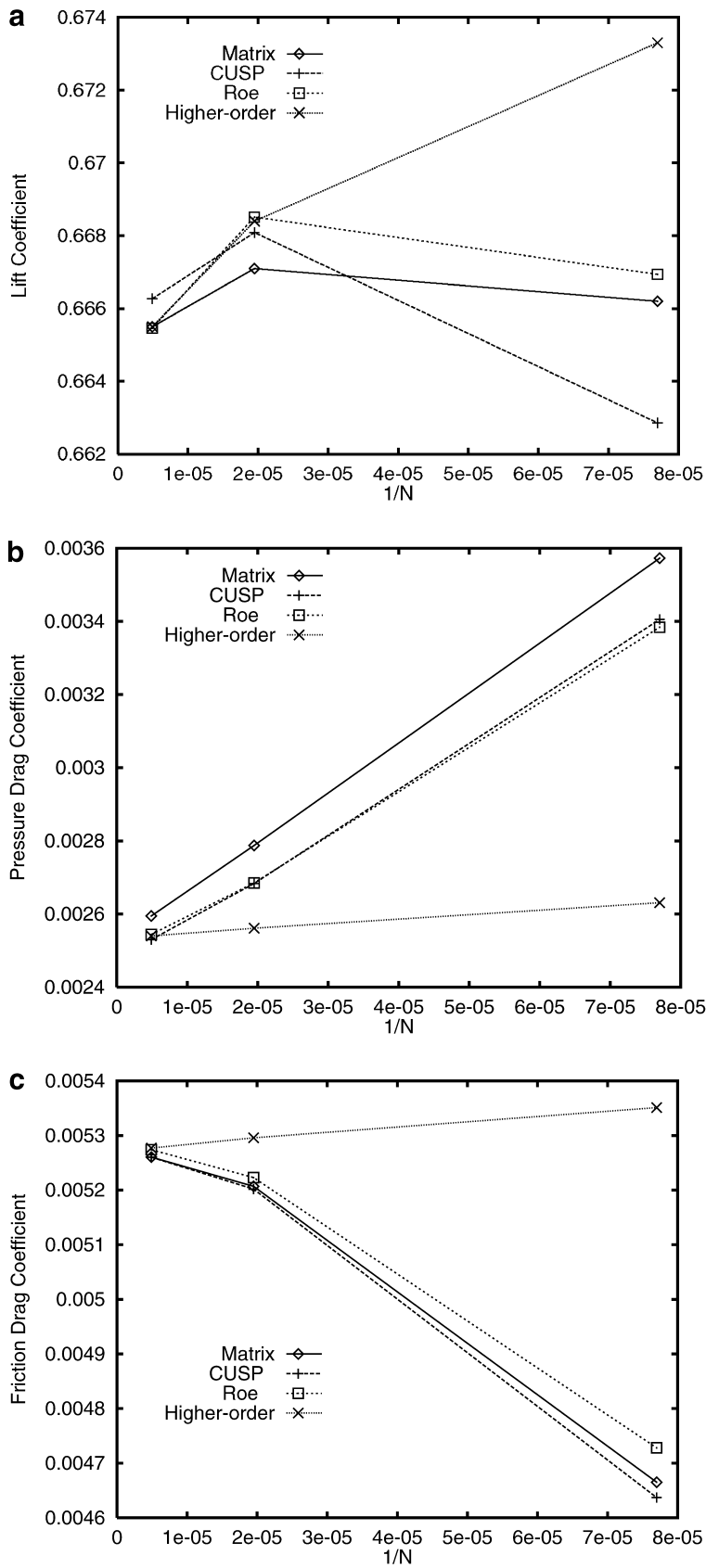


FIG. 1. Grid convergence for Case 1, (a) C_l . (b) C_{dp} . (c) C_{df} .

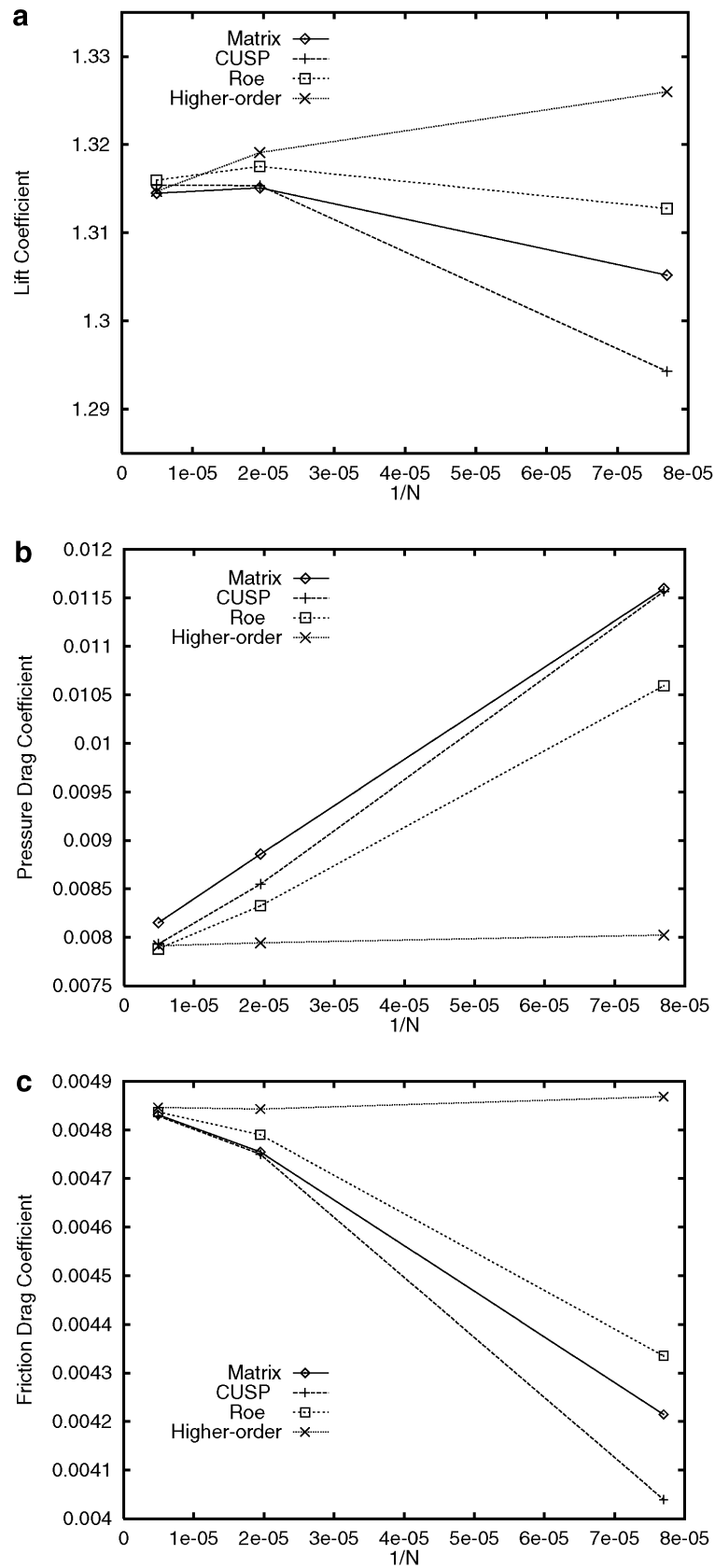


FIG. 2. Grid convergence for Case 2, (a) C_l . (b) C_{d_p} . (c) C_{d_f} .

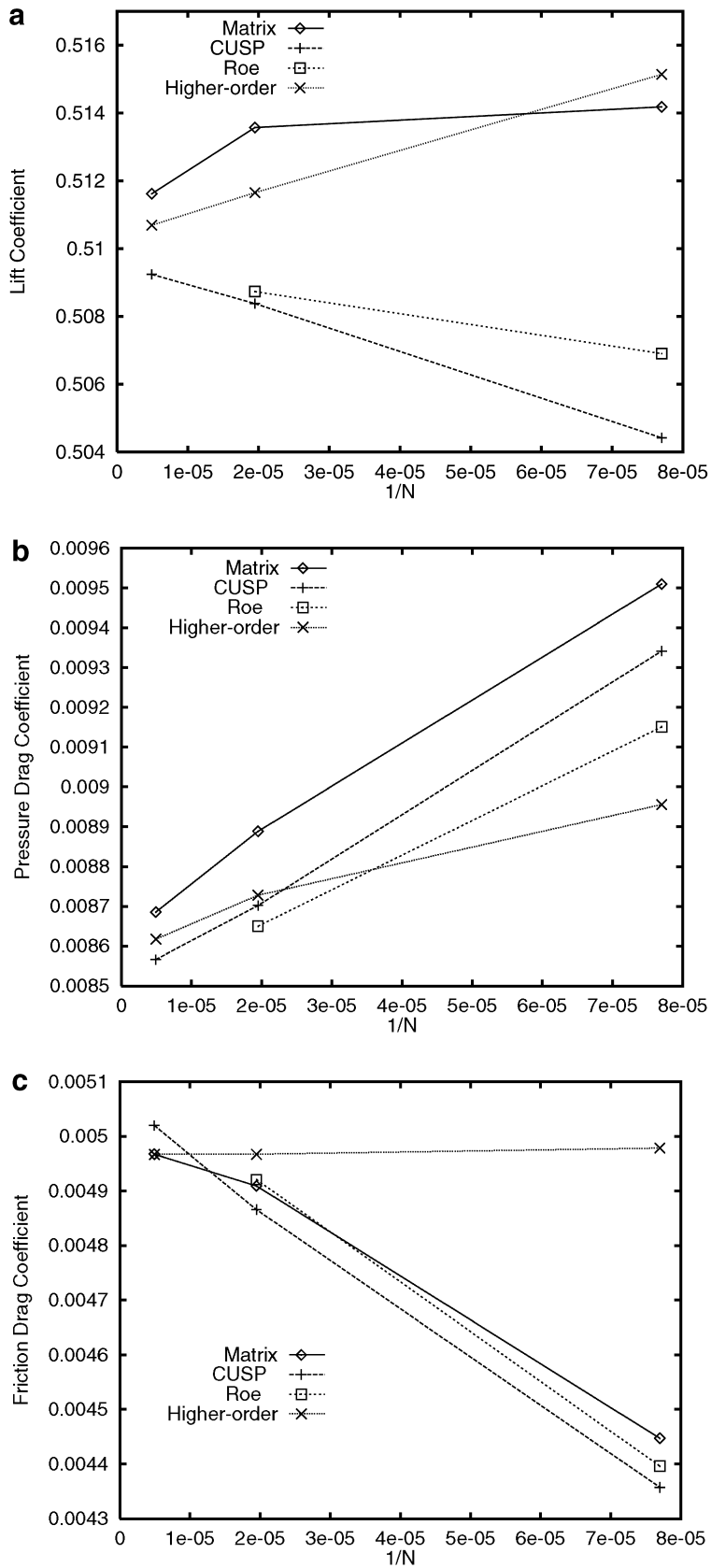


FIG. 3. Grid convergence for Case 3, (a) C_l . (b) C_{dp} . (c) C_{df} .

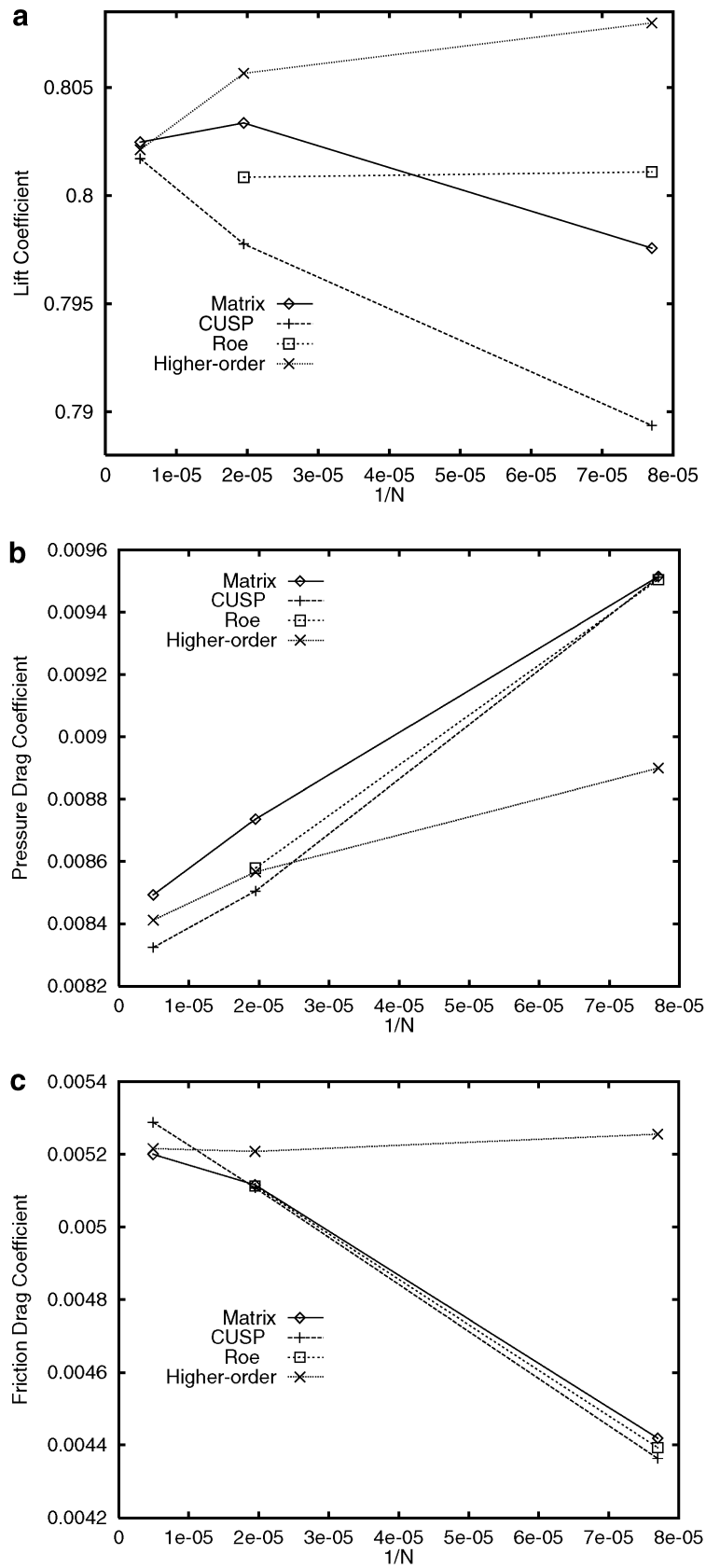


FIG. 4. Grid convergence for Case 4, (a) C_l . (b) C_{dp} . (c) C_{df} .

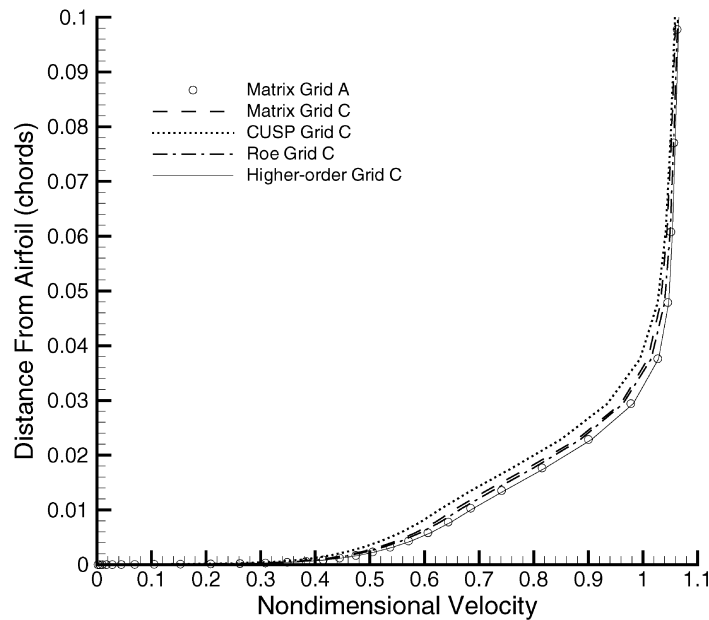


FIG. 5. Boundary layer velocity profiles on the upper surface at 85% chord, Case 2.

the higher-order scheme on grid C are consistently below two percent and are significantly smaller than those produced by the other three schemes. In fact, the errors produced by the higher-order scheme on grid C are generally smaller than those produced by the other schemes on grid B, which has four times as many nodes.

One cannot draw any general conclusions from these figures with respect to the asymptotic behavior of the error of the various discretizations. In principle, a second-order discretization will produce a linear variation of the force coefficients with $1/N$. Although some of the figures, such as the pressure drag for the subsonic cases, suggest second-order convergence rather strongly, others clearly do not. Furthermore, there is no concrete evidence that the higher-order algorithm leads to a higher-order asymptotic convergence rate in the presence of the trailing edge singularity. Hence our emphasis is on the actual error obtained on grid C, as determined by comparison with grid A. With this criterion, the benefits of the higher-order discretization are substantial.

Figure 5 shows the boundary-layer velocity profiles at 85% chord on the upper surface for Case 2 computed on grid C. The grid A results,⁵ which provide an accurate reference solution, show every fourth grid node. The higher-order scheme is superior to the other schemes, with very little error even on grid C. It achieves a low numerical error despite using the same numerical dissipation scheme as is used by the matrix scheme, demonstrating that the third-order numerical dissipation is not a major source of error on grid C. Hence the numerical dissipation scheme is achieving its goal of producing stability and damping under-resolved modes without introducing significant error.

For the transonic cases, shown in Figs. 3 and 4, the higher-order discretization produces the smallest pressure drag error on grid C, but it is nevertheless well in excess of two percent. One source of this error is the first-order dissipation introduced near the shock wave. All of the schemes produce lower pressure drag errors when run without any limiting, i.e., without any first-order dissipation, but visible oscillations result at the shock, and algorithm robustness can be compromised. Another possible source of error with the matrix dissipation

⁵ On grid A, the various discretizations produce solutions which are indistinguishable on the scale shown.

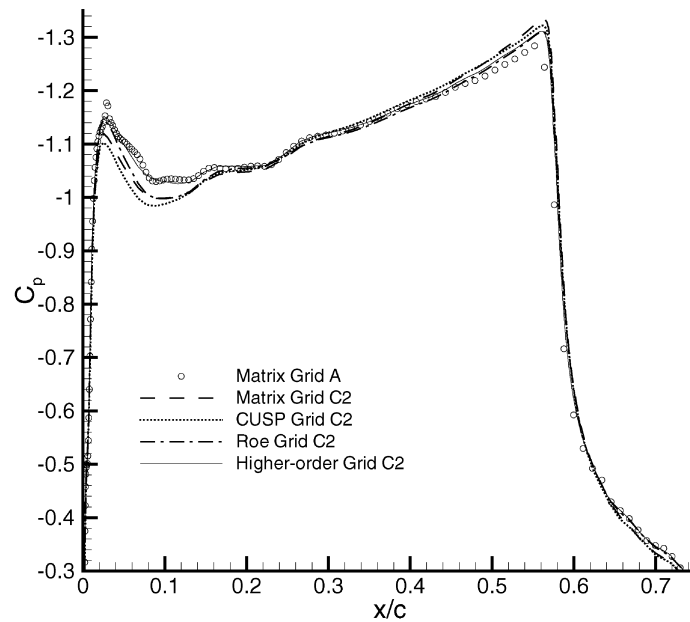


FIG. 6. Pressure coefficient for Case 4.

scheme (thus affecting the higher-order algorithm) is the requirement of nonzero values of V_l and V_n in Eq. (4) for transonic flows. However, given that the error in friction drag obtained using the higher-order discretization is very small, consistent with the subsonic cases, it is unlikely that the nonzero values of V_l and V_n are contributing significantly to the error in pressure drag. Adding nodes near the shock, as in grid C2, does not reduce the pressure drag error significantly. We will see later that adding grid nodes in the normal direction, possibly in order to better resolve the small separation bubbles seen at the shocks, is more critical for these cases. Figure 6 shows details of the pressure coefficient on the upper surface of the airfoil for Case 4 computed on grid C2. For the grid A solution, every second grid point is plotted. The higher-order discretization produces a significant reduction in error over the first 20% chord.

Figure 7 shows the values of the Koren limiter, Eq. (17), and the Nemec–Zingg modification of the Jameson limiter, Eq. (9), for Case 4, grid C. In both cases, the limiter is primarily active only near the upper-surface shock, as desired, although the Koren limiter is triggered slightly near the leading and trailing edges. In general, it is a worthwhile exercise to generate such plots in order to verify that appropriate parameter values have been selected. For example, as the value of K is reduced in Eq. (17), the limiter begins to trigger in many regions far from the shock wave, and hence the solution becomes polluted by excessive first-order dissipation. A theoretical basis for determining K would be of significant benefit.

For the subsonic cases, the higher-order algorithm generally produces a numerical error of less than two percent in the lift, pressure drag, and friction drag coefficients on grid C, which has 12,985 nodes. However, for the transonic cases, the error in the pressure drag exceeds two percent on grid C. We now revisit the transonic cases using an additional family of grids (designated A3, B3, and C3), generated in the same manner as the previous grids, whose properties are summarized in Table II. These grids are designed specifically for the transonic cases, with increased node density in the normal direction and more nodes on the upper surface than the lower surface (but no clustering at the shock). Grid C3 has 17,485 nodes, which is 35% more than grid C, but far fewer than grid B. Furthermore, for all of the discretizations, we add the first-order numerical dissipation at the shock in the streamwise

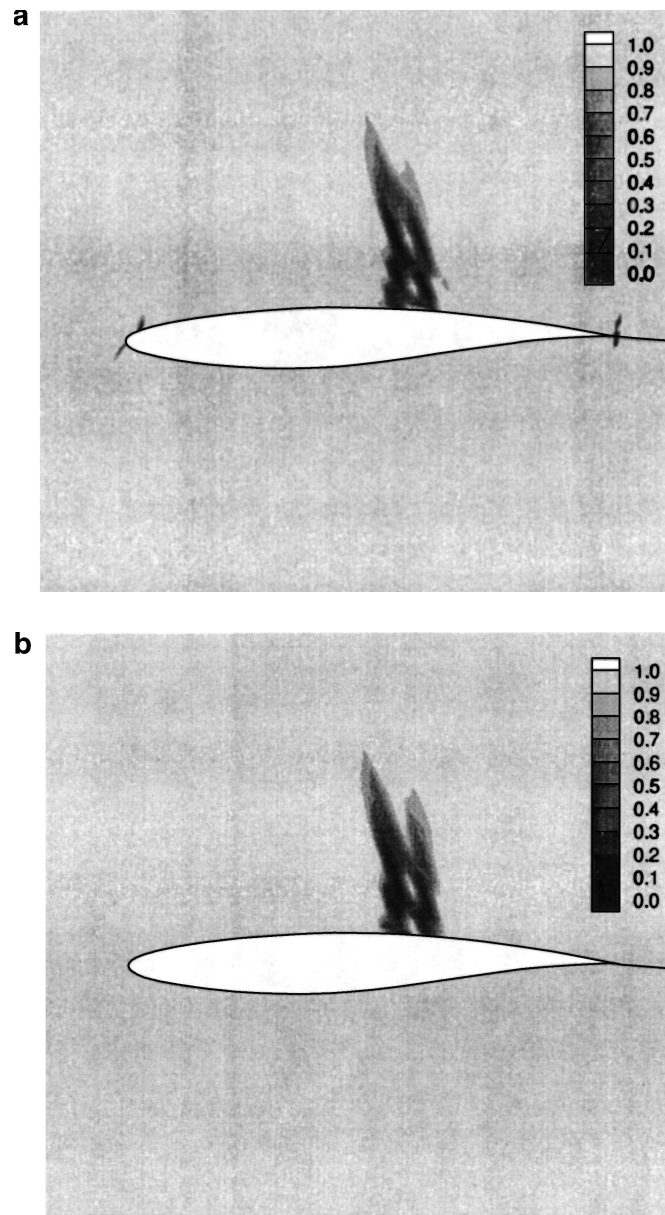


FIG. 7. (a) Limiter values for Case 4, Koren limiter. (b) Limiter values for Case 4, Nemec–Zingg modification of Jameson limiter.

direction only, which also reduces the errors. The results achieved on grid A3 are very close to those on grid A, providing further evidence that the numerical error is very small on such dense grids. Tables III and IV show the lift, pressure drag, and friction drag coefficients computed using grid C3 for Cases 3 and 4, respectively. In addition, the tables show the

TABLE II
Additional Grids for Transonic Cases

Grid	Dimensions	Points on airfoil	Off-wall spacing ($\times 10^{-6}$)	Leading edge clustering ($\times 10^{-3}$)	Trailing edge clustering ($\times 10^{-3}$)
A3	1073×257	849	0.23	0.1	0.25
B3	537×129	425	0.53	0.2	0.5
C3	269×65	213	1.2	0.4	1.0

TABLE III
Errors for Case 3, Grid C3

Algorithm	C_l	$\%C_l$	C_{dp}	$\%C_{dp}$	C_{df}	$\%C_{df}$
Matrix	0.5135	0.7	0.009058	5.2	0.004776	4.0
CUSP	0.5053	0.9	0.008885	3.2	0.004714	5.2
Roe	0.5078	0.4	0.008772	1.9	0.004723	5.0
Higher-order	0.5122	0.4	0.008719	1.3	0.004970	0.06

Note. The columns labelled % show the magnitude of the percent error relative to the values computed using the higher-order algorithm on grid A3, which are $C_l = 0.5100$, $C_{dp} = 0.008609$, $C_{df} = 0.004973$.

magnitude of the percent error relative to the higher-order results computed on grid A3, which are given in the figure captions. The higher-order algorithm produces much less than 2% error in all quantities, while the other algorithms produce errors between 3 and 5% for the drag components.

In order to examine the relative importance of the higher-order treatment of the viscous fluxes, we have run the higher-order algorithm with lower-order viscous terms. Although the results vary from case to case, the higher-order viscous terms generally account for roughly 10% of the error reduction associated with the higher-order discretization relative to the second-order scheme with matrix dissipation. For the friction drag in the transonic cases, the contribution from the higher-order viscous terms is over 25%. Using case 2 as an example, the second-order matrix algorithm produces an error in pressure drag on grid C of roughly 47% in comparison with the grid A solution. Using the higher-order algorithm, this error is reduced to 1.3%. If lower-order approximations are used for the viscous terms, the error increases to 4.6%. Although the higher-order viscous terms account for a relatively small fraction of the overall error reduction, they reduce the error by a factor greater than three in this example.

In the higher-order algorithm, the treatment of the inviscid terms, including the grid metrics, accounts for the largest portion of the error reduction. The present implementation of the third-order upwind scheme does not produce comparable benefits, largely because second-order approximations are used for the grid metrics. Our results indicate that fourth-order metric approximations should be used with the third-order scheme, consistent with the fact that the skew-symmetric portion of the third-order upwind-biased operator is a fourth-order centered difference operator. The higher-order grid metrics and first-derivative

TABLE IV
Errors for Case 4, Grid C3

Algorithm	C_l	$\%C_l$	C_{dp}	$\%C_{dp}$	C_{df}	$\%C_{df}$
Matrix	0.7984	0.4	0.008863	5.2	0.005019	3.8
CUSP	0.7886	1.6	0.008718	3.5	0.004978	4.6
Roe	0.7979	0.4	0.008754	3.9	0.004988	4.4
Higher-order	0.8023	0.1	0.008517	1.1	0.005223	0.1

Note. The columns labelled % show the magnitude of the percent error relative to the values computed using the higher-order algorithm on grid A3, which are $C_l = 0.8013$, $C_{dp} = 0.008425$, $C_{df} = 0.005217$.

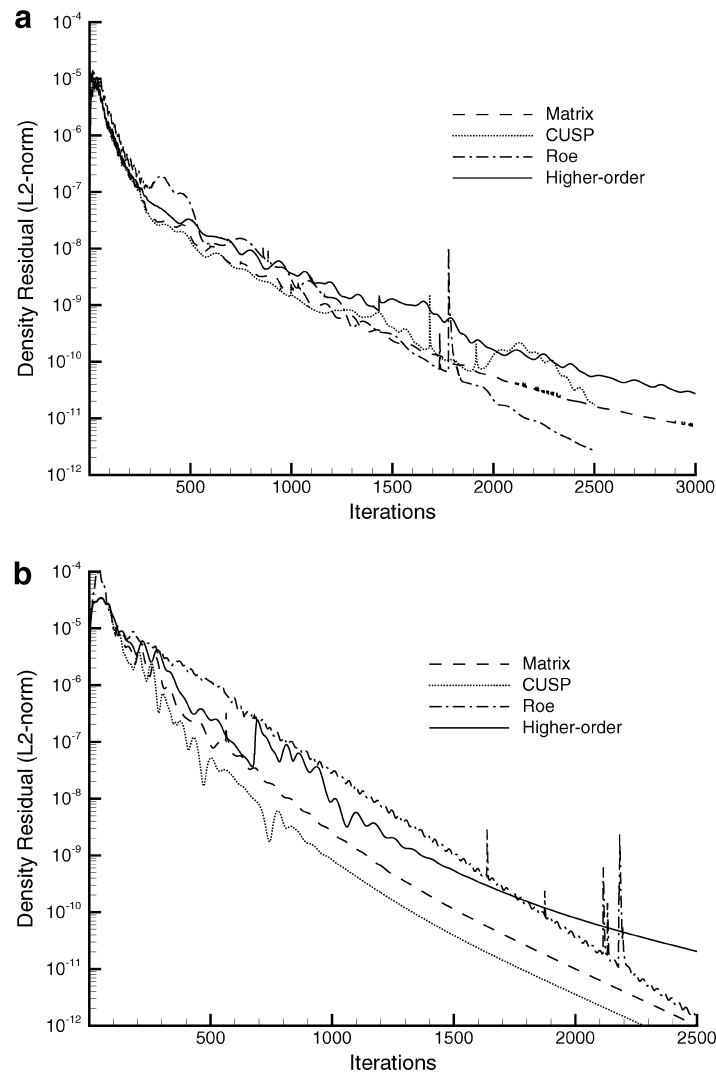


FIG. 8. (a) Residual convergence history for Case 1, grid C. (b) Residual convergence history for Case 3, grid C.

approximation used in calculating the velocity derivative at the surface also account for much of the reduction in friction drag error. The second-order three-point one-sided difference operator typically used is particularly susceptible to error from grid stretching.

Finally, we consider the cost of the various discretizations studied. In terms of the number of iterations required to achieve a steady state on grid C, the discretizations perform similarly, with some variation from case to case, as shown in Fig. 8. All cases are converged to within plotting accuracy for all results shown. In cases where the residual does not reach machine zero, the cause is either the turbulence model⁶ or the limiter.⁷ Table V shows the relative cost of a single evaluation of the right-hand side for each discretization, excluding the boundary-condition calculation. The first column shows the relative cost for a subsonic case, while the second column shows a transonic case, with the results for the matrix scheme normalized to unity in each case. Although these relative costs should not be considered definitive (with more effort, each algorithm could be programmed more efficiently), they provide a rough indication which is consistent with expectations. The higher-order algorithm costs

⁶ If the turbulence model is frozen, full convergence is obtained.

⁷ Freezing the limiter produces convergence to machine zero.

TABLE V
Relative Cost of a Right-Hand-Side Evaluation

Algorithm	Subsonic	Transonic
Matrix	1	1
CUSP	0.80	0.83
Roe	1.13	1.32
Higher-order	1.06	1.06

only 6% more than the lower-order algorithm with matrix dissipation. The high cost of the Roe scheme for the transonic case is associated with the use of a separate limiter for each variable, which can be avoided for the cases considered here. The differences in cost per iteration are somewhat smaller than those for a right-hand-side evaluation, since the setup and solution of the left-hand-side is roughly equivalent for each algorithm. Overall, these cost differences per right-hand-side evaluation are quite small, and since the higher-order algorithm produces equivalent accuracy on a grid with several times fewer nodes, its overall computational expense needed to achieve a given level of accuracy is by far the least of the schemes considered.

5. CONCLUSIONS

Detailed grid convergence studies have been performed in order to compare the accuracy of several modern spatial discretizations, specifically in the context of subsonic and transonic turbulent flows over airfoils. The following conclusions can be drawn from the results:

1. The present implementation of the third-order upwind-biased scheme for the inviscid fluxes produces little improvement over the second-order schemes. The results indicate that fourth-order approximations of grid metric terms should be used with the third-order upwind-biased scheme.

2. Matrix dissipation, the CUSP scheme, and flux-difference splitting all lead to comparable accuracy. When coupled with second-order schemes, the numerical dissipation introduced using these approaches is not a major source of error. This is clearly shown by the accuracy of the higher-order algorithm, which includes the same matrix numerical dissipation as the second-order matrix algorithm studied.

3. The accuracy associated with the popular approach to flux limiting exemplified by the Koren limiter is highly dependent on a parameter introduced to prevent excessive triggering of the limiter. Depending on the choice of this parameter, too much or too little limiting can be obtained. We have suggested a value for this parameter which is optimal for the two transonic cases considered, but it is not known whether this value is optimal for other flow cases and grids. Similar comments apply to the approach used in the Nemec–Zingg modification of the Jameson limiter, but it is less sensitive to the value of the parameter.

4. Using the higher-order discretization, less than 2% numerical error in lift and drag components can be obtained on grids with less than 13,000 nodes for subsonic cases and less than 18,000 nodes for transonic cases. Since the cost of the higher-order discretization per grid node is comparable to that of the other discretizations, it is capable of producing accurate solutions with a substantial reduction in computing expense.

Once the numerical error is reduced to below 2% on practical grids, one must give serious consideration to other sources of error, including the thin-layer approximation and the treatment of laminar-turbulent transition. Finally, it would be interesting to see the relative accuracy of discretizations for unstructured grids assessed using grid convergence studies similar to those presented here.

REFERENCES

1. S. R. Allmaras, Contamination of laminar boundary layers by artificial dissipation in Navier–Stokes solutions, in *Numerical Methods for Fluid Dynamics*, edited by M. J. Baines and K. W. Morton (Clarendon, Oxford, UK, 1993).
2. B. S. Baldwin and H. Lomax, *Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows*, AIAA Paper No. 78-257, 1978.
3. T. J. Barth, *Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier–Stokes Equations*, Lecture Notes, Von Karman Institute for Fluid Dynamics Lecture Series, 1994-05, Brussels, 1994.
4. R. Beam and R. F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation law form, *J. Comput. Phys.* **22**, 87 (1976).
5. P. H. Cook, M. A. MacDonald, and M. C. P. Firmin, *Aerofoil RAE 2822—Pressure Distributions, and Boundary-Layer and Wake Measurements*, AGARD-AR-138, May 1979.
6. S. De Rango and D. W. Zingg, *Aerodynamic Computations Using a Higher-Order Algorithm*, AIAA Paper 99-0167, January 1999.
7. K. Frew, D. W. Zingg, and S. De Rango, Artificial dissipation schemes for viscous airfoil computations, *AIAA J.* **36**(9), 1732 (1998).
8. N. Gregory and C. L. O'Reilly, *Low-Speed Aerodynamic Characteristics of NACA 0012 Airfoil Section, Including the Effects of Upper-Surface Roughness Simulating Hoar Frost*, Aeronautical Research Council, Reports and Memoranda No. 3726, UK, January 1970.
9. T. L. Holst, *Viscous Transonic Airfoil Workshop Compendium of Results*, AIAA Paper 87-1460, June 1987.
10. 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 Dynam.* **4**, 171 (1995).
11. A. Jameson, Analysis and design of numerical schemes for gas dynamics. II. Artificial diffusion and discrete shock structure, *Int. J. Comput. Fluid Dynam.* **5**, 1 (1995).
12. A. Jameson, W. Schmidt, and E. Turkel, *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time Stepping*, AIAA Paper 81-1259, June 1981.
13. D. Jespersen, T. Pulliam, and P. Buning, *Recent Enhancements to OVERFLOW*, AIAA Paper 97-0644, January 1997.
14. G. H. Klopfer, R. F. Van der Wijngaart, and C. M. Hung, *A Diagonalized Diagonal Dominant Alternating Direction Implicit (D3ADI) Scheme and Subiteration Correction*, AIAA Paper No. 98-2824, 1998.
15. B. Koren, Upwind schemes, multigrid and defect correction for the steady Navier–Stokes equations, in *Proc. 2nd Int. Conf. on Nonlinear Hyperbolic Problems*, edited by J. Ballman and R. Jeltsch (Vieweg, Braunschweig, 1988).
16. B. Koren, Upwind schemes for the Navier–Stokes equations, in *Proc. 11th Int. Conf. on Num. Meth. in Fluid Dynamics*, edited by D. L. Dwoyer, M. Y. Hussaini, and R. G. Voigt (Springer-Verlag, Berlin, 1989).
17. C. M. Maksymiuk, R. C. Swanson, and T. H. Pulliam, *A Comparison of Two Central Difference Schemes for Solving the Navier–Stokes Equations*, NASA TM-102815, July 1990.
18. M. Nemeć and D. W. Zingg, Aerodynamic computations using the convective upstream split pressure scheme with local preconditioning, *AIAA J.* **38**(3), (2000).
19. T. H. Pulliam, *Efficient Solution Methods for the Navier–Stokes Equations*, Lecture Notes, Von Karman Institute for Fluid Dynamics Lecture Series, Brussels, 1986.
20. T. H. Pulliam and D. S. Chaussee, A diagonal form of an implicit approximate-factorization algorithm, *J. Comput. Phys.* **39**, 347 (1981).

21. P. J. Roache, *Verification and Validation in Computational Science and Engineering* (Hermosa, New Mexico, 1998).
22. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
23. R. C. Swanson and E. Turkel, On central-difference and upwind schemes, *J. Comput. Phys.* **101**, 292 (1992).
24. R. C. Swanson, R. Radespiel, and E. Turkel, On some numerical dissipation schemes, *J. Comput. Phys.* **147**, 518 (1998).
25. S. Tatsumi, L. Martinelli, and A. Jameson, *A New High Resolution Scheme for Compressible Viscous Flow with Shocks*, AIAA Paper 95-0466, January 1995.
26. G. D. Van Albada, B. Van Leer, and W. W. Roberts, A comparative study of computational methods in cosmic gas dynamics, *Astron. Astrophys.* **108**, 76 (1982).
27. V. Venkatakrisnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, *J. Comput. Phys.* **118**, 120 (1995).
28. D. W. Zingg, Grid studies for thin-layer Navier–Stokes computations of airfoil flowfields, *AIAA J.* **30**(10), 2561 (1992).