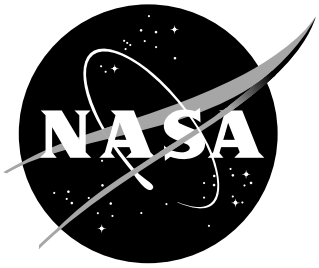
NASA/TM–2016–219003



Comparison of NACA 0012 Laminar Flow Solutions: Structured and Unstructured Grid Methods

R. C. Swanson
NASA Langley Research Center, Hampton, Virginia

S. Langer
DLR, Deutsches Zentrum für Luft- und Raumfahrt
Institut für Aerodynamik und Strömungstechnik,
Braunschweig, Germany

January 2016

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

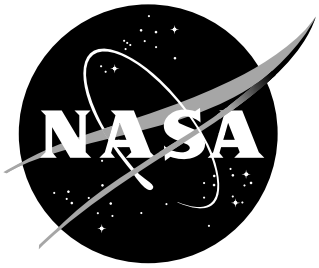
- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621–0134
- Phone the NASA STI Help Desk at (301) 621–0390
- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–2016–219003



Comparison of NACA 0012 Laminar Flow Solutions: Structured and Unstructured Grid Methods

R. C. Swanson
NASA Langley Research Center, Hampton, Virginia

S. Langer
DLR, Deutsches Zentrum für Luft- und Raumfahrt
Institut für Aerodynamik und Strömungstechnik,
Braunschweig, Germany

National Aeronautics and
Space Administration

January 2016

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an offical endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7115 Standard Drive
Hanover, MD 21076–1320
(301) 621–0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161–2171
(703) 605–6000

Abstract

In this paper we consider the solution of the compressible Navier-Stokes equations for a class of laminar airfoil flows. The principal objective of this paper is to demonstrate that members of this class of laminar flows have steady-state solutions. These laminar airfoil flow cases are often used to evaluate accuracy, stability and convergence of numerical solution algorithms for the Navier-Stokes equations. In recent years, such flows have also been used as test cases for high-order numerical schemes. While generally consistent steady-state solutions have been obtained for these flows using higher order schemes, a number of results have been published with various solutions, including unsteady ones. We demonstrate with two different numerical methods and a range of meshes with a maximum density that exceeds 8×10^6 grid points that steady-state solutions are obtained. Furthermore, numerical evidence is presented that even when solving the equations with an unsteady algorithm, one obtains steady-state solutions.

Contents

1 Introduction **2**

2 Governing Equations **3**
2.1 Physical Boundary Conditions 4

3 Structured RK/Implicit Scheme **4**
3.1 Elements of Solution Algorithm 5
3.2 Discrete Boundary Conditions: Structured Scheme 7

4 Unstructured RK/Implicit Scheme **8**
4.1 Description of Steady-State Solver for Unstructured Scheme 8
4.2 Description of Unsteady Solver for Unstructured Scheme 9
4.3 Discrete Boundary Conditions: Unstructured Scheme 11

5 Laminar Airfoil Flow Results **11**
5.1 Computational Grids 11
5.2 Subsonic flows (Cases 1–4) 12
5.2.1 Numerical solutions 13
5.2.2 Possible sources of differences in solutions 14
5.2.3 Effect of outer boundary location 15
5.2.4 Effect of skin-friction methods 16
5.2.5 Grid convergence of aerodynamic coefficients 16
5.3 Transonic flow (Case 5) 17

6 Concluding Remarks **18**

1 Introduction

In the development and evaluation of numerical schemes it is essential to have a comprehensive set of test problems. Such problems can allow specific properties of the solution algorithm to be validated. Furthermore, they can possibly delineate deficiencies in the algorithm that must be overcome. These test cases should provide either an analytic solution or a fully converged and resolved numerical solution. While experimental data can often provide confirmation for the validity of the computed solution, one should not necessarily conclude that everything is working correctly for a steady-state problem just because there is reasonable agreement with the data according to some engineering criterion (such as a small variation in the aerodynamic coefficients) when the residual has only been reduced a few orders of magnitude (rather than machine zero). An appropriate evaluation should involve a hierarchy of test cases. For example, if we consider solving the Navier-Stokes equations, the test cases should include laminar flows. Laminar flow test cases have the distinct advantage that they eliminate the need at sufficiently high Reynolds numbers to either resolve turbulence in a flow field or to model the turbulence appropriately.

Some possible laminar flow test cases involving the NACA 0012 airfoil were considered by the first author of this paper in 1984. One of these test cases (Mach number (M) of 0.5, angle of attack (α) of 0°, and Reynolds number (Re) equal to 5000) was introduced in the 1985 paper by Swanson and Turkel [1] to evaluate an algorithm for solving the compressible Navier-Stokes equations. This particular case was chosen because it has a small amount of trailing edge separation (beginning at approximately the 81% chord location), and thus, represented a good way to check the levels of dissipation being produced by the numerical scheme. For example, if the scheme is too dissipative, the effective Reynolds number is reduced, and the separation point moves downstream. Since that time, this case has been considered in numerous evaluations of schemes, such as Refs. [1–7]. It has also been used in a study of vortex-airfoil interaction by Svärd et al. [8]. In addition, this case and another one of the original cases ($\alpha = 3^\circ$) have also been used by Venditti [9] in evaluating a grid adaptive scheme for functional outputs of flow simulations.

In recent years, some additional laminar flow cases for the NACA 0012 airfoil have also been considered. The flow conditions are the same as for the original cases of Swanson except $\alpha = 1^\circ$ or $\alpha = 2^\circ$. These cases, as well as the original cases, have been considered in the European project ADIGMA to develop adaptive high-order variational methods for aerospace applications [10]. Due to the strong interest in these laminar flow cases, there is a need to have documentation of their solutions.

Some recent efforts to compute members of this class of flow problems have produced solutions that are inconsistent with the rather large number of previous solutions (computed with a wide range of numerical methods), including those already cited in this paper. For example, Abgrall and De Santis [11] show unexpected solutions for the $\alpha = 0^\circ$ case, which should have a symmetric solution based on previous results, using both second-order and third-order discretizations. One possibility for such results is the presence of an asymmetry in their solution algorithm. Another example concerns the unsteady solution for the $\alpha = 2^\circ$ case presented in the ADIGMA Project [10] paper by Taube et al. starting on page 427. Even though the computational method included adaptation, the calculation was performed on a rather coarse mesh (815 cells) and the adaptation seems to have been focused in the wake. Thus, there could have been insufficient resolution in the neighborhood of the separation point, which can produce a result that appears to be an unsteady solution. There is also the possibility that the solution algorithm is not sufficiently strong, as measured by the level of implicitness. In this paper, we will show how a weak solution algorithm can produce what appears to be an unsteady solution. Dolejší [12] has also obtained an unsteady solution for this laminar flow case. A nonlinear system of algebraic equations (from discontinuous Galerkin finite element method) is solved with an inexact Newton-time method. An inner iterative method is required to solve a linear system at each time level. Since the residual (algebraic error) of the iteration is not being reduced to approximately zero, there can be sufficient error to produce an unsteady result. In any of these examples, one cannot dismiss the possible influence of boundary conditions. It is

difficult to assess this because descriptions of the discrete implementation of the boundary conditions for the problems being considered are not given.

As a consequence of such examples, a primary objective of this paper is to demonstrate that there are steady-state solutions, even on high density meshes, to this class of problems. Two additional objectives of this paper are as follows. The first of these is to examine and document the behavior of a set of laminar flow problems, which includes those just discussed, as well as a commonly considered lower Re case with a higher angle of attack. The second additional objective is to compare two methods for solving the compressible Navier-Stokes equations. One method [13, 14] is based on structured grids, and the other [15] is based on unstructured grids. Both methods apply a finite-volume approach for spatial discretization. The structured grid method uses a cell-centered formulation, and the unstructured grid method uses a node-centered formulation. Each method applies a matrix form for numerical dissipation. In addition, using the structured grid approach a Roe dissipation (which is frequently used in numerical schemes) is also considered. When comparing what we call structured and unstructured schemes it is important to clarify the meaning of this terminology with respect to contrasting the schemes. In the paper, the same set of structured grids (i.e., quadrilateral elements) is used for both methods. Thus, the comparison of the schemes is based on the fact that there are differences in their elements and implementation.

For all computations in the paper we solve the full Navier-Stokes equations (i.e., all physical diffusion terms are retained). The solvers for the two methods are comprised of a Runge-Kutta (RK) scheme with an implicit preconditioner (herein also designated as RK/Implicit scheme) to extend stability and allow a large Courant-Friedrichs-Lewy (CFL) number. A multigrid scheme is employed for convergence acceleration. In the comparison of the methods, we consider accuracy, stability, and convergence when computing solutions of the laminar flow cases.

In the first section of the paper, we present the governing flow equations and the boundary conditions for the continuous problem of flow past an airfoil. The next two sections describe and discuss the structured and unstructured methods, including the discrete boundary conditions, that are considered. After delineating the specific differences in the two methods, we then present a set of laminar flow results that compare the methods. Convergence histories are shown for both steady-state and unsteady computations. Comparisons are made between the results with the structured and unstructured methods. The effects of mesh density on the solutions, which contain between $8{,}192$ and about 8.39×10^6 cells, as well as the surface pressure and skin-friction distributions for each laminar flow case are presented and discussed. In all cases, there is flow separation, and both the separation location and downstream extent of separation are compared. Overall flow details are also illustrated with Mach and streamline plots. Such details provide a reference for other investigators to compare against.

2 Governing Equations

We consider the two-dimensional (2-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

$$\iint_{\mathcal{V}} \frac{\partial \mathbf{W}}{\partial t} d\mathcal{V} + \int_{\mathcal{S}} \mathcal{F} \cdot \mathbf{n} d\mathcal{S} = 0, \tag{1}$$

where $\mathbf{W}$ 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, surface, and 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

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

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

$$\bar{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \end{bmatrix}, \qquad \mathbf{Q} = k \begin{bmatrix} \partial T/\partial x \\ \partial T/\partial y \end{bmatrix} \tag{3}$$

with

$$\tau_{xx} = \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + 2\mu \frac{\partial u}{\partial x}, \quad \tau_{yy} = \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) + 2\mu \frac{\partial v}{\partial y}, \quad \tau_{xy} = \tau_{yx} = \lambda \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right). \tag{4}$$

Here, the symbol ∂ indicates partial differentiation, μ and λ are the first and second coefficients of molecular viscosity, k denotes the coefficient of thermal conductivity and T represents the temperature. By the Stokes hypothesis, $\lambda = -2/3\mu$, and μ is determined with Sutherland's viscosity law [16]. The thermal conductivity k is evaluated with the constant Prandtl number assumption (i.e., the Prandtl number $Pr = 0.72$).

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

$$p = \rho R T \tag{5}$$

where R is the specific gas constant.

2.1 Physical Boundary Conditions

In the continuum case, we consider an external flow problem (i.e., flow past a given geometry) on an infinite domain $\Omega \subset \mathbb{R}^2$. Thus, we need to define appropriate conditions at a wall boundary, which we assume to be solid. Later, in the discrete case, we define a finite domain. It will then be necessary to define suitable inflow and outflow boundary conditions.

For viscous flows, the non-penetration condition and the no-slip condition are imposed,

$$\mathbf{q} \cdot \mathbf{n} = 0, \quad \mathbf{q} \cdot \mathbf{t} = 0, \tag{6a}$$

where $\mathbf{t}$ is the unit vector tangent to the surface of a control volume. In addition, a boundary condition is required to determine the temperature and one other thermodynamic variable. For the temperature, we impose the adiabatic condition

$$\mathbf{Q} \cdot \mathbf{n} = 0, \tag{6b}$$

where $\mathbf{Q}$ is the heat flux vector given in (3). Usually, for the other thermodynamic variable required, the pressure is selected, which is determined from the normal momentum equation. For example, consider the differential form of the normal momentum equation at a solid and stationary wall boundary. This equation can be written as

$$(\nabla p \cdot \mathbf{n})_w = \left[(\nabla \cdot \bar{\tau}) \cdot \mathbf{n} \right]_w, \tag{6c}$$

where ∇ is the gradient operator.

3 Structured RK/Implicit Scheme

We apply a finite-volume approach to discretize the fluid dynamic equations and use a three point second-order discretization of the advection terms with either the matrix-type or Roe-type numerical dissipation. For details of these forms of numerical dissipation see Swanson and Turkel [17] and

Roe [18]. The viscous terms are discretized with a second-order central difference approximation. Using this discretization, we describe in this section the solution algorithm for the structured scheme.

The numerical solver called the RK/Implicit scheme is a member of a general class of schemes with a nonlinear outer iteration and a linear inner iteration (e.g., see [15, 19]). In the outer iteration, a Full Approximation Storage (FAS) multigrid method with a RK smoother such as those of Ref. [14] is applied. The inner iteration determines an approximate implicit preconditioner for the RK scheme, resulting in a type of implicit RK smoother. The particular implementations for the structured grid and unstructured grid methods are described below in Section 3.1 and later in Section 4.1, respectively.

3.1 Elements of Solution Algorithm

A semi-discrete form of Eq. (1) can be written as (omitting subscripts of discrete quantities for convenience),

$$\frac{d\mathbf{W}}{dt} + R(\mathbf{W}) = 0, \tag{7}$$

defining a system of ordinary differential equations. The $R(\mathbf{W})$ denotes the residual function determined by the spatial discretization briefly described previously. Applying a three-stage RK scheme to update the discrete solution, we obtain on the q-th stage

$$\mathbf{W}^{(q)} = \mathbf{W}^{(0)} + \delta\mathbf{W}^{(q)}, \tag{8}$$

where the change in the solution vector $\mathbf{W}$ is

$$\delta\mathbf{W}^{(q)} = \mathbf{W}^{(q)} - \mathbf{W}^{(0)} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \mathcal{L}\, \mathbf{W}^{(q-1)} = -\alpha_q \Delta t R(\mathbf{W}), \tag{9}$$

and $\mathcal{L}$ is the complete difference operator for the system of equations, and $\mathcal{V}$ is the volume of the mesh cell being considered. Here, α_q is the RK coefficient of the q-th stage, and Δt is the local time step. To extend the support of the difference scheme, we consider an implicit residual smoothing approach [20]. Adapting the smoothing technique to a system of equations, we have the following:

$$\mathcal{L}_i\, \overline{\delta\mathbf{W}}^{(q)} = \delta\mathbf{W}^{(q)}, \tag{10}$$

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

$$\overline{\delta\mathbf{W}}^{(q)} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \mathcal{P}\, \mathcal{L}\, \mathbf{W}^{(q-1)} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \mathcal{P} \sum_{\text{all faces}} \mathbf{F}_n^{(q-1)}\, S, \tag{11}$$

where $\mathcal{P}$ is a preconditioner defined by the approximate inverse $\widetilde{\mathcal{L}}_i^{-1}$, and the summation denotes a flux balance on a mesh cell. We can also view Eq. (11) as Eq. (9) with $R(\mathbf{W})$ replaced by $\mathcal{P}R(\mathbf{W})$, which we will denote as $\overline{R}(\mathbf{W})$. The change $\overline{\delta\mathbf{W}}^{(q)}$ replaces the explicit update appearing in Eq. (8). Thus, each stage in the RK scheme is preconditioned by an implicit operator.

A first order upwind approximation based on the Roe scheme is used for the convective derivatives in the implicit operator. To derive this operator, one treats the spatial discretization terms in the flow equations implicitly and applies linearization. For a detailed derivation see Rossow [13]. Substituting for the implicit operator in Eq. (10), we obtain for the q-th stage of the RK scheme

$$\left[I + \varepsilon \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{A}_n\, S \right] \overline{\delta\mathbf{W}}^{(q)} = -\alpha_q \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{F}_n^{(q-1)}\, S = -\alpha_q \Delta t \widehat{\mathbf{R}}^{(q-1)}, \tag{12}$$

where the matrix $\mathbf{A}_n$ is the flux Jacobian associated with the normal flux density vector $\mathbf{F}_n$ at a cell face, $\widehat{\mathbf{R}}^{(q-1)}$ represents the residual function for the $(q-1)$-th stage, and ε is a relaxation parameter. The parameter ε is taken to be 0.5.

The matrix $\mathbf{A}_n$ can be decomposed into $\mathbf{A}_n^+$ and $\mathbf{A}_n^-$, which are defined by

$$\mathbf{A}_n^+ = \frac{1}{2}\left(\mathbf{A}_n + |\mathbf{A}_n|\right), \quad \mathbf{A}_n^- = \frac{1}{2}\left(\mathbf{A}_n - |\mathbf{A}_n|\right). \tag{13}$$

If we substitute for $\mathbf{A}_n$ in Eq. (12) using the definitions of Eq. (13), then the implicit scheme can be written as

$$\left[I + \varepsilon \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{A}_n^+ S\right] \overline{\delta \mathbf{W}}_{i,j}^{(q)} = -\alpha_q \Delta t \widehat{\mathbf{R}}_{i,j}^{(q-1)} - \varepsilon \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{A}_n^- \overline{\delta \mathbf{W}}_{NB}^{(q)} S, \tag{14}$$

where the indices (i, j) indicate the cell of interest, and NB refers to all the direct neighbors of the cell being considered. Consider traversing the boundary of a mesh cell in a counterclockwise manner, with the positive surface normal vector always pointing outward from the cell. Then, as discussed by Rossow [21], 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) 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)). These flux density changes are determined only by information from within the cell (i, j).

In order 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. (12) to the primitive variables $\mathbf{U} = [\rho \; p \; u \; v]^T$. Thus,

$$\left[I + \varepsilon \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{P}_n S\right] \overline{\delta \mathbf{U}}^{(q)} = -\frac{\partial \mathbf{U}}{\partial \mathbf{W}} \alpha_q \frac{\Delta t}{\mathcal{V}} \sum_{\text{all faces}} \mathbf{F}_n^{(q-1)} S, \tag{15}$$

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

$$\mathbf{P}_n = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \mathbf{A}_n \frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \left(\mathbf{A}_n^+ + \mathbf{A}_n^-\right) \frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \mathbf{P}_n^+ + \mathbf{P}_n^-. \tag{16}$$

The Jacobian $\partial \mathbf{U}/\partial \mathbf{W}$ on the right-hand side of Eq. (15) must multiply the conservative flux balance in order to ensure conservation. Using the definitions of Eq. (13) and the dissipation matrix, which is defined in the appendix of Ref. [14], one can determine the matrices $\mathbf{P}_n^+$ and $\mathbf{P}_n^-$, which are also given in that 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 primitive variables (see Ref. [22]). Note that $\overline{\delta \mathbf{U}}$ is simply proportional to a new residual $\overline{R}(\mathbf{U})$ that requires transforming back to $\overline{R}(\mathbf{W})$.

To solve Eq. (14) for the changes in conservative variables $\overline{\delta \mathbf{W}}_{i,j}^{(q)}$, the 4×4 matrix on the left-hand side of Eq. (14) must be inverted. It is sufficient to approximate the inverse of the implicit operator. An adequate approximate inverse is obtained with two symmetric Gauss-Seidel sweeps. The relaxation procedure uses lexicographic ordering of the solution points. To initialize the iterative process, the unknowns are set to zero.

For the RK/Implicit scheme, the relaxation is a combination of point and line SGS relaxation, which is the method frequently used with unstructured schemes. With line SGS, we apply the implicit operator across the wake line (the line emanating from the trailing edge of an airfoil), which we call a fully implicit treatment. It is also possible to lag the boundary condition at the wake line rather than extending the line solve across the wake line. This approach can also work well provided local relaxation or subiterations are performed to compensate for the lagging of the wake boundary condition. Although this may be a more convenient option from the implementation point of view for multiblock structured grid methods, we use the the fully implicit method in this paper to obtain

a measure of the best possible convergence with the line SGS relaxation part of the smoother for the multigrid method. Furthermore, a variation on this wake treatment can be applied to unstructured grid methods by appropriate reordering of the unknowns. The linear stability and high-frequency damping properties of the basic scheme have been investigated using Fourier analysis. As indicated in Refs. [14,23,24], this scheme has good smoothing properties, making it well suited as a smoother for a multigrid algorithm.

Due to the upwind approximation used for the implicit operator, the coefficients for the RK scheme are also based on an upwind scheme. For computational efficiency, we apply a three-stage RK scheme with the coefficients

$$[\alpha_1, \alpha_2, \alpha_3] = [0.15,\ 0.4,\ 1.0] \tag{17}$$

from Ref. [25]. A weighting of the physical and numerical dissipation is used in order to extend the parabolic stability limit of the scheme. Then, the residual function $\mathbf{R}^{(q)}$ is defined by

$$\mathbf{R}^{(q)} = \mathbf{R}(\mathbf{W}^{(q)}) = \frac{1}{\mathcal{V}} \left[\mathcal{L}_c \mathbf{W}^{(q)} + \sum_{r=0}^{q} \gamma_{qr}\ \mathcal{L}_v \mathbf{W}^{(r)} + \sum_{r=0}^{q} \gamma_{qr}\ \mathcal{L}_d \mathbf{W}^{(r)} \right], \tag{18}$$

with $\sum \gamma_{qr} = 1$ for consistency. The operators $\mathcal{L}_c$, $\mathcal{L}_v$, and $\mathcal{L}_d$ relate to the convective, viscous, and numerical dissipative terms. The coefficients γ_{qr} are the weights of the viscous and dissipative terms on each stage. Thus, the dissipation is weighted on the second and third stages with that of the previous stages. For the present three-stage scheme, the weights corresponding to each RK stage are as follows: $[1,\ 0.5,\ 0.5]$. Analysis and numerical experiments have demonstrated that this weighting augments the robustness of the algorithm. Additional discussion of weighting is given in [14, 22]. A comparison of performance of this three-stage scheme with that of a frequently used five-stage scheme [6], which also applies weighting that results in a similar parabolic limit, has been performed. This comparison demonstrated essentially the same convergence behavior for the two RK schemes.

In the application of the three-stage RK/Implicit scheme as the smoother of a multigrid method, the CFL number is increased to 1.0×10^6 after 10 multigrid cycles. When computing difficult flow cases, sometimes it is necessary to use some additional cycles (typically 5 cycles as done in Case 5 of this paper) to increase the CFL to 1.0×10^6. For the ramping procedure see [26]. A W-type cycle is used to execute the multigrid. Details of the multigrid method are given in the paper by Swanson et al. [14].

3.2 Discrete Boundary Conditions: Structured Scheme

A critical element in the solution of any system of partial differential equations such as the Navier-Stokes equations is the selection of proper boundary conditions. The choice of conditions must be consistent with physical constraints of the problem of interest as well as the interior discrete formulation. Moreover, the physical conditions, in general, must be supplemented with a sufficient number of numerical relations to allow determination of all dependent variables. In addition to defining the conditions at wall boundaries, which can be solid or porous, the infinite domain problem must be adequately simulated in the case of external flows. This is usually done by delineating boundaries at some distance from the primary region of consideration, and then prescribing suitable boundary conditions for that location. In the case of a lifting airfoil, the outer boundary position must be sufficiently far so as to not compromise the development of the lift. For example, five airfoil chords may be too close for subcritical flows, and it is generally too close for transonic flows. A choice of 10 to 20 chords is often found to be satisfactory. At the same time, we should point out that even for inviscid nonlifting flow over a circular cylinder an outer boundary placed too close can cause inaccurate prediction of the flow over the aft portion of the cylinder. There is an increased flexibility in the location of the outer boundary if the far-field vortex effect [27] is taken into account.

The calculations with the structured grid method include this effect. Further discussion about the boundary location and far-field vortex effect for the laminar airfoil flows considered in this paper will be given later in the section on numerical results.

When considering a solid boundary for the structured grid algorithm, a row of auxiliary cells is created exterior to the domain of the flow. For viscous flows with no surface transpiration, the nonpenetration and no-slip conditions are required, and this is imposed by treating the Cartesian velocity components as antisymmetric functions with respect to the solid surface. Thus,

$$u_{i,1} = -u_{i,2}, \qquad v_{i,1} = -v_{i,2}.$$

The surface values of pressure p and temperature T are computed using the reduced normal momentum and energy equations

$$\frac{\partial p}{\partial \eta} = 0, \qquad \frac{\partial T}{\partial \eta} = 0, \tag{19}$$

where η is the coordinate normal to the surface, which is associated with a transformation from Cartesian coordinates (x, y) to computational coordinates (ξ, η).

In order to compute the unknown flow variables at the outer boundary of an external aerodynamics problem, we appeal to characteristic theory, some simplifying assumptions, and the concept of a point vortex. Using this approach, one can easily derive the Riemann invariants associated with the normal direction to the boundary. The number of quantities specified locally at the outer boundary depend on whether there is inflow or outflow. For an inflow boundary, the tangential velocity (q_t), the Riemann invariant (R^+) corresponding to an incoming characteristic, and the entropy (s) are specified by their free-stream values, and the Riemann invariant associated with the characteristic from the interior of the domain (R^-) is extrapolated. If there is outflow, there is only one incoming characteristic, and only the free-stream R^+ is specified; the quantities q_t, R^-, and s are extrapolated from the interior of the domain. At the downstream outer boundary of the C-type mesh, only the free-stream value of the pressure is given, while the dependent variables ρ, ρu, and and ρv are extrapolated from the interior. For additional details and discussion see Ref. [1].

4 Unstructured RK/Implicit Scheme

The unstructured algorithm employs a node centered, finite volume space discretization on general meshes. The computational mesh, which is often called a dual mesh, is constructed from the primary grid in a preprocessing step. The dual grid forms the control volumes with the unknowns at vertices of the primary grid.

We discretize the inviscid part by a central difference scheme with an added matrix–valued artificial viscosity [17, 28]. On a Cartesian mesh, the discretization can be shown to be second order. To handle shocks, a pressure switch is included into the dissipative part. This reduces the order of the method in the neighborhood of shock waves to first order. To deal with highly stretched meshes, a cell stretching coefficient is included into the scheme. Derivatives required to discretize the viscous terms are computed by a Green–Gauss method. For further details about the discretization strategy we refer to [19, 29].

4.1 Description of Steady-State Solver for Unstructured Scheme

To obtain an update to the solution vector for the flow equations, we use the three-stage RK scheme (9) with the stage coefficients (17). Compared to (10), the operator $\mathcal{L}_i$ is defined by

$$\mathcal{L}_i := I + \varepsilon \frac{\Delta t}{\mathcal{V}} \frac{\partial \widehat{\mathbf{R}}^{(q-1)}}{\partial \mathbf{W}} \left[\mathbf{W}^{(q-1)} \right]. \tag{20}$$

The exact construction of $\frac{\partial \widehat{\mathbf{R}}^{(q-1)}}{\partial \mathbf{W}}$ is avoided by considering matrix-free Krylov subspace methods, such as the Generalized Minimal Residual (GMRES) method. Then the construction of $\frac{\partial \widehat{\mathbf{R}}^{(q-1)}}{\partial \mathbf{W}}$ is done by approximating the corresponding matrix vector product by a symmetric finite difference,

$$\frac{\partial \widehat{\mathbf{R}}^{(q-1)}}{\partial \mathbf{W}} \left[\mathbf{W}^{(q-1)}\right] \mathbf{h} \quad \approx \quad \frac{1}{2\delta} \left\{ \widehat{\mathbf{R}} \left(\mathbf{W}^{(q-1)} + \delta \mathbf{h} \right) - \widehat{\mathbf{R}} \left(\mathbf{W}^{(q-1)} - \delta \mathbf{h} \right) \right\}, \delta := 10^{-6}. \tag{21}$$

To approximately solve the linear systems (10), we apply a left preconditioned GMRES method [30]; that is, we approximately solve

$$\left(\mathbf{Prec}_j\right)^{-1} \mathcal{L}_j \overline{\delta \mathbf{W}}^{(j)} = -\alpha_j \left(\mathbf{Prec}_j\right)^{-1} \delta \mathbf{W}^{(j)} \tag{22}$$

by the algorithm

- Compute $\mathbf{r}_0 := -\alpha_j \left(\mathbf{Prec}_j\right)^{-1} \delta \mathbf{W}^{(j)}, \beta := \|\mathbf{r}_0\|_2, \mathbf{v}_1 := \frac{1}{\beta}\mathbf{r}_0$
- for $j = 1, \ldots, m$
  - $\mathbf{w} := \left(\mathbf{Prec}_j\right)^{-1} \mathcal{L}_j \mathbf{v}_j$
  - for $i = 1, \ldots j$
    - $h_{i,j} := \langle \mathbf{w}, \mathbf{v}_i \rangle$
    - $\mathbf{w} := \mathbf{w} - h_{i,j}\mathbf{v}_i$
  - $h_{j+1,j} = \|\mathbf{w}\|_2, \mathbf{v}_{j+1} = \frac{1}{h_{j+1,j}}\mathbf{w}$
  - $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_m), H_m = (h_{i,j})_{1 \le i \le j+1, 1 \le j \le m}$
  - Solve $\mathbf{y}_m := \text{argmin}_{\mathbf{y}} \|\beta \mathbf{e_1} - H_m \mathbf{y}\|_2$ by Given's-rotations
  - $\overline{\delta \mathbf{W}}^{(j)} := \overline{\delta \mathbf{W}}^{(j),\text{approx}} = \mathbf{V}_m \mathbf{y}_m$

From the equation for $\mathbf{w}$ of the GMRES algorithm above it is obvious that for our preconditioner $\mathbf{Prec}_j$ we need to approximately solve the linear systems

$$\mathbf{Prec}_j \mathbf{w} = \mathcal{L}_j \mathbf{v}_j. \tag{23}$$

So, the preconditioner consists of two parts:

a) the linear operator $\mathbf{Prec}_j$ itself,

b) an iterative solution method for approximately solving the linear systems (23) required for GMRES.

For a detailed description for the construction of both the operator $\mathbf{Prec}_j$ and the iterative solution method, we refer to [19]. In the application of the three-stage RK/Implicit scheme as the smoother of a multigrid method, the CFL number is increased to 1000 after 10 multigrid cycles. During the first 10 cycles, the CFL number is 10. A W-type cycle is used to execute the multigrid. Details of the multigrid method are given in the paper [19]. For all our computations, we used a fixed number of GMRES iterations, namely 10.

4.2 Description of Unsteady Solver for Unstructured Scheme

To confirm the independency of steady results for the considered laminar flow cases, we additionally carry out unsteady computations. To this end, we assume that after spatial discretization, we have the following semi-discrete set of ordinary differential equations

$$\frac{d}{dt}\left(\overline{\mathbf{M}}(t)\mathbf{W}(t)\right) = -\mathbf{R}(t, \mathbf{W}(t)), \qquad \overline{\mathbf{M}}(t) := \text{diag}\left(\text{Vol}(\Omega_i)(t)\right), \tag{24}$$

which needs to be integrated in time. Since we do not consider mesh deformations, the mass matrix $\overline{M}$ is constant in time; thus, Eq. (24) is simplified to

$$\frac{d}{dt}\left(\mathbf{M}\mathbf{W}(t)\right) = -\mathbf{R}(t, \mathbf{W}(t)), \qquad \mathbf{M} := \text{diag}\left(\text{Vol}(\Omega_i)\right). \tag{25}$$

To integrate (25) in time, we apply a BDF2 method, which is a member of the class of backward differentiation formula (BDF) schemes [31]. This method. is widely used for its stability, moderate accuracy and computational efficiency. Then a fully discrete form of (25) is given by

$$\frac{3\mathbf{M}\mathbf{W}^{n+1} - 4\mathbf{M}\mathbf{W}^{n} + \mathbf{M}\mathbf{W}^{n-1}}{2\Delta t} = -\mathbf{R}(\mathbf{W}^{n+1}), \tag{26}$$

where the superscript denotes the time level, $\mathbf{W}^n := \mathbf{W}(t_n)$ and Δt the physical time step. Equation (26) in general represents a large scale nonlinear equation, which needs to be solved approximately such that the outer time iteration is not influenced by the approximation error. To solve (26) we rearrange the equation and obtain

$$\frac{3}{2\Delta t}\mathbf{M}\mathbf{W}^{*} = -\mathbf{R}(\mathbf{W}^{*}) + \mathbf{Z}, \; \mathbf{Z} := \frac{1}{\Delta t}\left(2\mathbf{M}\mathbf{W}^{n} - \frac{1}{2}\mathbf{M}\mathbf{W}^{n-1}\right). \tag{27}$$

The approximate solution $\mathbf{W}^{*}_{\text{app}}$ of (27) is then our new time level $\mathbf{W}^{n+1} := \mathbf{W}^{*}_{\text{app}}$. To apply the existing steady-state solution procedure to find some approximate solution of (27), we introduce a pseudo-time τ. Then, we may apply our steady-state solution procedure as an inner iterative procedure to

$$\frac{d\mathbf{W}^{*}}{d\tau} = \mathbf{R}^{*}(\mathbf{W}^{*}), \qquad \mathbf{R}^{*}(\mathbf{W}^{*}) := \frac{3}{2\Delta t}\mathbf{W}^{*} + \mathbf{M}^{-1}\left(\mathbf{R}(\mathbf{W}^{*}) - \mathbf{Z}\right). \tag{28}$$

Hence, to approximately solve Eq. (28), we modify method (12) to

$$\begin{aligned}
\mathbf{W}^{(0)} &:= \mathbf{W}^{n} = \mathbf{W}(t_n) \\
\mathbf{W}^{(1)} &= \mathbf{W}^{(0)} - \text{CFL}\alpha_{21}\Delta T\mathbf{P}_1^{-1}\mathbf{R}^{*}\left(\mathbf{W}^{(0)}\right) \\
&\vdots \\
\mathbf{W}^{(s)} &= \mathbf{W}^{(0)} - \text{CFL}\alpha_{s+1,s}\Delta T\mathbf{P}_s^{-1}\mathbf{R}^{*}\left(\mathbf{W}^{(s-1)}\right) \\
\mathbf{W}(t_{n+1}) = \mathbf{W}^{n+1} &= \mathbf{W}^{(s)},
\end{aligned} \tag{29}$$

where $\Delta T := \text{diag}(\Delta\tau_i)$ denotes the matrix of local pseudo time steps. The sequence of preconditioners $\mathbf{P}_j$ are given by some approximate derivative of the residual $\mathbf{R}^{*}$. Differentiation of (28) with respect to $\mathbf{W}^{*}$ yields

$$\frac{\partial\mathbf{R}^{*}}{\partial\mathbf{W}^{*}} = \frac{3}{2\Delta t}\mathbf{I} + \mathbf{M}^{-1}\frac{\partial\mathbf{R}}{\partial\mathbf{W}^{*}}. \tag{30}$$

With respect to the analysis given in [15], we conclude that using (30) the preconditioner required to solve the inner nonlinear equations (28) is given by

$$\mathbf{P}_j := \left(\frac{1}{\text{CFL}}(\Delta T)^{-1}\mathbf{M} + \left(\frac{3}{2\Delta t}\mathbf{I} + \frac{\partial\mathbf{R}^{1st}}{\partial\mathbf{W}^{*}}\right)\right).$$

To approximately solve the corresponding linear systems five symmetric Gauss-Seidel sweeps are applied. Then, algorithm (29) is used as a smoother in a FAS multigrid method to approximately solve (28).

4.3 Discrete Boundary Conditions: Unstructured Scheme

When considering a node centered grid metric at the boundary, we utilize what are called half cells. As a consequence, the unknowns are located directly on the boundary, which is different from the cell centered scheme that was used for the structured computer code. Thus, imposing conditions (6a) and (6b) for a no-slip boundary flux, the convective distribution is evaluated by

$$\mathcal{F}_{c,\mathrm{no-slip}} = \begin{bmatrix} 0 \\ p_{\mathrm{no-slip}}\mathbf{e}_x \\ p_{\mathrm{no-slip}}\mathbf{e}_y \\ 0 \end{bmatrix}, \qquad p_{\mathrm{no-slip}} = (\gamma - 1)\rho E.$$

Here the equation for $p_{\mathrm{no-slip}}$ follows from the equation of state (5) using vanishing velocity, and ρE is determined using the adiabatic wall condition. To evaluate the viscous flux, the required velocity gradients on the boundary are approximated by

$$\frac{\partial u_{\mathbf{p}_b}^{(l)}}{\partial x_k} \approx n_k \frac{-u_{\mathbf{p}_n}^{(l)}}{\|\mathbf{x}_{\mathbf{p}_b} - \mathbf{x}_{\mathbf{p}_n}\|_2}, \qquad l, k = 1, 2,$$

where $\mathbf{x}_{\mathbf{p}_n}$ denotes the closest inner mesh in the normal direction to the boundary point $\mathbf{x}_{\mathbf{p}_b}$, and $u_{\mathbf{p}_n}$ are the unknowns corresponding to the velocity located at the mesh point $\mathbf{x}_{\mathbf{p}_n}$. Then the stress tensor $\overline{\tau}$ defined in (3) is evaluated using these approximate gradients, and the term $\mathbf{Q} = 0$ because of condition (6b). Finally, to implement the boundary condition in full correlation to the inner fluxes, dissipative terms are constructed and added for the boundary edge. Such an implementation of the boundary condition is close to mimicking the cell centered approach where a row of auxiliary cells is created exterior to the domain (see Section 3.2).

At the outer boundary, the same philosophy is followed as for the no-slip wall. After initialization of the flow solver, the far-field values $(\rho_\infty, (\rho u)_\infty, (\rho v)_\infty, (\rho E)_\infty)$ define the exterior state. These values are used to compute the full flux for the exterior boundary edge. For the inner state, the variables for the unknowns located on the boundary edge are used. Note that this procedure automatically mimics the characteristics because of the construction of the Roe matrix weighting the dissipative terms. The outer boundary conditions do not include the far-field vortex effect. However, as we demonstrate in the next section, this has a relatively small effect on the laminar flow solutions considered in this paper.

5 Laminar Airfoil Flow Results

We have considered five cases, and the flow conditions are presented in Table 1. There is flow separation in each case, with the extent of separation ranging from trailing-edge type to extensive type (i.e., starting at approximately $x/c = 0.36$ on the upper surface of the airfoil and extending to the trailing edge of the airfoil). As indicated previously, we have also considered Case 5, which has a lower Re and a higher angle of attack, because it is a frequent test case for laminar flow, and it has extensive separation. For all the calculations, we used structured meshes with a C-type topology.

5.1 Computational Grids

Two families of grids were generated. Starting on the finest grid, each successive grid in a family was obtained by eliminating every other mesh line in each coordinate direction. The finest mesh in the family of grids for the first four cases contains 4,096 cells around the airfoil, with half of the cells on the airfoil and the other half in the wake. Comparable emphasis is given to the wake region in order to capture the closure of the recirculation region occurring in the laminar flow cases. There are 2,048 cells in the radial (normal) direction. The outer boundary of the solution domain for

Cases 1–4 is located at 10 chords away from the airfoil. Later in this section, the effect of the outer boundary location on the computed aerodynamic coefficients is discussed. In Table 2, we define a set of parameters describing the finest grids for the test cases. For comparison purposes, the parameters for the 1024×512 grid are also given. These parameters include the minimum normal mesh spacing (Δy_{min}) at the surface boundary, characteristic spacings at the leading edge (LE) and trailing edge (TE) of the airfoil, and the tangential spacing at the mid-chord location. Figures 1 and 2 show the near field and enlargements of the LE and TE regions of the 512×256 cell grid. In the fifth case, we needed to generate another grid with the outer boundary located farther away (20 chords), due to transonic disturbances and the large extent of flow separation. In addition, the mesh density in the circumferential direction was increased in order to have appropriate resolution in the neighborhood of the separation location. Otherwise, it is possible that the flow may appear to be unsteady. The mesh density of this grid is 1280×512 (with 768 cells on the airfoil), and the Δy_{min} along with other grid parameters are given in Table 2. For all grids, there is clustering of the grid points at the LE and TE of the airfoil.

The discrete boundary conditions of the two numerical schemes have been described succinctly in sections 3.2 and 4.3. Additional discussion of the boundary conditions is given in Swanson and Turkel [22] and Galle et al. [32] (see also Langer et al. [33]) for the structured and unstructured schemes, respectively. The initial solution on each computational grid is determined from free-stream conditions. As an additional numerical test to evaluate sensitivity to asymmetry in the initial solution for Case 1 ($\alpha = 0^\circ$), we have also considered an initial solution generated using a random number generator.

In the next section (5.2), we initially describe convergence behavior of the computations. The following five sections concern the flow solutions. Section 5.2.1 discusses the details of the numerical solutions for the first four cases. Then, Section 5.2.2 identifies possible sources of differences in the solutions on successive grids. This is followed by Sections 5.2.3-5.2.4 that provide details concerning the effects on the solutions and computed results due to outer boundary location, far-field vortex, and method of computing skin-friction. The final discussion about the first four cases is given in Section 5.2.5, which presents examples of grid convergence for the aerodynamic coefficients. To conclude the results section, we describe and discuss the details concerning the transonic laminar flow of Case 5.

5.2 Subsonic flows (Cases 1–4)

In comparing the structured and unstructured grid methods, we first consider the convergence histories for the five laminar flow cases. Figures 3– 11 show how the L_2 norm of the residual of the continuity equation changes with multigrid cycles. The convergence histories are given for four to six grids, depending upon the solution algorithm used, and include the finest grid considered. Currently, the structured grid computer code does not have parallel processing capability, which prevents computations on the two finest grids, which have approximately 2.1 and 8.4 million cells. The residual on all grids is reduced to essential or absolute machine zero. All of the convergence plots exhibit a similar behavior for the first four grids, where the finest grid has 1024×512 cells. There is faster convergence with the structured grid method. Moreover, in all of the laminar flow cases, machine zero convergence is attained with the structured grid method in roughly 90 cycles. With the unstructured grid method, the convergence on the first four grids is attained between about 120 and 300 cycles for Cases 1–4. There is a significant slowdown in the convergence rate for the two finest grids.

To further substantiate the existence of steady-state solutions for Cases 1–4, calculations were performed in an unsteady mode with the unstructured grid algorithm (see Section 4.2 for details). For all cases the 512×256 grid was used, and two different time steps (1.0×10^{-3} and 1.0×10^{-4}) were considered. The time step is nondimensionalized [22] by the reference length (airfoil chord) divided by the free-stream speed of sound $\bar{c}/\sqrt{\gamma}$. Convergence histories for the unsteady computations are shown in Figs. 7 – 10. The number of multigrid cycles required to converge

the solution at each physical time step is determined by a preset convergence criterion of 10^{-12}. We should point out that even if this criterion is satisfied, the solution algorithm still performs a minimum of five multigrid cycles for the inner iteration (sometimes called a subiteration) to solve the nonlinear system (28). In each case, we observe that after a given number of cycles, which is comparable to approximately 0.035 in nondimensional time, a steady-state solution is attained independent of the time step. Thus, we obtain the same flow topology as for the original steady-state computations. Note that the eventual drop of the residual to machine zero near the completion of each calculation is a consequence of the minimum requirement of five cycles for each time step.

Now, having demonstrated the existence of stationary solutions for all the laminar flow cases considered, we address the question of why numerical difficulties can be encountered. One can acquire enormous insight into the reason by performing linear stability analysis (i.e., eigensystem analysis) such as that of Roberts and Swanson [23] and Langer [34]. The analysis of Langer includes the additional advantage that the actual problem being considered is used in the analysis. From such analyses, a distinction can be made between what may be called weak and strong solution algorithms. The measure of weak and strong can be assessed based on the level of implicitness in the algorithm (i.e., the more implicit an algorithm is, the stronger it is). On this basis, we observe that when numerical problems are encountered with a given algorithm, this can often be attributed to the solution algorithm having a weak measure. Moreover, when an algorithm has a weak measure and numerical difficulties are encountered, then the spectral radius of the amplification matrix for the linear system can have eigenvalues that are close to, even perhaps exceeding, one.

5.2.1 Numerical solutions

In Figs. 12 and 13, the surface pressure coefficient (C_p) distributions of Case 1 for the structured and unstructured grid methods are shown. Hereafter, we refer to the structured and unstructured grid methods as SGM and USGM, respectively. All plots include the C_p distributions for the grids considered with each solution algorithm. The presence of the boundary layer is evident by the absence of a strong pressure recovery. For either method, there appears to be essentially no difference (to plotting accuracy) in the results due to grid density, except near the airfoil trailing edge (TE). When comparing the results of the two methods with each other, there is close agreement with only small differences occurring close to the TE. The surface skin-friction coefficient (C_f) distributions are displayed in Figs. 14 – 16. For each method, there is essentially no effect due to grid density on the C_f variations, except near the airfoil leading edge (LE) and TE. At the LE, there is a smaller difference in the peak C_f value between successive grids with the SGM than the USGM. Again, the differences on the first four grids near the TE seem to be a consequence of how the TE value is computed with the USGM (i.e., differences arising because one scheme is cell centered and the other is cell vertex).

The separation point of the flow computed with the SGM is approximately 0.808 chords. As indicated in Tables 3 and 4 for matrix and Roe dissipation, respectively, the location of the separation point relative to chord length (x_{sep}/c) remains almost the same starting with the 256×128 cell grid. These tables also give the aerodynamic coefficients: C_L is the lift coefficient, C_D is the total drag coefficient, $(C_D)_p$ is the pressure drag coefficient, and $(C_D)_{sf}$ is the skin-friction drag coefficient. All coefficients are based on surface integration using the trapezoidal rule. The C_L for Case 1 is approximately zero. Comparing the C_D from the SGM and USGM computations, we find that the difference is less than one drag count (less than 0.0001).

Figure 17 shows the Mach contours and overall streamline pattern for Case 1 obtained from the computation with SGM on the 1024×512 cell grid. It should be pointed out for clarification that the line emanating from the TE of the airfoil is simply a reference wake centerline, which will also appear in subsequent streamline plots. The details of the streamline pattern of the flow recirculation region are revealed in Fig. 18. The streamlines for the 4096×2048 grid are displayed in Fig. 19. Here we observe the longitudinal extent of the separation region. The closure of the symmetric separation region is as far downstream of the TE as the separation point is upstream of the TE.

Previously, we mentioned that results described as unsteady solutions (see for example ADIGMA Project [10]) have been computed. Here we show that with a weak solver a result can be obtained for Case 1 ($\alpha = 0^\circ$) that suggests that the solution may be unsteady. A steady-state calculation was performed with the USGM using multigrid acceleration with the LU-SGS smoother [33, 35]. A CFL = 5 was used. We should emphasize that the LU-SGS scheme is a member of a hiearchy of preconditioning techniques [15] one can use with the unstructured grid code; and thus, one can just choose the desired scheme. Figure 20 presents the convergence history and Mach contours for this computation. In the convergence history, we observe a residual reduction exceeding four orders of magnitude before it abruptly increases two orders at approximately 450 cycles, and then oscillates about that level. The Mach contours of the final result reveal what suggests a possible vortex shedding behavior. However, we showed in Section 5.2 machine zero convergence with both the SGM and USGM in Fig. 3 for this case. As already displayed, the Mach contours and streamlines corresponding to these computations reveal the dramatic difference between the steady-state solution and the result obtained with the LU-SGS scheme. This example not only identifies what can occur with a weak solution algorithm but also it emphasizes the importance of reducing the residual to machine zero. We should note that it is possible that with alternative weak solvers the residual could even be reduced to a much lower level before suddenly increasing, revealing what appears to be an unsteady solution.

In Figs. 21 – 36, we present the C_p and C_f distributions for Cases 2, 3 and 4. As indicated in Table 1, the α increases from one to three degrees for these cases. With the increasing α, we observe the expected increase of the suction peak in the C_p plots, along with the increasingly larger upper surface peak C_f value near the LE. Furthermore, the separation point on the upper surface of the airfoil moves toward the LE. When $\alpha = 3^\circ$, the boundary layer separates slightly before midchord location (0.464 chords) on the upper surface. The trends with respect to mesh density for both SGM and USGM are quite similar to those seen for Case 1. We also see similar trends to those discussed for Case 1 when comparing the solutions obtained with SGM and USGM. In addition, we have examined for Case 2 the effect on the C_f near the TE due to the magnitude of the numerical dissipation. Figure 26 shows that there is only a relatively small change in the C_f variation when the numerical dissipation is reduced locally by a factor of two. The changes in the aerodynamic coefficients with increasing mesh density for this set of cases are given in Table 3. On the finest mesh, the C_L computed with SGM for Cases 2, 3 and 4 increases from about 0.01847 to 0.05133, and the C_D increases only a small amount (from 0.05590 to 0.05851). These results are close to those obtained with USGM, as seen in Table 5.

The global behavior of the $\alpha = 3^\circ$ flow solution computed with SGM is displayed in Fig. 37. The Mach contours indicate the accelerated flow regions, and the streamline pattern delineates the large separated flow region. From the streamline patterns shown in Fig. 38 and Fig. 39, we can observe both primary and secondary recirculating regions, with the secondary region attached to the TE. This flow pattern is quite similar to the one for laminar flow over a NACA 64A015 airfoil at $\alpha = 5^\circ$ presented in Van Dyke's album [36] of fluid motion. For both airfoils, the flow separates near the 50% chord location on the upper surface, and it leaves tangentially to the lower surface.

5.2.2 Possible sources of differences in solutions

Small differences in solutions computed with different numerical algorithms can easily occur, even if precisely the same discretization methods and boundary conditions are being considered. Clearly, one of the principal reasons for this is that the implementation of such critical factors as approximations of the spatial derivatives in the governing equations and the boundary conditions (both physical and numerical boundary conditions) are not the same. In addition, there are often implementational changes in the numerical dissipation (e.g., regarding such elements as entropy fixes) that are introduced into the complete discrete formulations. It should also be recognized that variations in the solutions computed with different numerical schemes, especially when comparing aerodynamic coefficients, have frequently occurred in the literature and various computational workshops [37, 38].

This is not surprising regarding the aerodynamic coefficients since they depend on what method was used for numerical integration. In the subsequent discussion, we consider the specific prominent factors possibly affecting the relatively small differences that occur in results for the laminar flow simulations.

One possible contributing factor to the variations in the current solutions is the difference in the metrics, which is a consequence of the differences between cell centered and nodal point finite volume formulations. It is evident from [17, 22] and [19] that there are important dissimilarities in the discrete representation of the operators for the numerical dissipation (e.g., number of points in second and fourth difference approximations are not the same). With regards to the boundary conditions, there are clearly implementation disparities, which are indicated in Sections 3.2 and 4.3, that can possibly affect the relatively small observed differences in the solutions. As will be discussed in a later section, the skin-friction coefficient is computed in a different way for the structured and unstructured methods. Note, if we assume consistency of discretization for both numerical schemes, then the discretization variations should vanish with mesh refinement. At this point, based on the examination of the aerodynamic data, one cannot infer grid convergence, which as shown in [39] is very difficult to attain. Nevertheless, and even though it is not the focus of this paper, a careful further study of the underlying causes of the differences in the solutions, with special attention to the implementation in the computer codes, would certainly be beneficial.

5.2.3 Effect of outer boundary location

In defining any discrete external flow problem with the outer boundary of the solution domain located a finite distance from the geometry being considered, it is important to indicate the effect of boundary location on the flow solution. In Table 6, we examine the effects of the outer boundary location and far-field vortex (FFV) on the computed aerodynamic coefficients C_L and C_D. The results are for Case 4, since it has the highest angle of attack (α) that we have considered at $Re = 5000$. We consider the outer boundary at 5 chords, 10 chords, and 20 chords as well as the influence of the far-field vortex that represents the induced velocity effect. For each boundary location, we approximately maintain the same 10 chord grid to generate the 5 chord and 20 chord grids. Moreover, for the 20 chord grid, additional solution points are created in both coordinate directions so as to ensure a very small variation in the size of the mesh cells from 10 chords to 20 chords. The stretching factor SF is only about 1.01. Comparing the C_L and C_D values with the far-field vortex activated, we find that the variations over all boundary locations are 1.2% and 0.34%, respectively. The percent differences for C_L and C_D between the 10 chord and 20 chord locations are only 0.14% and 0.06%, respectively. Although not shown, a comparison of the surface C_p and C_f distributions for the three boundary locations and the FFV effect exhibits no discernible difference to plotting accuracy. These results are not surprising considering the subcritical flow and the activation of the FFV effect, as seen in [27].

A comparison is now made of the C_L and C_D values without and with the FFV effect. Table 6 indicates the expected continuous decrease in the differences of the corresponding aerodynamic coefficients as the location of the outer boundary increases from 5 chords to 20 chords. From the table, we see that at 10 chords there is only a 0.53% increase in C_L and a 0.03% increase in C_D due to the FFV. Furthermore, over the range of boundary locations the C_L percent difference decreases from 1.11% to 0.27%, and the C_D difference decreases from 0.06% to 0.015%. Thus, there is a linear variation of the diminishing percent differences in the aerodynamic coefficients due to the FFV effect as the outer boundary location increases. The much smaller influence of the FFV on C_D is expected, based on the FFV paper by Thomas and Salas [27].

Here we also consider the FFV effect on the computed aerodynamic coefficients of Case 5 in which there is transonic flow and a higher angle of attack ($\alpha = 10^\circ$). Table 7 shows results without and with the FFV when the outer boundary is located 20 chords away from the airfoil. There are somewhat larger changes in the C_L and C_D due to the physics of the flow. The increases in C_L and C_D due to the FFV effect are 1.11% and 0.62%, respectively. Nevertheless, the changes in the

aerodynamic coefficients are relatively small, and although not shown, the topology of the flow (just as in Case 4) is not significantly affected.

5.2.4 Effect of skin-friction methods

Another consideration that should be addressed is the method of computing the skin-friction drag, which as indicated previously is denoted by $(C_D)_{sf}$. For the structured and unstructured grid computations of skin friction, we used the stress tensor and vorticity methods, respectively. Since the methods are different, a comparison is now made between these methods and a third one that is designated as the tangential velocity gradient method. These three approaches can be succinctly described as follows. Method 1 is the stress tensor method, which is based on the inner product of the stress tensor and the unit surface tangent vector $\mathbf{e_t}$. Method 2 is the vorticity vector method, and it uses the inner product of the vorticity vector and $\mathbf{e_t}$. Method 3 is the tangential velocity gradient method, which uses the surface normal gradient of the tangential velocity. In Table 8, we compare the effect of these different methods on the $(C_D)_{sf}$ of Case 1, using the 1024×512 grid. This table shows that there are very small differences in the $(C_D)_{sf}$ computed with the three methods. Moreover, the maximum percent difference relative to Method 1 on even the coarsest grids is less than 0.01%. The results for Method 2 and Method 3 are the same to 5 significant digits for all grid densities greater than 128×64.

5.2.5 Grid convergence of aerodynamic coefficients

As examples of the grid convergence of the aerodynamic coefficients, we now consider in more detail Cases 1 and 4, which are the original laminar flow cases proposed by Swanson in 1984 and introduced in [1]. The variation in the computed total drag C_D with mesh refinement for Case 1 (nonlifting) is shown in Fig. 40. Results for the SGM include both the matrix and Roe forms of numerical dissipation. A comparison of differences in C_D on each of the first four grids (128×64 – 1024×512) between the SGM and USGM reveals a decrease from 1.2×10^{-3} to 5.1×10^{-5}. Moreover, if we consider the C_D results relative to the solution on the finest grid (4096×2048), we find that the difference for the pressure contribution $(C_D)_p$ as well as the difference for the skin-friction contribution $(C_D)_{sf}$ is less than 0.2%. The $(C_D)_p$ difference is the dominant part of the total C_D difference. It is important to note that the C_D with SGM is still increasing; and thus, there is an expectation that the C_D values will essentially match on the finest grid.

The grid convergence behavior of the C_L and C_D for Case 4 (having the highest α of the $Re = 5000$ cases) is displayed in Fig. 41. The percent differences in the C_L and C_D between the SGM and USGM computations (using matrix dissipation) are less than 0.5% and 0.2%, respectively. From this comparison of the computed C_L with the SGM and USGM, we observe a lower C_L with the SGM and an unexpected sudden change in the trend from increasing to decreasing occurring between the 512×256 and 1024×512 grids, while the C_L with USGM is continuously slowly decreasing for the full range of grids while maintaining a higher value. At this point, we cannot dismiss the possibility that the sudden change in trend for the C_L with SGM is simply a small oscillation and the apparent trend will reverse itself on the finer meshes. Note, even though the far-field vortex (FFV) effect is not used with the USGM but is used with SGM, this is not expected to decrease the difference in C_L, but rather to increase it slightly, as pointed out in [27]. Consideration of the information available does not clearly indicate which or what combination of the possible differences in the numerical methods, as discussed in Section 5.2.2, cause the differences in the C_L. As evident in Fig. 41b, the behavior of C_D with mesh refinement is quite similar to that for non-lifting case. As we have already seen, and we should emphasize here, the topology of the flow determined with the two methods is essentially the same.

In this paper, we have not discussed the order property as established by the asymptotic convergence with mesh refinement of the numerical schemes. Nevertheless, even though this is clearly not the focus of this paper, some succinct remarks on this topic would be appropriate. While the

design order of accuracy of a scheme is determined by the numerical discretization, which requires sufficiently fine meshes to verify, the asymptotic order depends on local reduction of approximation order (e.g., at the LE of an airfoil or at a shock wave) and singularities in the solution field. It is difficult in general to demonstrate the asymptotic convergence order for a given discrete problem. This difficulty can be exacerbated when there is a structural singularity such as the one at the TE of the airfoil being considered. For example, see Diskin et al. [39] and Hu et al. [40]. Note that due to the TE singularity and the elliptic nature of the flow near the airfoil surface, the asymptotic behavior of the solution error with mesh refinement would be expected to exhibit at best first order accuracy. Additional information on a corner (or reentrant) flow singularity can be found in Refs. [39, 41]. There is no evidence in this work that the asymptotic convergence regime has been attained. We should also mention that even with incredibly fine grid spacing near the TE using an adaptive method [40], it is difficult to determine the necessary grid spacing near the TE to establish the singularity behavior. While the family of grids used for this paper is quite reasonable with respect to the objectives, it is evident that this family is not optimal from the standpoint of an order property study. That is, the coarser grids would have to be much finer in the LE and TE regions, where the dominant errors occur [42], to allow an early onset of the design order of the scheme.

5.3 Transonic flow (Case 5)

The final laminar flow case comes from the GAMM workshop [43] on compressible Navier-Stokes solutions held in Nice, France during December of 1985. We designate this flow over the NACA 0012 airfoil as Case 5. This laminar flow is also considered in this paper because it is frequently used as a test problem. This problem was selected for the workshop because the solution is expected to be steady, the flow is transonic, and there is flow separation. For this case, the Re is reduced to 500 and the angle of attack is increased to 10°. With a higher α and a free-stream Mach number of 0.8, the flow becomes transonic on the forward portion of the airfoil upper surface. Furthermore, even though the boundary layer is quite thick, there is sufficient flow acceleration to create a small supersonic zone but not a shock wave.

All the computational results presented in [43] indicate steady solutions. The highest mesh density considered was 256×64 cells. While similar results were obtained for the C_p distributions by the workshop participants, there were significant differences in the surface C_f variations. In the present calculations, we have increased the mesh density to 1280×512, which is a factor of 40 times larger than the highest value used in the workshop.

The surface pressure and skin-friction distributions computed with the structured and unstructured algorithms are presented in Figs. 42–45. In general, there is good agreement between the two sets of results, with the only discernible differences occurring at the airfoil LE and TE. As evident in the blowup of the LE region skin-friction results in Fig. 46, there are significant differences on the coarser grids. For this case, there is considerable flow separation, which begins at about 0.36 chords and extends over the entire airfoil. The supersonic zone of the flow is revealed in the Mach contours (computed with the structured scheme) that are displayed in Fig. 47. Streamline patterns near the airfoil TE are shown in Figs. 48 and 49. Here, we observe two recirculation regions resembling those seen for Case 4. However, there is a much larger lateral extent of the recirculation region, both on the airfoil and in the wake.

A comparison of the aerodynamic coefficients given in Tables 3 – 5 reveals a very close agreement between the computed values with SGM and USGM. As for Case 4, the percent difference in the C_L (again relative to the finest grid value) is less than 0.5%. There is a very small percent difference in the C_D, which is less than 0.05%. It is clear that these results, as well as some of those from the workshop, indicate that reasonably good estimates of the C_L and C_D can be achieved on fairly coarse grids. For example, with the SGM using matrix dissipation, the percent differences in the computed C_L and C_D on the 80×32 grid and the finest grid are less than 4.0% and 2.6%, respectively.

Even though reasonably good estimates of the C_L and C_D can be achieved on fairly coarse

grids, we should emphasize that there is a need to have appropriate resolution in the neighborhood of the separation point to obtain the steady flow solution. Furthermore, the current results such as the streamline plots demonstrate the importance of having sufficient resolution in both coordinate directions to capture the extent of separation as well as the two vortices that occur.

6 Concluding Remarks

A family of laminar subsonic ($M = 0.5$) airfoil flows with Reynolds number $Re = 5000$ and angle of attack (α) variation from zero to three degrees, along with a transonic case at $Re = 500$ and $\alpha = 10^\circ$, have been considered. The purpose of such flow problems is to provide a set of test cases for evaluating accuracy and solution algorithms of a given computational method for solving the compressible Navier-Stokes (N-S) equations. These cases provide a challenge for a computational method, since they have flow separation regions of varying sizes that begin on the airfoil and extend into the airfoil wake, where they terminate. There must be appropriate resolution of the flow solution on the airfoil and in the wake. Thus, the characteristics of this class of flows provides evaluation of the discretization of the N-S equations with particular emphasis on the physical diffusion effects.

A concern relating to this class of laminar flow problems has emerged recently due to the occurrence of solutions that are not consistent with a large number of previous solutions. These recent results include an asymmetric solution for a nonlifting NACA 0012 airfoil and an unsteady solution. In this paper, we have given several possible reasons why such solutions have been generated. One of these reasons is inadequate spatial resolution, especially in the neighborhood of the flow separation point and possibly in the vicinity of closure of the recirculation region in the wake of the airfoil. Without appropriate resolution, stationary flows may give the appearance of an unsteady flow. Another of these reasons, which pertains to Case 1 ($\alpha = 0^\circ$), is the presence of an asymmetry in the numerical method, which may actually prevent convergence or create what appears to be an unsteady flow solution. A third reason, which is discussed subsequently, is that convergence has not been achieved. In fact, the residual of the discrete equations may have only been reduced two or three orders of magnitude.

It should be pointed out that weak solution schemes may experience numerical difficulties and not achieve steady-state solutions for increasing mesh densities. An indicator for this assertion is given in [34]. There it was demonstrated using spectral analysis for Case 2 on the 256×128 mesh that weak solution schemes are not stable in general, and hence, cannot always converge to the steady-state solution. Such schemes have been briefly discussed in this paper. Moreover, we have shown that the computation of Case 1 with the LU-SGS scheme [33, 35] can produce what appears to be an unsteady solution, while computation with a stronger solver gives the symmetric, steady solution.

In this paper, numerical solutions have been obtained for the aforementioned family of laminar flows on a range of grid densities, with the highest being nearly 8.4 million cells. Both structured and unstructured grid algorithms have been used to calculate the solutions. Comparisons have been made of the computed aerodynamic coefficients computed with the two algorithms. The computations support the assertion that steady-state solutions exist and that there is a good agreement between the solutions determined with the two methods. Moreover, unsteady flow computations have also been performed obtaining the same steady-state solutions, confirming the existence of steady-state solutions for these laminar flows. This indicates that it is appropriate to use these flows for test cases of steady-state solvers for solving the N-S equations.

In comparing the solutions computed with the structured grid and unstructured grid methods, we have observed that the primary differences generally occur at the LE and TE of the airfoil. This is not unexpected since these are the locations where the dominant errors occur, which is revealed by the eigenvectors associated with the error [42]. In addition, at the airfoil TE, variations between the solutions would be expected due to the cell centered and nodal point finite volume schemes of the two methods. With mesh refinement, comparisons of the aerodynamic coefficients calculated with

the two methods have revealed relatively small differences. For example, for Case 4 ($\alpha = 3^\circ$), the percent differences for C_L and C_D are less than 0.5% and 0.2%, respectively. Possible sources of these differences have been identified. These sources include the type of finite volume formulation and implementations of both the numerical dissipation and boundary conditions. In addition, there are differences in how the aerodynamic coefficients were computed (e.g., method of computing the surface skin friction). Although there exist some relatively small differences in a computed solution with the two methods, we have demonstrated that the topology of the flow is essentially the same.

A question that arises is what happens to these laminar flow solutions in the limit of infinitely fine mesh. Certainly, in this limit, one cannot totally dismiss the possibility of quasi-steady solutions, which correspond to a weak unsteadiness producing extremely small variations in the aerodynamic coefficients while essentially maintaining the same flow structure. However, even if this were true, there is still a high level of functionality of these laminar flow cases consistent with the original intended purpose of evaluating accuracy, stability, and efficiency of numerical schemes for solving the compressible Navier-Stokes equations.

Table 1: Conditions for laminar flow over the NACA 0012 airfoil.

Cases	M_∞	α (deg.)	Re_c
Case 1	0.5	0.0	5×10^3
Case 2	0.5	1.0	5×10^3
Case 3	0.5	2.0	5×10^3
Case 4	0.5	3.0	5×10^3
Case 5	0.8	10.0	5×10^2

Table 2: Descriptive parameters of grids for laminar flow computations. The parameters are: minimum normal grid spacing (Δy_{min}), tangential spacing at airfoil leading edge (Δs_{le}), x-coordinate grid spacing at the mid-chord location ($\Delta x_{x=0.5c}$), x-coordinate spacing at airfoil trailing edge (Δx_{te}), and the grid stretching factor for the normal direction (SF).

Grid	Δy_{min}	Δs_{le}	$\Delta x_{x=0.5c}$	Δx_{te}	SF
1024×512	1.266×10^{-4}	1.238×10^{-3}	9.198×10^{-3}	1.230×10^{-3}	1.014
4096×2048	3.119×10^{-5}	3.076×10^{-4}	2.309×10^{-3}	3.087×10^{-4}	1.003
1280×512	5.249×10^{-4}	5.027×10^{-3}	7.732×10^{-3}	4.995×10^{-3}	1.012

Table 3: Computed lift and drag coefficients for laminar airfoil flow cases. Structured grid RK/Implicit scheme with matrix dissipation.

Case	Grid	C_L	C_D	$(C_D)_p$	$(C_D)_{sf}$	x_{sep}/c
1	128×64	9.73932×10^{-12}	0.0560794	0.0237143	0.0323651	0.809236
1	256×128	6.30245×10^{-12}	0.0554656	0.0227134	0.0327523	0.809147
1	512×256	2.51916×10^{-12}	0.0555269	0.0227424	0.0327845	0.808484
1	1024×512	6.96415×10^{-13}	0.0555743	0.0227887	0.0327855	0.808324
2	128×64	0.0171399	0.0563769	0.0240700	0.0323070	0.673162
2	256×128	0.0182808	0.0557857	0.0230892	0.0326965	0.677438
2	512×256	0.0184589	0.0558544	0.0231239	0.0327304	0.677719
2	1024×512	0.0184668	0.0559045	0.0231720	0.0327325	0.677768
3	128×64	0.0340812	0.0572544	0.0251267	0.0321277	0.551588
3	256×128	0.0356422	0.0567422	0.0242105	0.0325319	0.557332
3	512×256	0.0357575	0.0568367	0.0242637	0.0325575	0.561063
3	1024×512	0.0358071	0.0568914	0.0243173	0.0325741	0.561379
4	128×64	0.0506531	0.0586205	0.0268267	0.0317938	0.469026
4	256×128	0.0513175	0.0583079	0.0260407	0.0322673	0.468541
4	512×256	0.0513379	0.0584381	0.0261335	0.0323046	0.466286
4	1024×512	0.0513279	0.0585102	0.0261983	0.0323118	0.463885
5	80×32	0.420795	0.268071	0.146177	0.121894	0.386173
5	160×64	0.428373	0.272255	0.145338	0.125916	0.366441
5	320×128	0.433578	0.274268	0.147083	0.127185	0.363095
5	640×256	0.435635	0.274952	0.147426	0.127526	0.362257
5	1280×512	0.436331	0.275155	0.147544	0.127611	0.362180

Table 4: Computed lift and drag coefficients for laminar airfoil flow cases. Structured grid RK/Implicit scheme with Roe dissipation.

Case	Grid	C_L	C_D	$(C_D)_p$	$(C_D)_{sf}$	x_{sep}/c
1	128×64	4.30190×10^{-12}	0.0603439	0.0286745	0.0316694	0.797444
1	256×128	2.82493×10^{-12}	0.0560682	0.0233966	0.0326716	0.808790
1	512×256	1.46832×10^{-12}	0.0555955	0.0228201	0.0327754	0.809361
1	1024×512	6.14893×10^{-13}	0.0555786	0.0227945	0.0327841	0.809314
2	128×64	0.0142639	0.0606418	0.0289888	0.0316530	0.641873
2	256×128	0.0178031	0.0563917	0.0237691	0.0326226	0.675356
2	512×256	0.0183782	0.0559240	0.0232022	0.0327119	0.677467
2	1024×512	0.0184557	0.0559088	0.0231778	0.0327310	0.677736
3	128×64	0.0298003	0.0616101	0.0300498	0.0315604	0.545074
3	256×128	0.0349084	0.0573717	0.0249030	0.0324688	0.552189
3	512×256	0.0356412	0.0569117	0.0243459	0.0325657	0.560561
3	1024×512	0.0357909	0.0568967	0.0243240	0.0325727	0.561712
4	128×64	0.0458620	0.0633387	0.0320464	0.0312923	0.471476
4	256×128	0.0505787	0.0590025	0.0267893	0.0322132	0.470231
4	512×256	0.0512355	0.0585229	0.0262236	0.0322933	0.466906
4	1024×512	0.0513153	0.0585175	0.0262065	0.0323110	0.464062
5	80×32	0.473774	0.288645	0.162503	0.126142	0.399557
5	160×64	0.443147	0.278322	0.150869	0.127453	0.371976
5	320×128	0.436730	0.275698	0.148106	0.127593	0.363705
5	640×256	0.436133	0.275181	0.147588	0.127593	0.362256
5	1280×512	0.436328	0.275108	0.147521	0.127587	0.362006

Table 5: Computed lift and drag coefficients for laminar airfoil flow cases. Unstructured scheme for Green-Gauss (GG) approximation and matrix dissipation.

Case	Grid	C_L	C_D	$(C_D)_p$	$(C_D)_{sf}$	x_{sep}/c
1	128×64	5.72532×10^{-12}	0.0548571	0.0220249	0.0328322	0.822568
1	256×128	3.66707×10^{-12}	0.0553396	0.0224786	0.3286100	0.812725
1	512×256	1.43241×10^{-12}	0.0555549	0.0227244	0.0328305	0.809610
1	1024×512	3.08198×10^{-13}	0.0556251	0.0228086	0.0328165	0.808710
1	2048×1024	3.26406×10^{-14}	0.0556450	0.0228340	0.0328109	0.808450
1	4096×2048	1.21639×10^{-13}	0.0556505	0.0228396	0.0328108	0.808369
2	128×64	0.0184311	0.0549979	0.0221796	0.0328183	0.679040
2	256×128	0.0188355	0.0556344	0.0228022	0.0328321	0.678415
2	512×256	0.0186424	0.0558790	0.0230895	0.0327895	0.677853
2	1024×512	0.0185413	0.0559542	0.0231846	0.0327696	0.677682
2	2048×1024	0.0184852	0.0559747	0.0232172	0.0327575	0.677712
2	4096×2048	0.0184635	0.0559804	0.0232227	0.0327577	0.677611
3	128×64	0.0372565	0.0560317	0.0234804	0.0325513	0.562096
3	256×128	0.0365579	0.0566088	0.0239846	0.0326242	0.562351
3	512×256	0.0361443	0.0568585	0.0242485	0.0326099	0.562385
3	1024×512	0.0359898	0.0569381	0.0243367	0.0326014	0.562463
3	2048×1024	0.0359160	0.0569602	0.0243622	0.0325980	0.562382
3	4096×2048	0.0358777	0.0569659	0.0243692	0.0325967	0.562348
4	128×64	0.0545145	0.0574118	0.0252292	0.0321826	0.475247
4	256×128	0.0526257	0.0581509	0.0258241	0.0323269	0.470011
4	512×256	0.0519099	0.0584556	0.0261207	0.0323349	0.466490
4	1024×512	0.0516846	0.0585508	0.0262172	0.0323334	0.464571
4	2048×1024	0.0515882	0.0585768	0.0262443	0.0323335	0.464276
4	4096×2048	0.0515404	0.0585835	0.0262512	0.0323323	0.464192
5	80×32	0.229649	0.237335	0.116117	0.121219	0.371130
5	160×64	0.423565	0.271545	0.146140	0.125405	0.370509
5	320×128	0.433892	0.274046	0.146860	0.127185	0.367340
5	640×256	0.437006	0.274879	0.147233	0.127646	0.366329
5	1280×512	0.437998	0.275166	0.147402	0.127764	0.366083

Table 6: Effects of outer boundary location and far-field vortex on computed aerodynamic coefficients. Results are for laminar airfoil flow Case 4 ($\alpha = 3^o$). Structured grid RK/Implicit scheme with matrix dissipation. Outer Bdry means approximate outer boundary location in airfoil chord lengths, and FF Vortex means far-field vortex boundary condition.

Outer Bdry	FF Vortex	Grid	C_L	C_D	$(C_D)_p$	$(C_D)_{sf}$
5c	off	944×456	0.0512920	0.0586401	0.0262542	0.0323859
5c	on	944×456	0.0518624	0.0586766	0.0263024	0.0323742
10c	off	1024×512	0.0510556	0.0584924	0.0261749	0.0323174
10c	on	1024×512	0.0513279	0.0585102	0.0261983	0.0323118
20c	off	1096×560	0.0511207	0.0584662	0.0261671	0.0322991
20c	on	1096×560	0.0512576	0.0584751	0.0261789	0.0322962

Table 7: Effect of far-field vortex on computed aerodynamic coefficients for laminar airfoil flow Case 5 ($M = 0.8$, $\alpha = 10^o$, $Re = 500$). Structured grid RK/Implicit scheme with matrix dissipation.

Outer Bdry	FF Vortex	Grid	C_L	C_D	$(C_D)_p$	$(C_D)_{sf}$
20c	off	1280×512	0.432698	0.274846	0.147061	0.127784
20c	on	1280×512	0.437520	0.276531	0.147890	0.128641

Table 8: Computed lift and drag coefficients for laminar airfoil flow Case 1. Effect of skin friction computed with different methods (Skin-friction computed with Method 1: Stress tensor; Method 2: Vorticity; Method 3: Tangential gradient). Note, the separation location denoted by x_{sep}/c is determined by linear interpolation for zero skin-friction. Computations with structured grid RK/Implicit scheme using matrix dissipation.

Method	Grid	C_L	C_D	$(C_D)_p$	$(C_D)_{sf}$	x_{sep}/c
1	128×64	9.73932×10^{-12}	0.0560794	0.0237143	0.0323651	0.809236
2	128×64	9.73885×10^{-12}	0.0560816	0.0237143	0.0323673	0.808446
3	128×64	9.73886×10^{-12}	0.0560802	0.0237143	0.0323659	0.808446
1	256×128	6.30245×10^{-12}	0.0554656	0.0227134	0.0327523	0.809147
2	256×128	6.30228×10^{-12}	0.0554678	0.0227134	0.0327544	0.810003
3	256×128	6.30225×10^{-12}	0.0554673	0.0227134	0.0327540	0.810003
1	512×256	2.51916×10^{-12}	0.0555269	0.0227424	0.0327845	0.808484
2	512×256	2.51912×10^{-12}	0.0555288	0.0227424	0.0327863	0.809498
3	512×256	2.51912×10^{-12}	0.0555284	0.0227424	0.0327860	0.809497
1	1024×512	6.96415×10^{-13}	0.0555743	0.0227887	0.0327855	0.808324
2	1024×512	6.96410×10^{-13}	0.0555758	0.0227887	0.0327871	0.809330
3	1024×512	6.96421×10^{-13}	0.0555755	0.0227887	0.0327868	0.809330

References

1. R. C. Swanson and E. Turkel. A multistage time-stepping scheme for the Navier-Stokes equations. *AIAA Paper 85-0035*, 1985.

2. L. Martinelli, A. Jameson, and F. Grasso. A multigrid method for the Navier-Stokes equations. *AIAA Paper 86-0208*, 1986.

3. R. C. Swanson and E. Turkel. Artificial dissipation and central difference schemes for the Euler and Navier-Stokes equations. *AIAA Paper 87-1107*, 1987.

4. V. Venkatakrishnan. Newton solution of inviscid and viscous problems. *AIAA Paper 88-0413*, 1988.

5. M. Jayaram and A. Jameson. Multigrid solution of the Navier-Stokes equations for flow over wings. *AIAA Paper 88-0705*, 1988.

6. D. J. Mavriplis and A. Jameson. Multigrid solution of the Navier-Stokes equations on triangular meshes. *AIAA J.*, 28(8):1415–1425, 1990.

7. T. L. Tysinger and D. A. Caughey. Implicit multigrid algorithm for the Navier-Stokes equations. *AIAA Paper 91-0242*, 1991.

8. M. Svärd, J. Lundberg, and J. Nordström. A computational study of vortex-airfoil interaction using high-order finite difference methods. *Computers & Fluids*, 39(8):1267–1274, 2010.

9. D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *J. Comput. Phys.*, 187(1):22–46, May 2003.

10. N. Kroll. The ADIGMA project. In N. Kroll, H. Bieler, and Vol. 113 H. Deconinck, editors, *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Results of a Collaborative Research Project Funded by the European Union, 2006-2009*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Volume 113. Springer Berlin Heidelberg, 2010.

11. R. Abgrall and D. De Santis. High order preserving residual distribution schemes for the laminar and turbulent Navier-Stokes on arbitrary grids. *AIAA Paper 2013-2562*, 2013.

12. V. Dolejší. A design of residual error estimates for a high order bdf-dgfe method applied to compressible flows. *Int. J. Numer. Meth. Fluids*, 73:523–559, 2013.

13. C.-C. Rossow. Efficient computation of compressible and incompressible flows. *J. Comput. Phys.*, 220:879–899, 2007.

14. R. C. Swanson, E. Turkel, and C.-C. Rossow. Convergence acceleration of Runge-Kutta schemes for solving the Navier-Stokes equations. *J. Comput. Phys.*, 224(1):365–388, May 2007.

15. S. Langer. Hierarchy of preconditioning techniques for the solution of the Navier-Stokes equations discretized by 2nd order unstructured finite volume methods. In *ECCOMAS 2012 CONFERENCE*, Conference Proceeding Series, September 2012.

16. H. Schlichting. *Boundary Layer Theory*. McGraw-Hill, New York, 1979.

17. R. C. Swanson and E. Turkel. On central difference and upwind schemes. *J. Comput. Phys.*, 101:292–306, 1997.

18. P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

19. S. Langer. Agglomeration multigrid methods with implicit Runge-Kutta smoothers applied to aerodynamic simulations on unstructured grids. *J. Comput. Phys.*, 277(0):72 – 100, 2014.

20. A. Jameson. Solution of the Euler equations for two-dimensional, transonic flow by a multigrid method. *Appl. Math. Comput.*, 13:327–356, 1983.

21. C.-C. Rossow. Convergence acceleration for solving the compressible Navier-Stokes equations. *AIAA J.*, 44:345–352, 2006.

22. R. C. Swanson and E. Turkel. Multistage schemes with multigrid for Euler and Navier-Stokes equations. NASA Technical Paper 3631, NASA Langley Research Center, 1997.

23. T. W. Roberts and R. C. Swanson. A study of multigrid preconditioners using eigensystem analysis. *AIAA Paper 2005-5229*, 2005.

24. R. C. Swanson, E. Turkel, and S. Yaniv. Analysis of a RK/Implicit smoother for multigrid. In A. Kuzmin, editor, *Computational Fluid Dynamics 2010: Proceedings of the Sixth International Conference on Computational Fluid Dynamics, ICCFD6, St. Petersburg, Russia*, The International Conference on Computational Fluid Dynamics, pages 409–417. Springer-Verlag Berlin Heidelberg, 2011.

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

26. R. C. Swanson and C.-C. Rossow. An efficient solver for the RANS equations and a one-equation turbulence model. *Computers & Fluids*, 42:13–25, 2011.

27. J. L. Thomas and M. D. Salas. Far-field boundary conditions for transonic lifting solutions to the Euler equations. *AIAA J.*, 24:1074–1080, 1986.

28. E. Turkel. Improving the accuracy of central difference schemes. In D. L. Dwoyer and M. Y. Hussaini, editors, *11th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics*, volume 323. Springer-Verlag, 1989.

29. S. Langer. Investigation and application of point implicit Runge-Kutta methods to inviscid flow problems. *Int. J. Numer. Meth. Fluids*, 69(2):332–352, 2012.

30. Y. Saad. *Iterative Methods for Sparse Linear Systems*. International Thomson Publishing, Boston, 1996.

31. U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society of Industrial Applied Mathematics (SIAM), Philadelphia, 1998.

32. T. Gerhold, M. Galle, O. Friedrich, and J. Evans. Calculation of complex three-dimensional configurations employing the DLR-TAU-code. *AIAA Paper 97-0167*, 1997.

33. S. Langer, A. Schwöppe, and N. Kroll. The DLR flow solver TAU - status and recent algorithmic developments. In *Proc. 52nd Aerospace Sciences Meeting, January 2014*, number AIAA Paper 2014-0080 in Conference Proceeding Series, 2014.

34. S. Langer. Computer aided analysis of preconditioned multistage Runge-Kutta methods applied to solve the compressible Reynolds averaged Navier-Stokes equations. In H. P. Kreplin, editor, *New Results in Numerical and Experimental Fluid Mechanics X: Contributions to the 19th STAB/DGLR Symposium Munich, Germany, 2014*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, 2015. To appear.

35. S. Yoon and A. Jameson. Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations. *AIAA Paper 87-0600*, 1987.

36. M. Van Dyke, editor. *Album of Fluid Motion*. The Parabolic Press, Stanford, California, tenth edition, 1982.

37. D. W. Levy, T. Zickuhr, J. Vassburg, S. Agrawal, R. A. Wahls, S. Pirzadeh, and M. J. Hemsch. Summary of data from the First AIAA Drag Prediction Workshop. *AIAA Paper 2002-0841*, 2002.

38. D. W. Levy, K. R. Laflin, E. N. Tinoco, J. C. Vassberg, M. Mani, B. Rider, C. L. Rumsey, R. A. Wahls, J. H. Morrison, O. P. Brodersen, S. Crippa, D. J. Mavriplis, and M. Murayama. Summary of Data from the Fifth AIAA Drag Prediction Workshop. In *Proc. 51st AIAA Aerospace Sciences Meeting, Gravepine (Dallas/Ft. Worth Region), Texas, January 2013*, number 2013-0046 in Conference Proceeding Series. AIAA, 2013.

39. B. Diskin, J. L. Thomas, C. L. Rumsey, and A. Schwöppe. Grid convergence for turbulent flows. *AIAA Paper 2015-1746*, 2015.

40. Y. Hu, C. Wagner, S. R. Almaras, M. Galbraith, and D. L. Darmofal. Application of a higher-order adaptive method to rans test cases. *AIAA Paper 2015-1530*, 2015.

41. P. Laasonen. On the discretization error of the dirichlet problem in a plane region with corners. *Annales Academi Scientiarum Fennic Mathematica*, 408:1–16, 1967.

42. L. E. Eriksson and A. Rizzi. Computer-aided analysis of the convergence to steady state of discrete approximations to the euler equations. *J. Comput. Phys.*, 57(1):90 – 128, 1985.

43. M. O. Bristeau, R. Glowinski, J. Periaux, and H. Viviand. Presentation of problems and discussion of results. In M. O. Bristeau, R. Glowinski, J. Periaux, and H. Viviand, editors, *Numerical Simulation of Compressible Navier-Stokes Flows: A GAMM Workshop*, Notes on Numerical Fluid Mechanics, Vol. 18. F. Vieweg & Sohn, 1987.

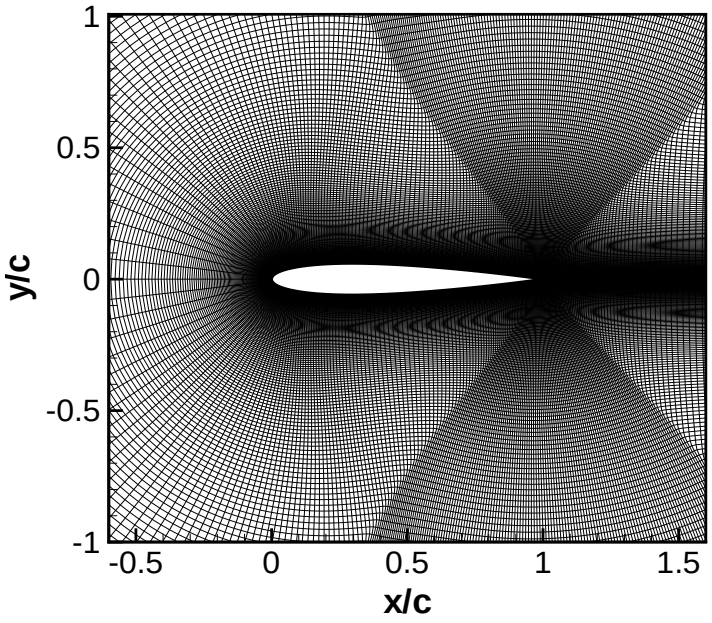
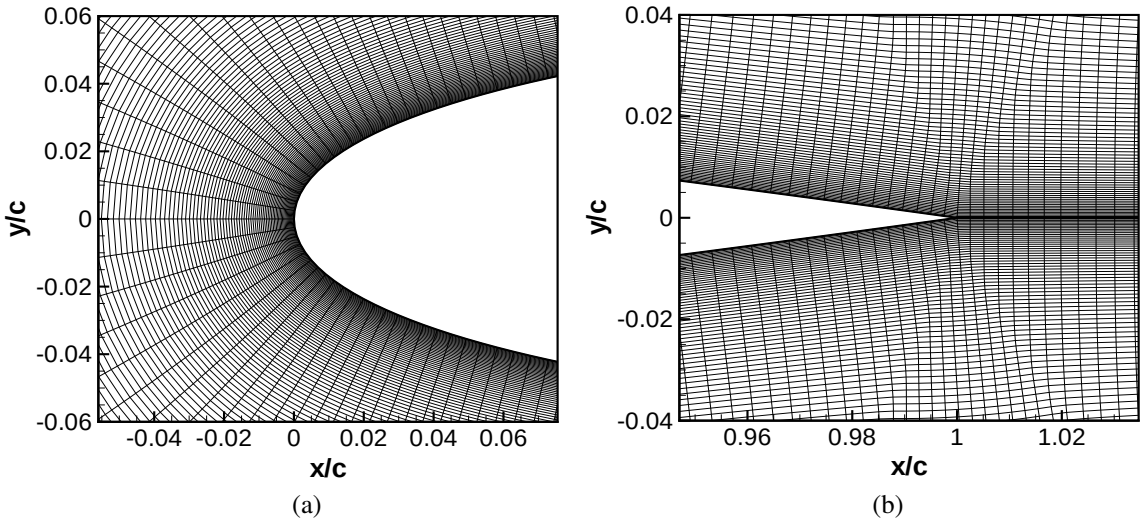
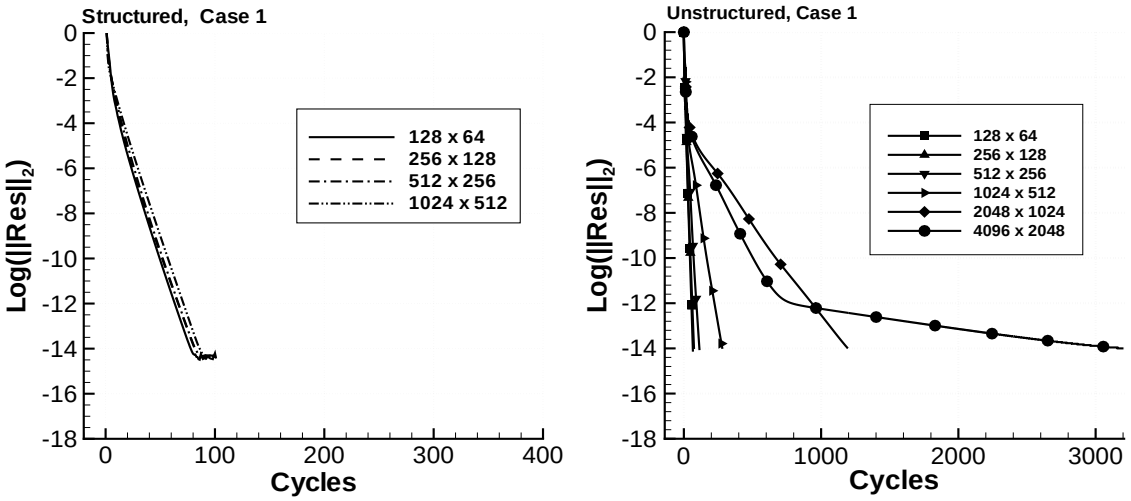


Figure 1: Near field view of 512×256 grid for laminar flow computations when $Re = 5000$.



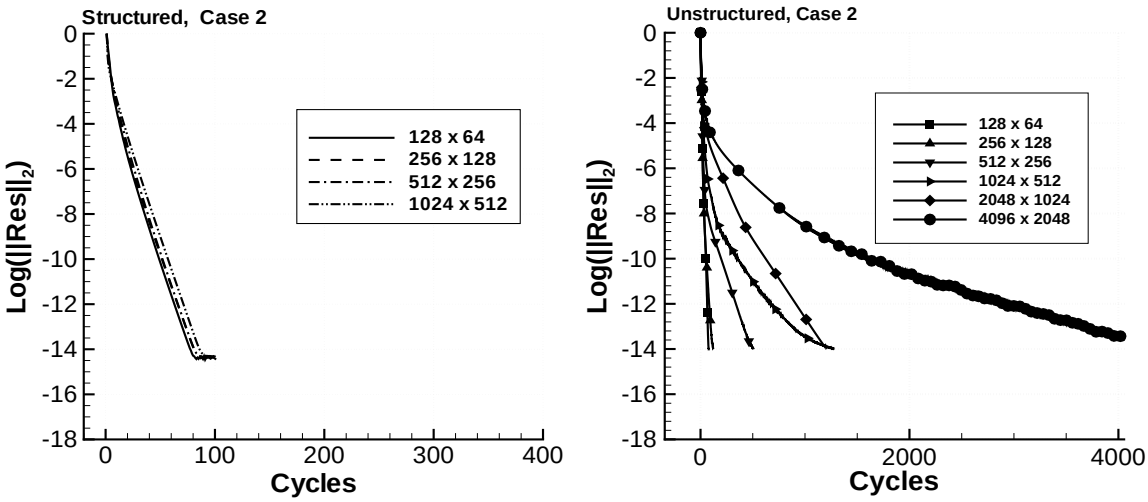
(a) (b)

Figure 2: Leading and trailing edges of 512×256 grid for laminar flow computations when $Re = 5000$. (a) Leading edge, (b) Trailing edge.



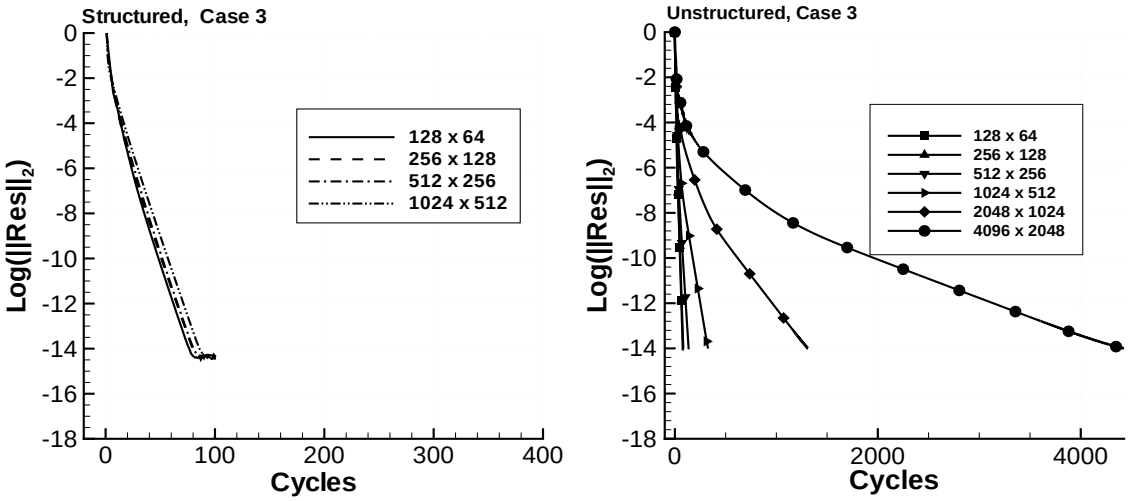
(a) (b)

Figure 3: Comparison of convergence histories for Case 1 computed with structured and unstructured grid methods. (a) Structured, (b) Unstructured.



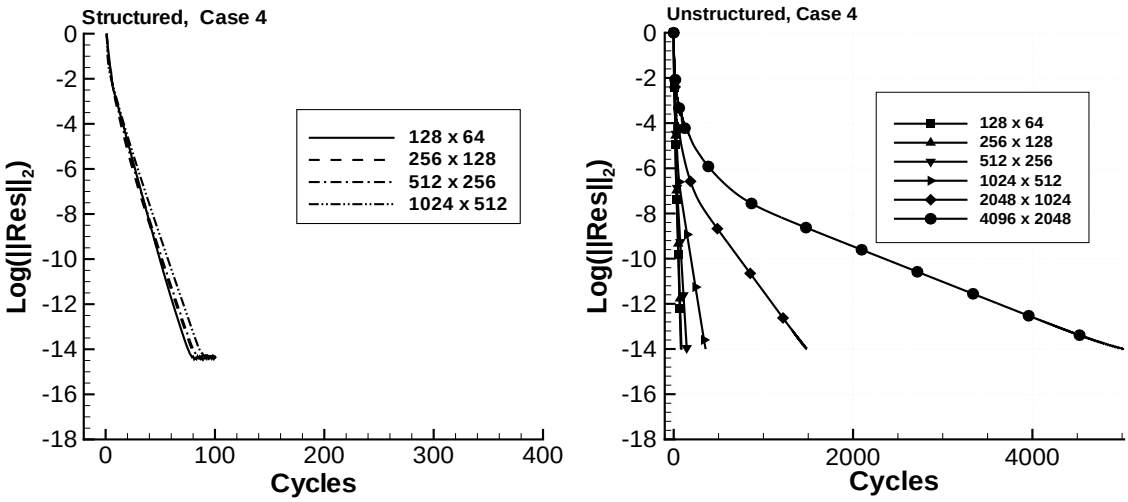
(a) (b)

Figure 4: Comparison of convergence histories for Case 2 computed with structured and unstructured grid methods. (a) Structured, (b) Unstructured.



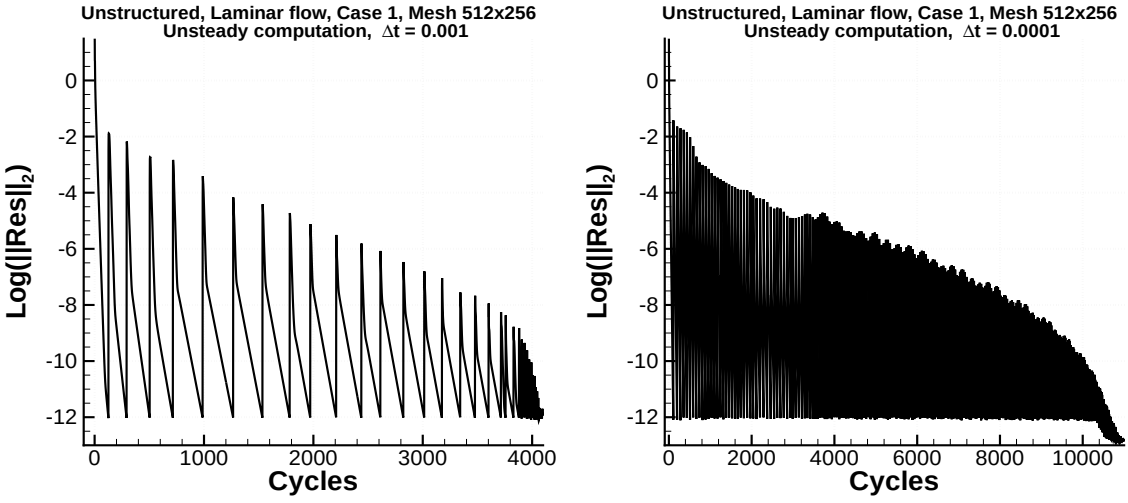
(a) (b)

Figure 5: Comparison of convergence histories for Case 3 computed with structured and unstructured grid methods. (a) Structured, (b) Unstructured.



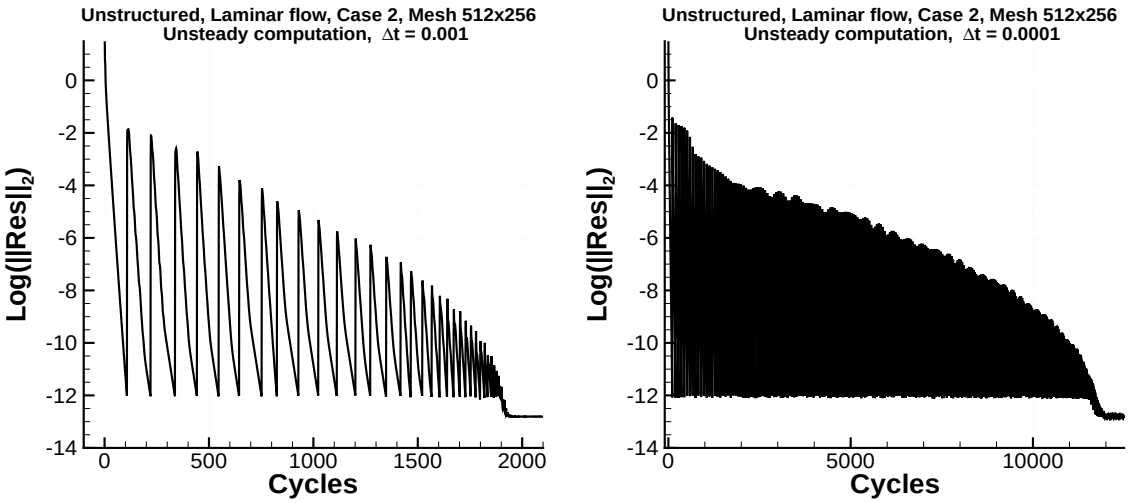
(a) (b)

Figure 6: Comparison of convergence histories for Case 4 computed with structured and unstructured grid methods. (a) Structured, (b) Unstructured.



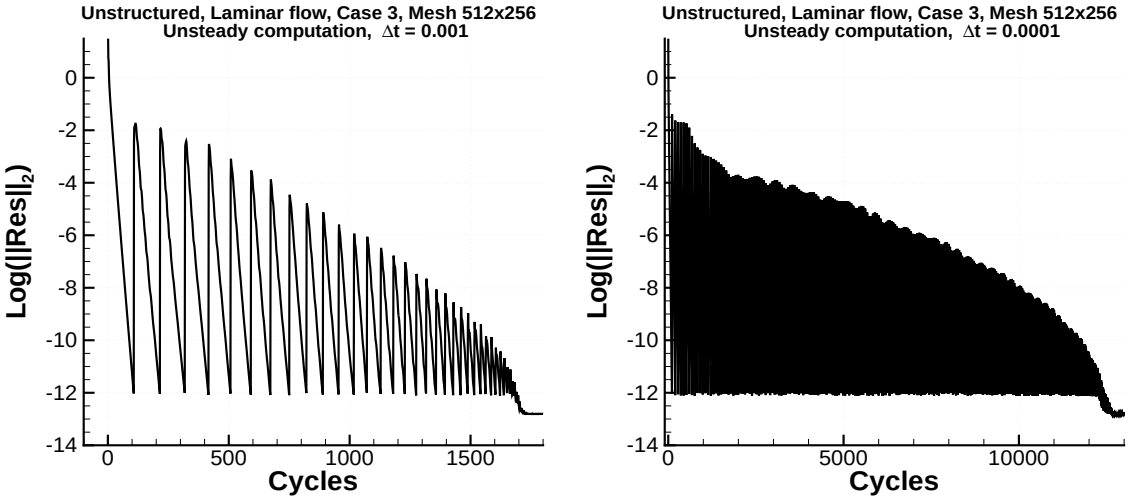
(a) (b)

Figure 7: Convergence histories for Case 1 computed with unstructured grid methods in unsteady mode (a) $\Delta t = 10^{-3}$, (b) $\Delta t = 10^{-4}$.



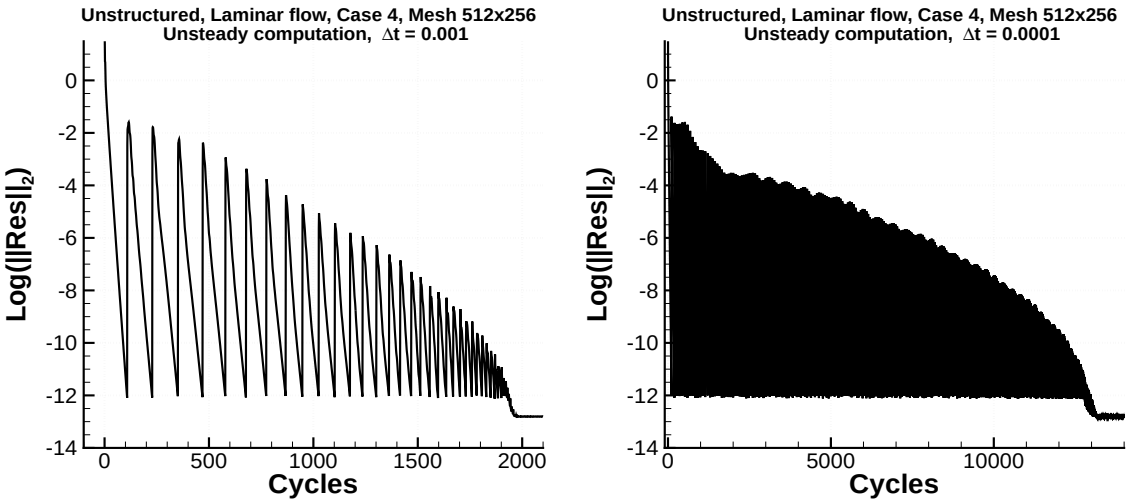
(a) (b)

Figure 8: Convergence histories for Case 2 computed with unstructured grid methods in unsteady mode (a) $\Delta t = 10^{-3}$, (b) $\Delta t = 10^{-4}$.



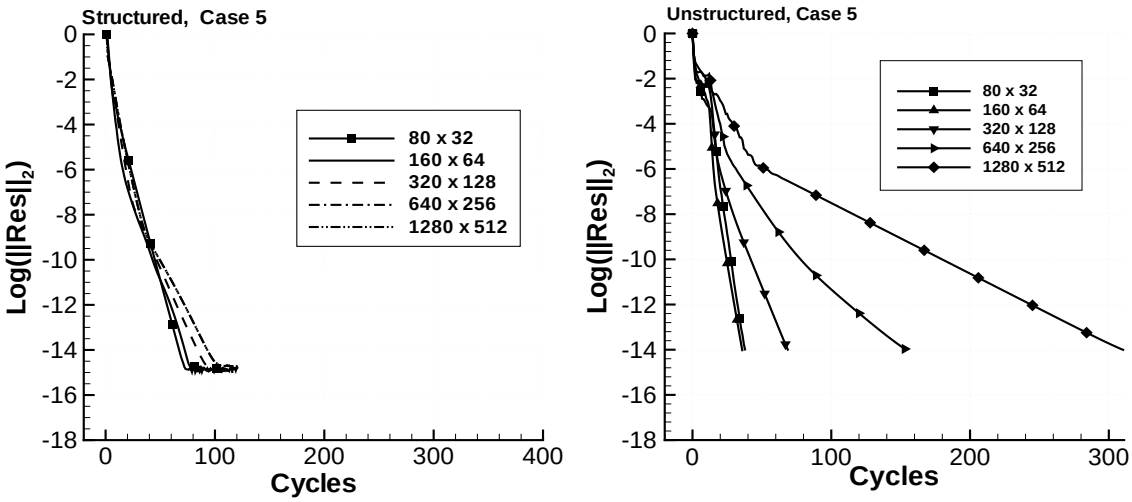
(a) (b)

Figure 9: Convergence histories for Case 3 computed with unstructured grid methods in unsteady mode (a) $\Delta t = 10^{-3}$, (b) $\Delta t = 10^{-4}$.



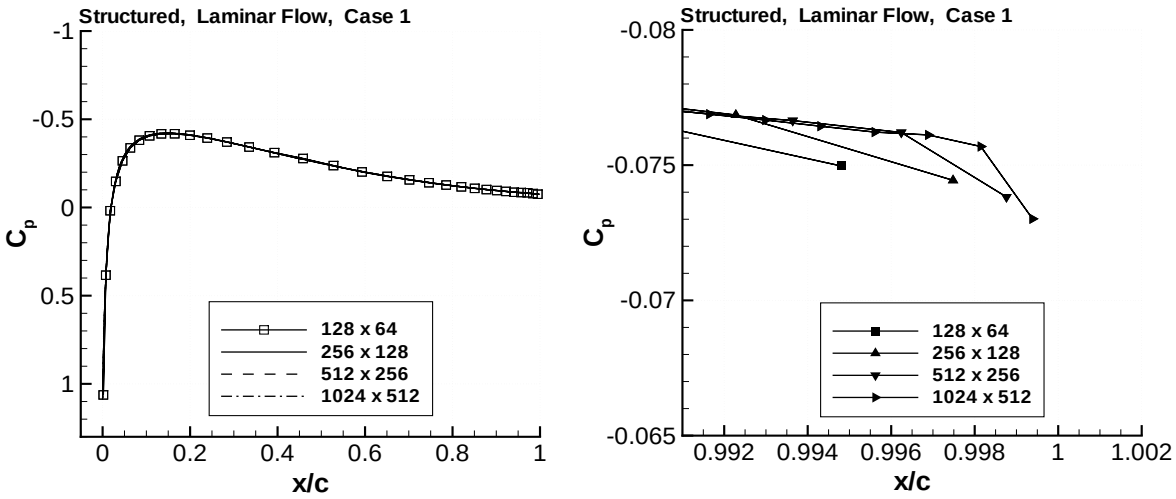
(a) (b)

Figure 10: Convergence histories for Case 4 computed with unstructured grid methods in unsteady mode (a) $\Delta t = 10^{-3}$, (b) $\Delta t = 10^{-4}$.



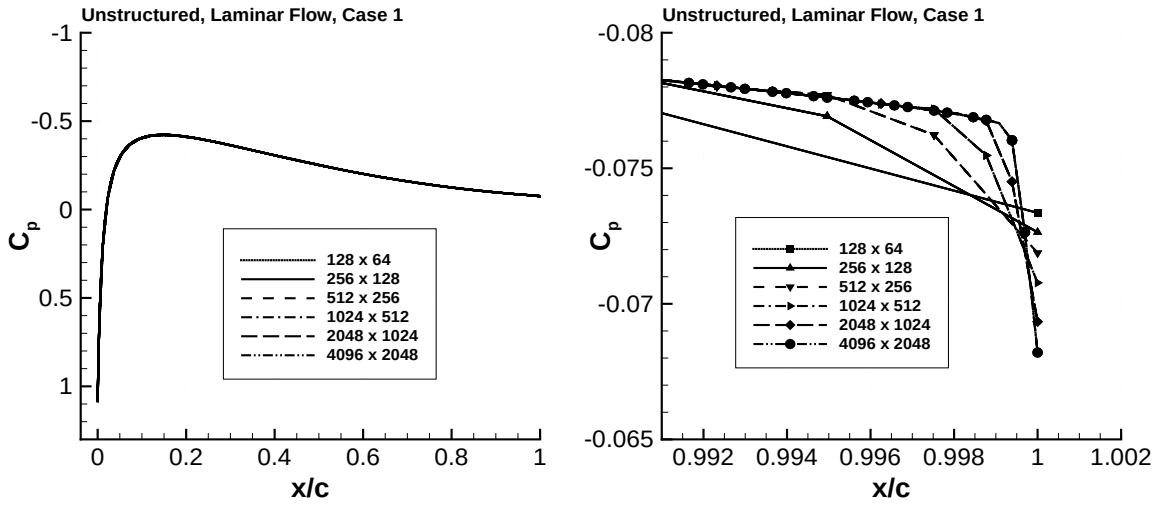
(a) (b)

Figure 11: Comparison of convergence histories for Case 5 computed with structured and unstructured grid methods. (a) Structured, (b) Unstructured.



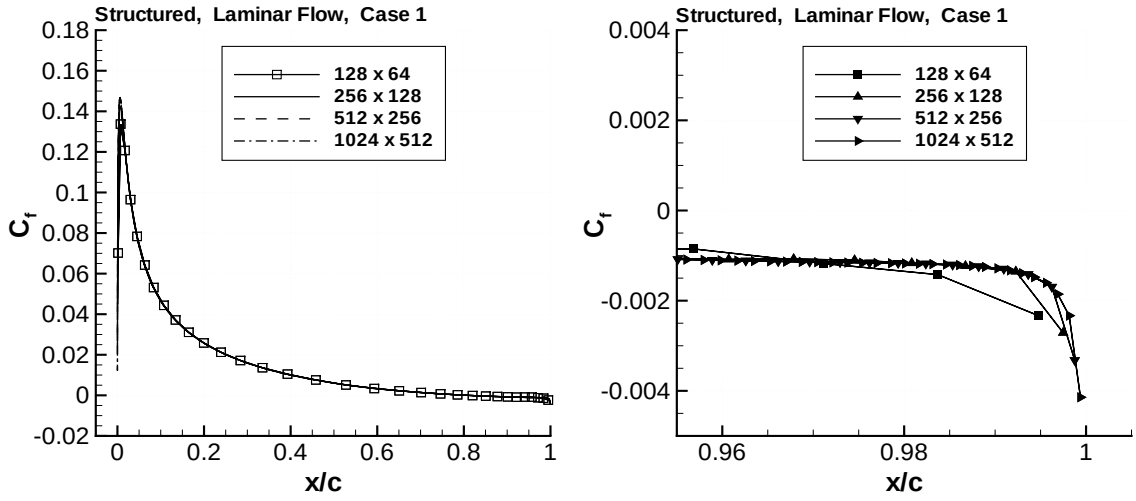
(a) (b)

Figure 12: Surface pressure distributions for Case 1 computed with structured grid method. (a) Surface pressures, (b) Trailing edge region.



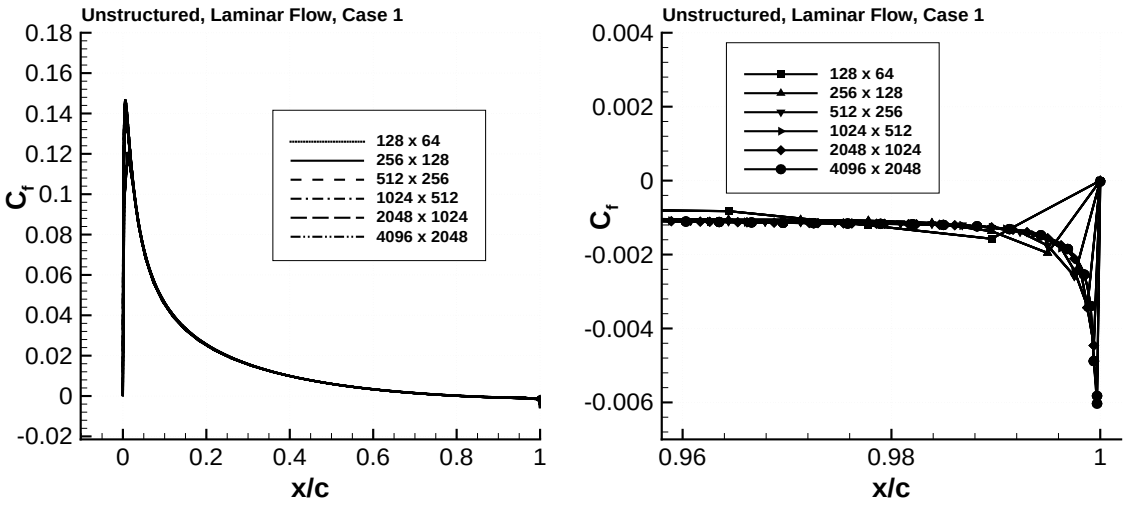
(a) (b)

Figure 13: Surface pressure distributions for Case 1 computed with unstructured grid method. (a) Surface pressures, (b) Trailing edge region.



(a) (b)

Figure 14: Surface skin-friction distributions for Case 1 computed with structured grid method. (a) Surface skin friction, (b) Trailing edge region.



(a) (b)

Figure 15: Surface skin-friction distributions for Case 1 computed with unstructured grid method. (a) Surface skin friction, (b) Trailing edge region.

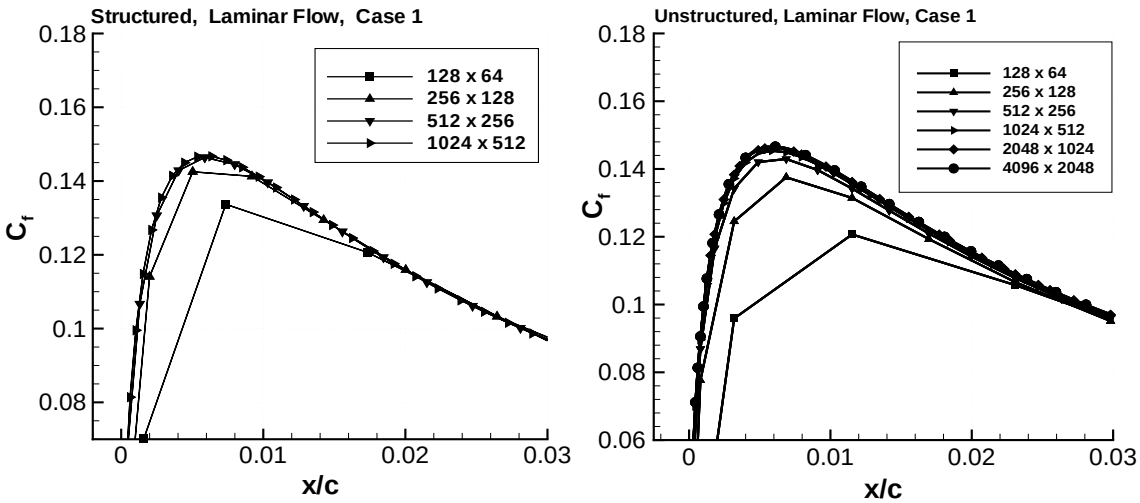


Figure 16: Leading edge skin-friction distributions for Case 1 computed with structured and unstructured grid methods. (a) Structured grid, (b) Unstructured grid.

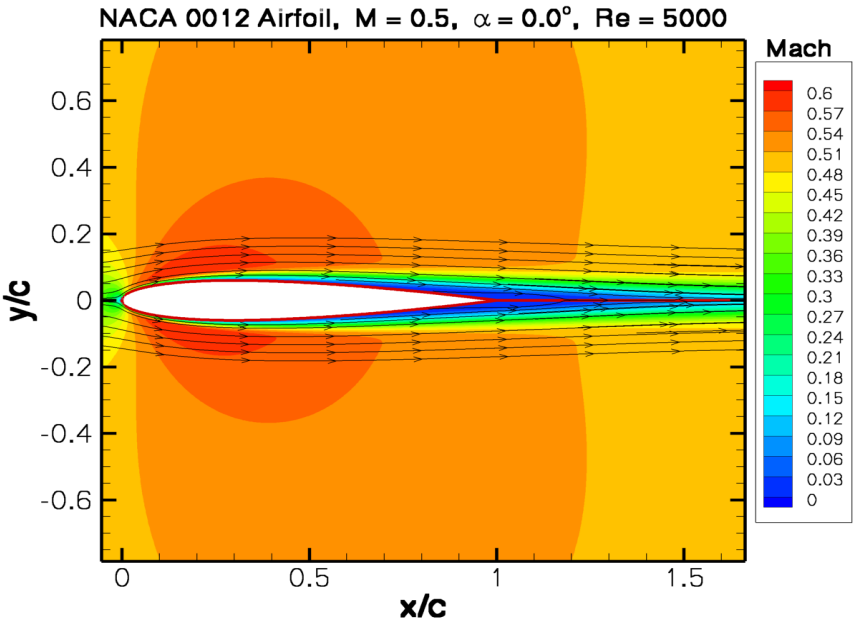


Figure 17: Mach contours and streamlines for laminar flow over NACA 0012 airfoil. Computed with structured grid method (Case 1, grid: 1024×512).

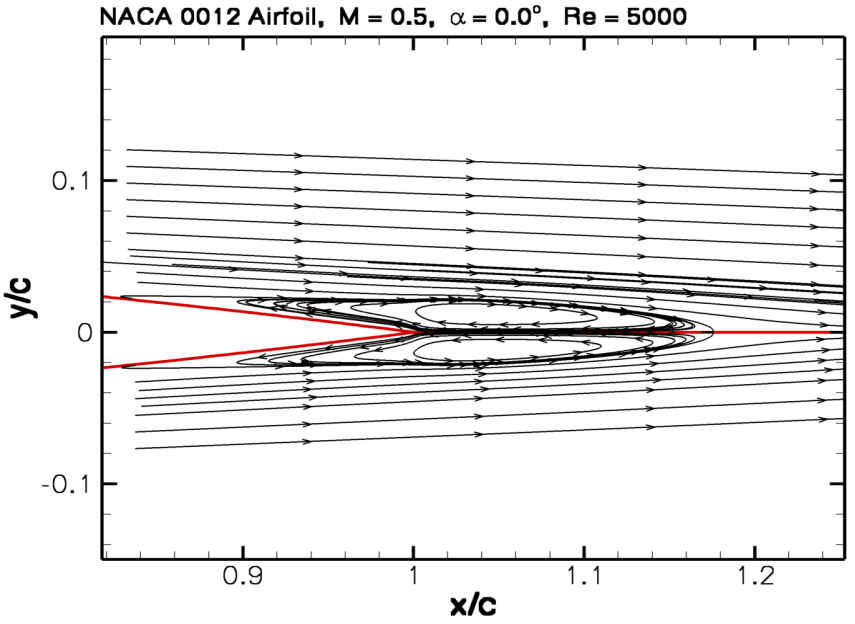
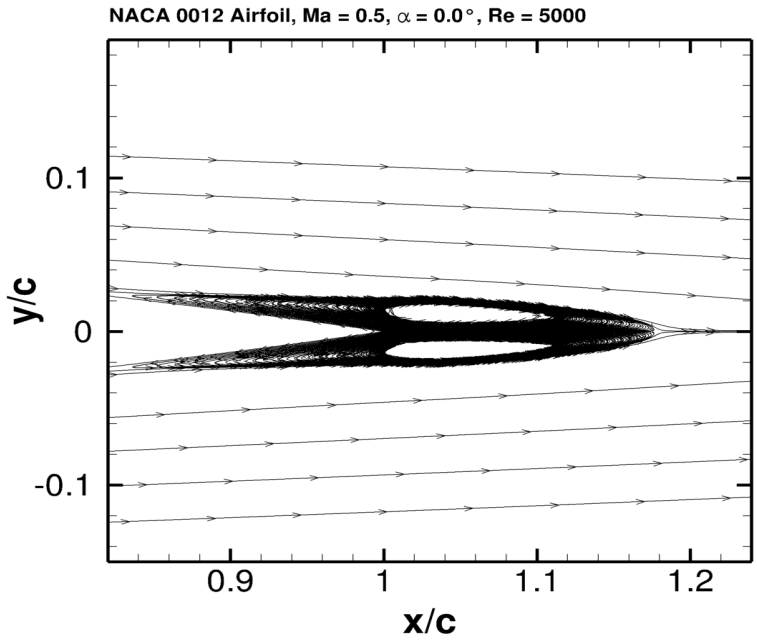
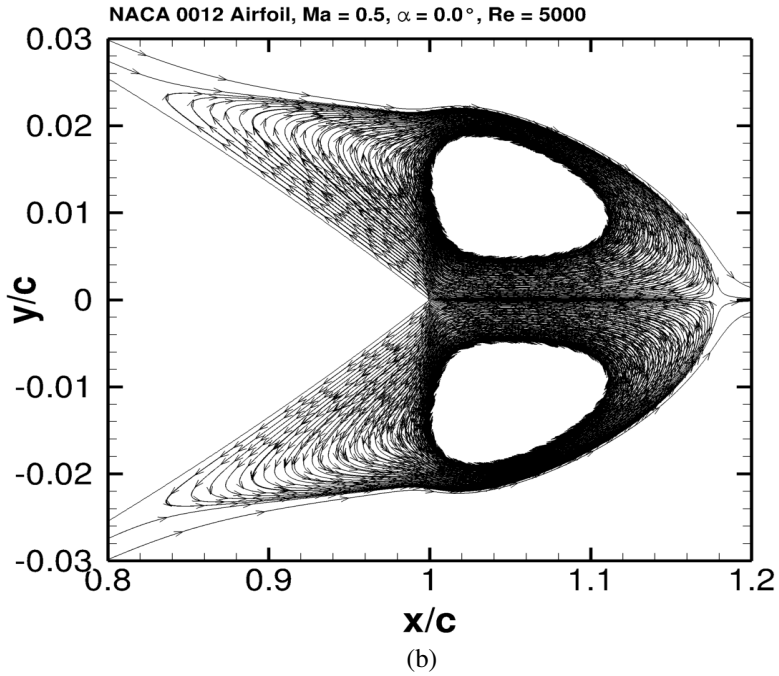


Figure 18: Blowup of streamline pattern in the airfoil trailing edge region. Computed with structured grid method (Case 1, grid: 1024×512).

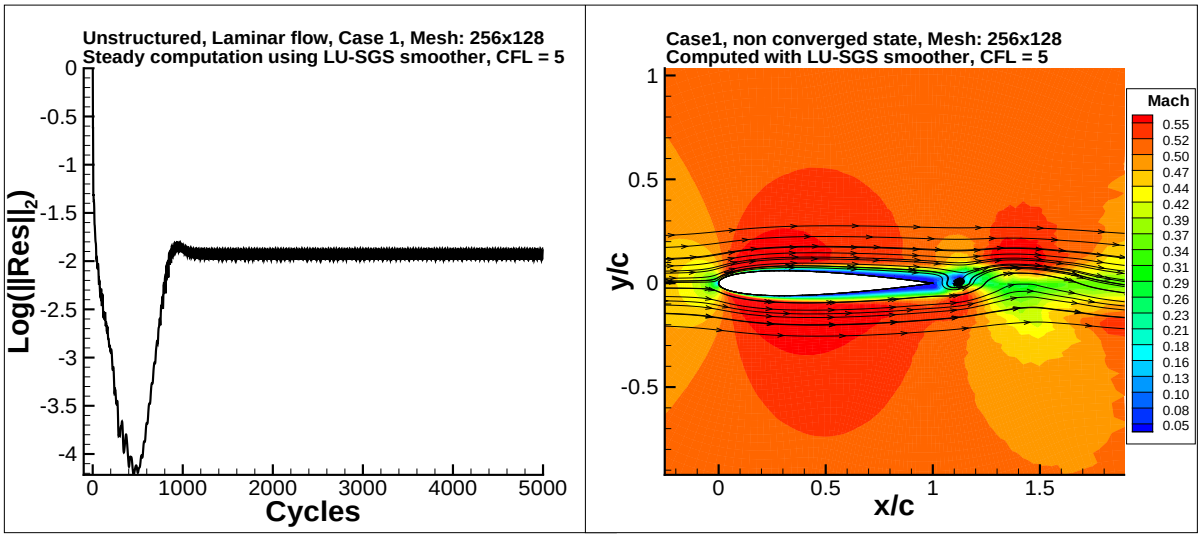


(a)



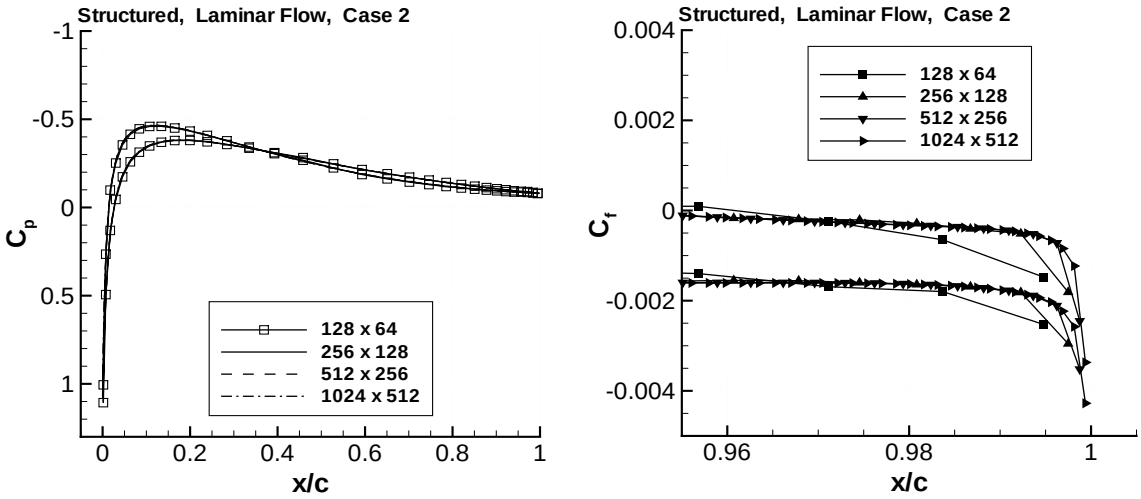
(b)

Figure 19: Streamline pattern in the airfoil trailing edge region. Computed with unstructured grid method (Case 1, grid: 4096×2048). (a) Global view, (b) Blowup.



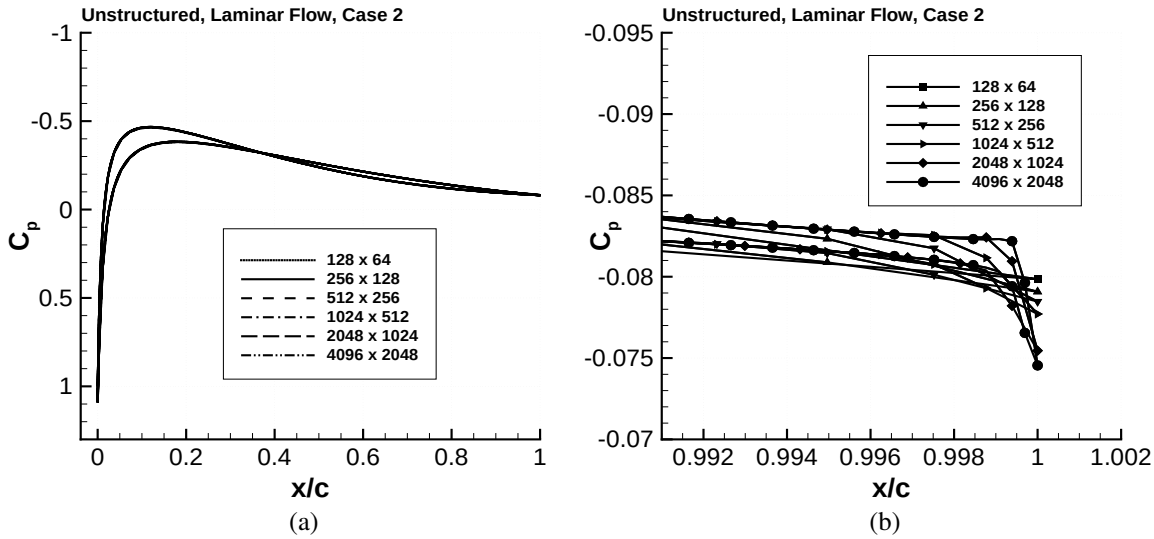
(a) (b)

Figure 20: Convergence history and Mach contours for Case 1 computed with the unstructured grid method using multigrid and LU-SGS smoother (CFL = 5). (a) Convergence history, (b) Mach contours.



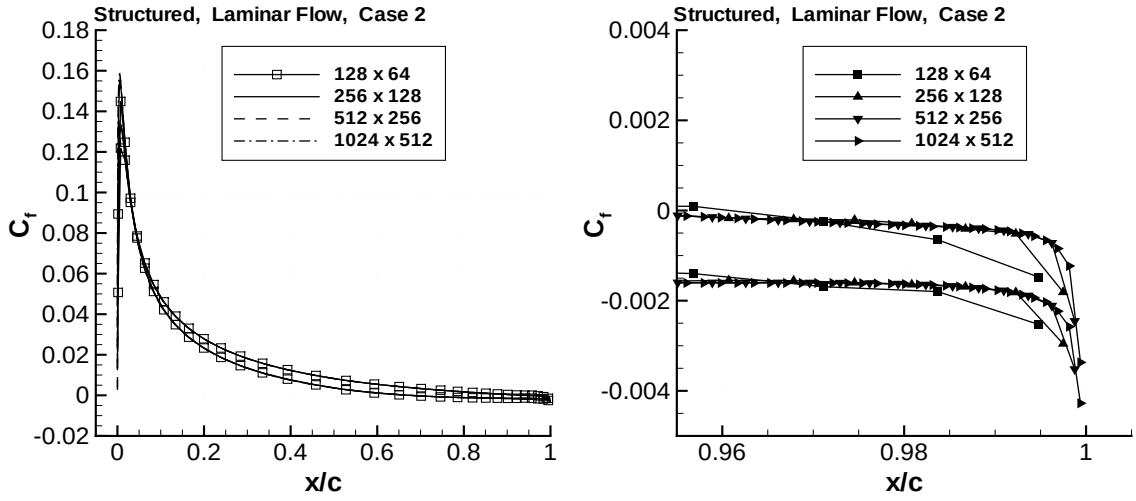
(a) (b)

Figure 21: Surface pressure distributions for Case 2 computed with structured grid method. (a) Surface pressures, (b) Trailing edge region.



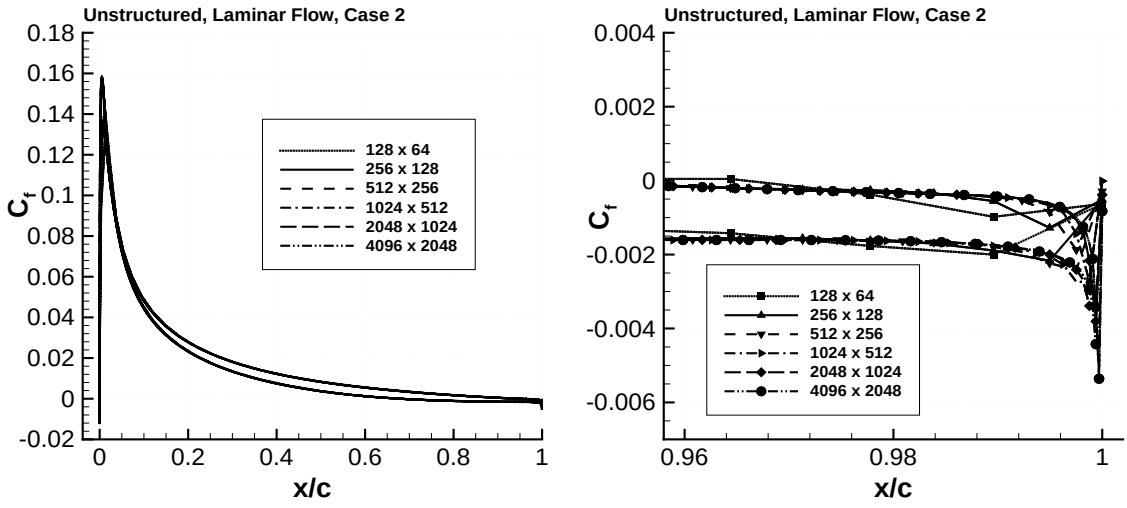
(a) (b)

Figure 22: Surface pressure distributions for Case 2 computed with unstructured grid method. (a) Surface pressures, (b) Trailing edge region.



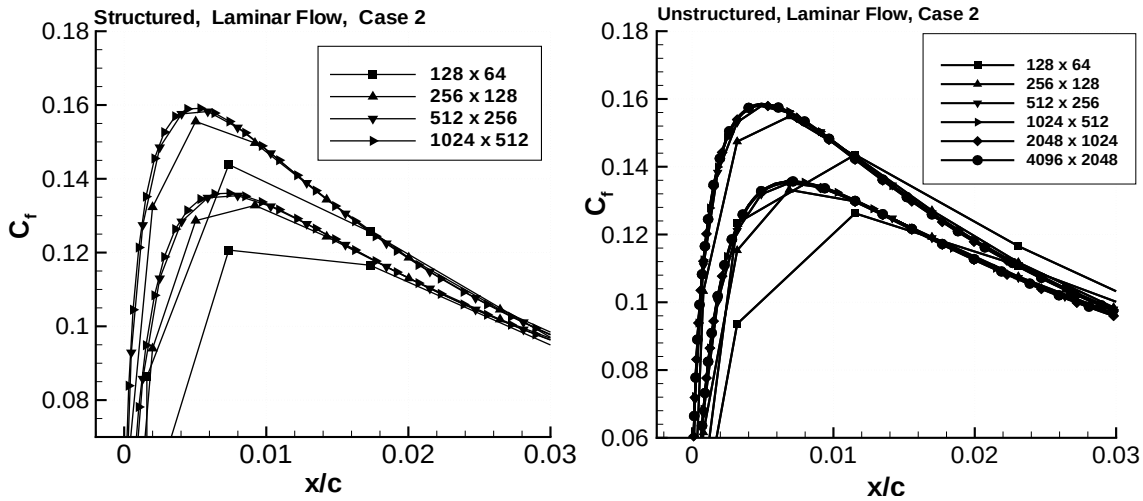
(a) (b)

Figure 23: Surface skin-friction distributions for Case 2 computed with structured grid method. (a) Surface skin friction, (b) Trailing edge region.



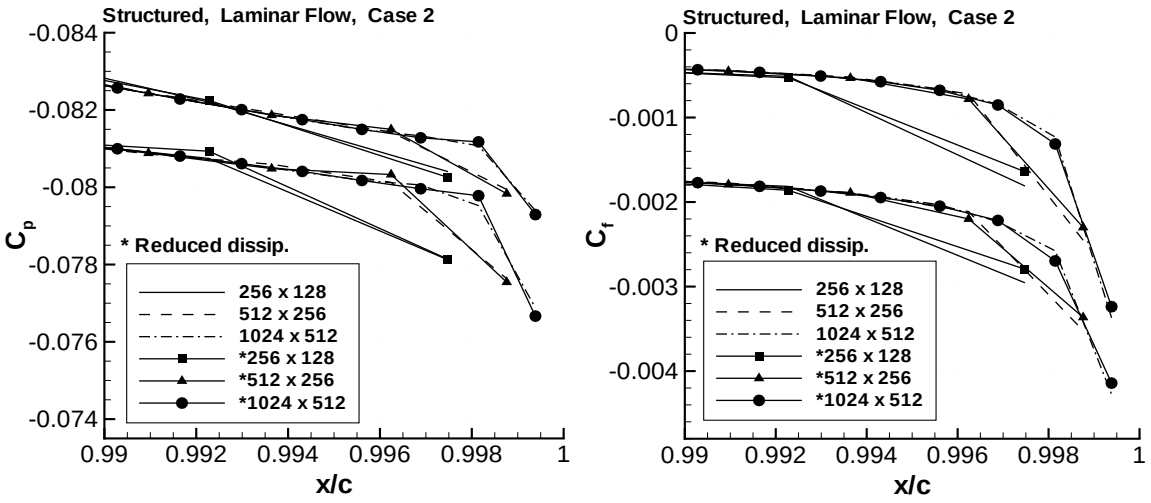
(a) (b)

Figure 24: Surface skin-friction distributions for Case 2 computed with unstructured grid method. (a) Surface skin friction, (b) Trailing edge region.



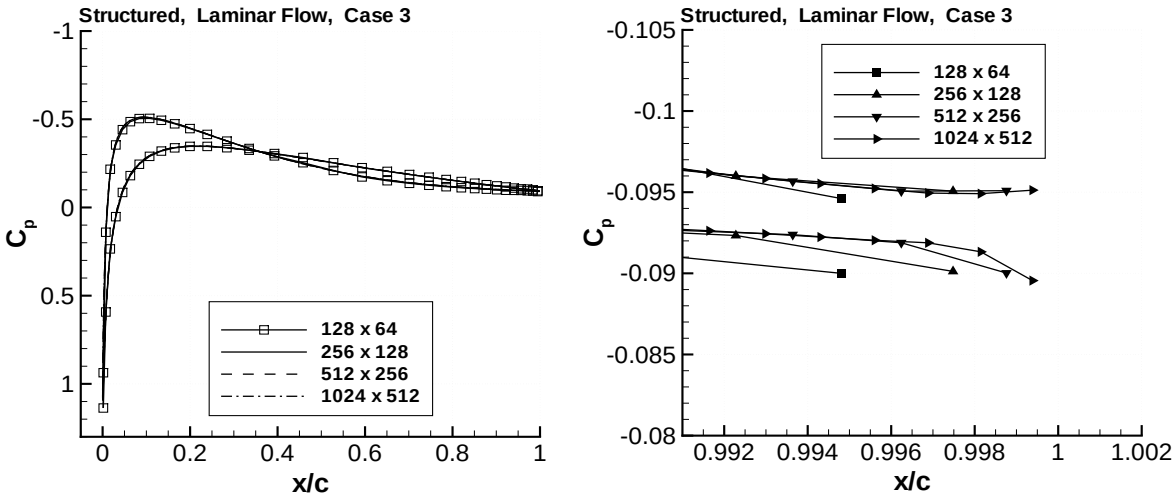
(a) (b)

Figure 25: Leading edge skin-friction distributions for Case 2 computed with structured and unstructured grid methods. (a) Structured grid, (b) Unstructured grid.



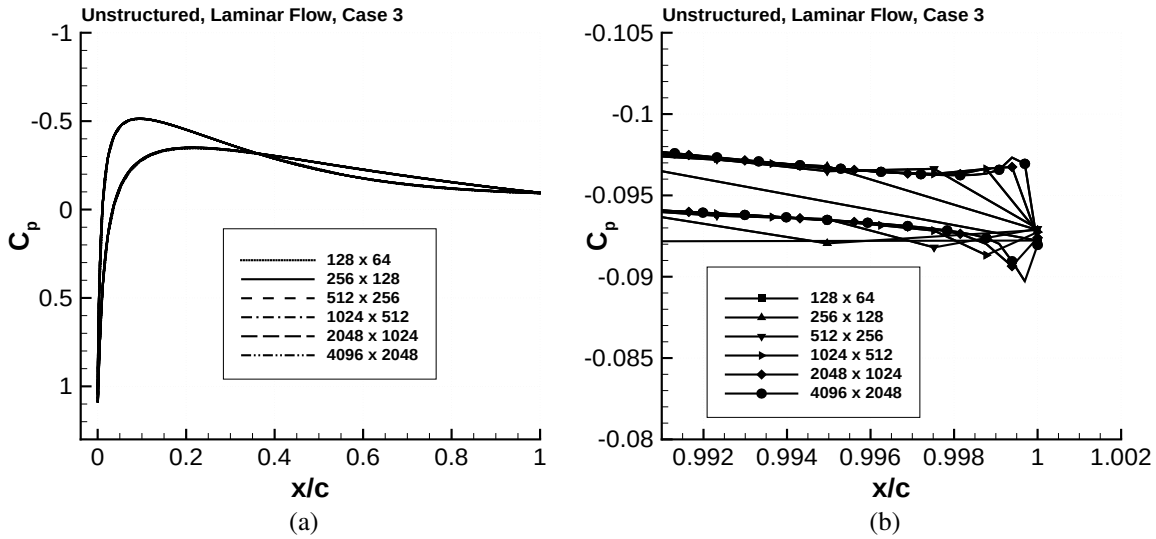
(a) (b)

Figure 26: Effect of numerical dissipation on surface pressure and skin-friction distributions in trailing edge region. Case 2 results computed with structured grid method. The symbol * beside the grid numbers in the legend means that the numerical dissipation has been reduced by a factor of two. (a) Surface pressure, (b) Skin friction.



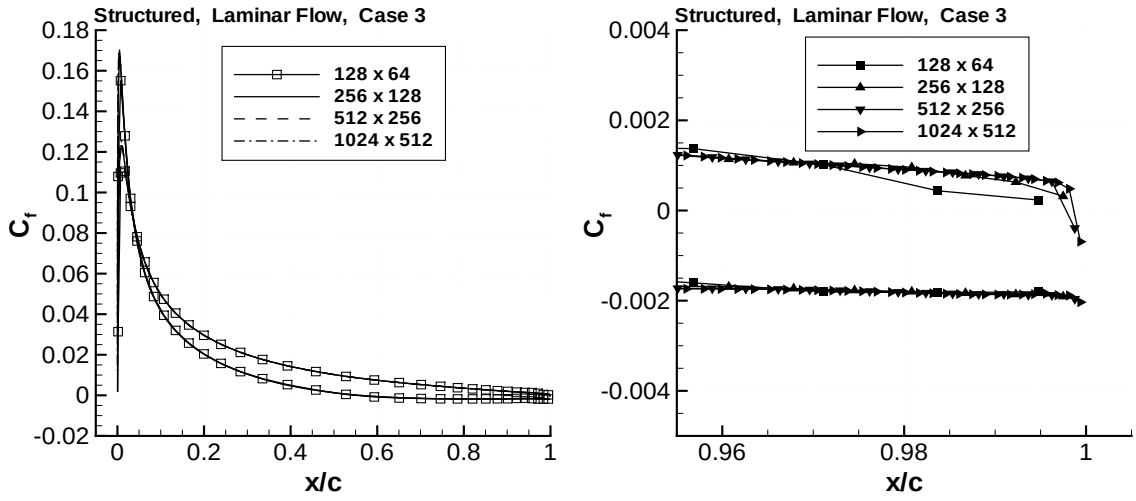
(a) (b)

Figure 27: Surface pressure distributions for Case 3 computed with structured grid method. (a) Surface pressures, (b) Trailing edge region.



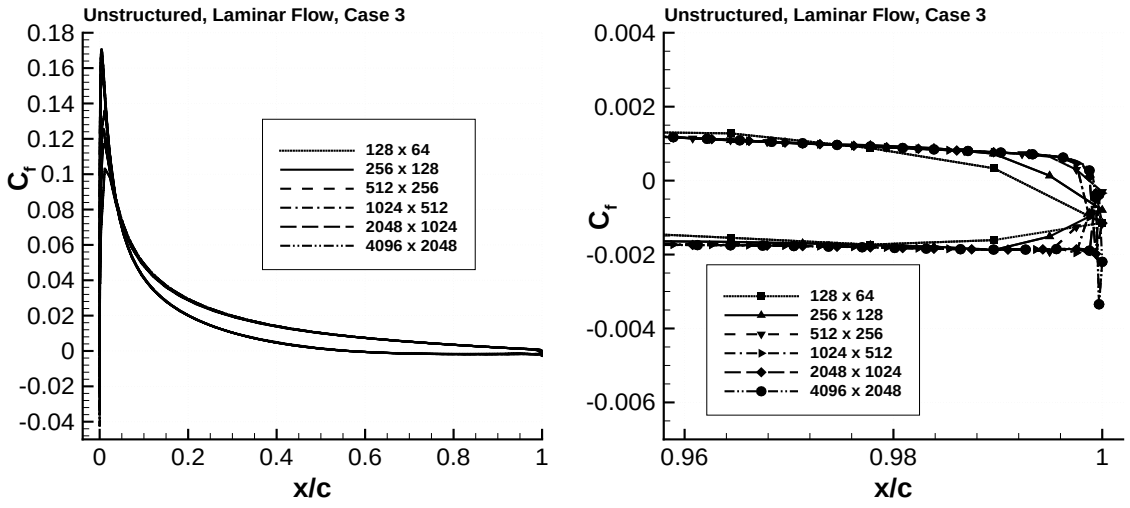
(a) (b)

Figure 28: Surface pressure distributions for Case 3 computed with unstructured grid method. (a) Surface pressures, (b) Trailing edge region.



(a) (b)

Figure 29: Surface skin-friction distributions for Case 3 computed with structured grid method. (a) Surface skin friction, (b) Trailing edge region.



(a) (b)

Figure 30: Surface skin-friction distributions for Case 3 computed with unstructured grid method. (a) Surface skin friction, (b) Trailing edge region.

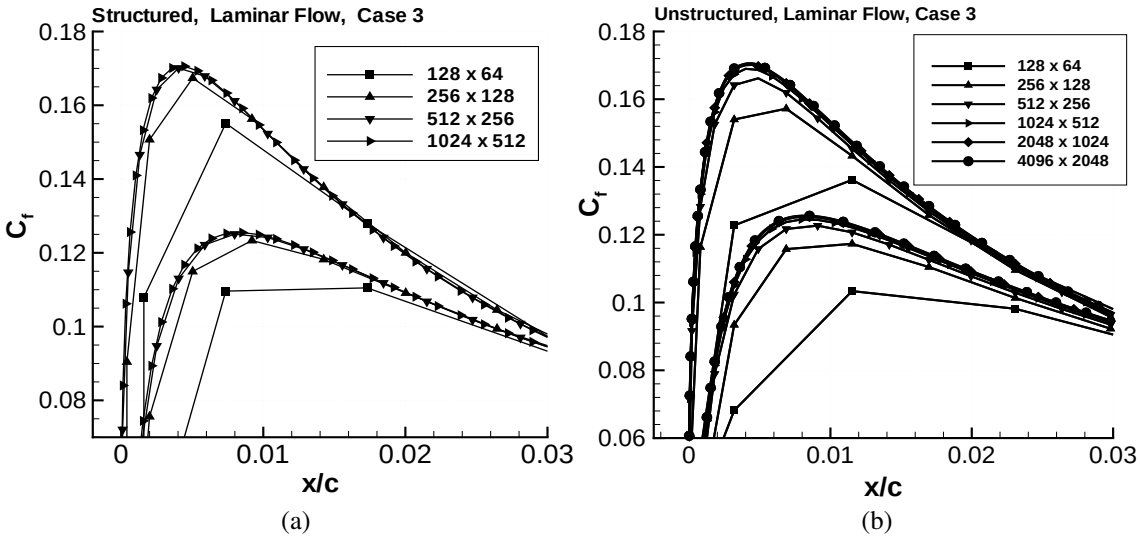
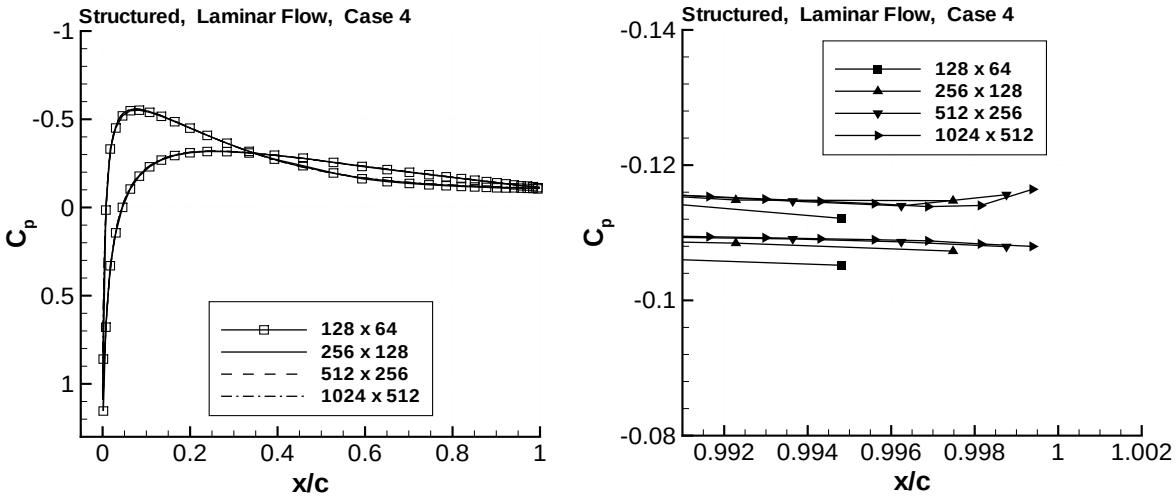
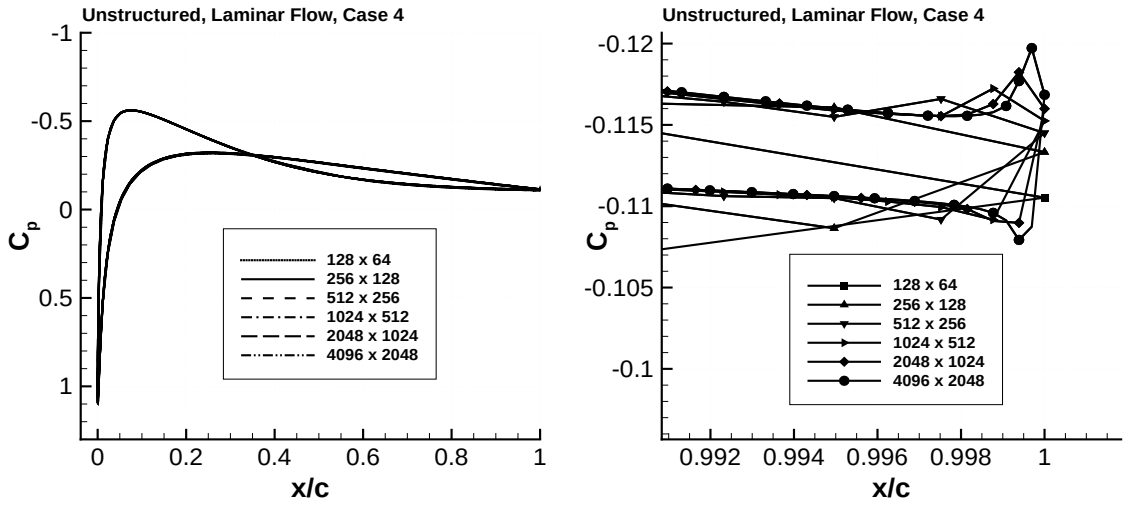


Figure 31: Leading edge skin-friction distributions for Case 3 computed with structured and unstructured grid methods. (a) Structured grid, (b) Unstructured grid.



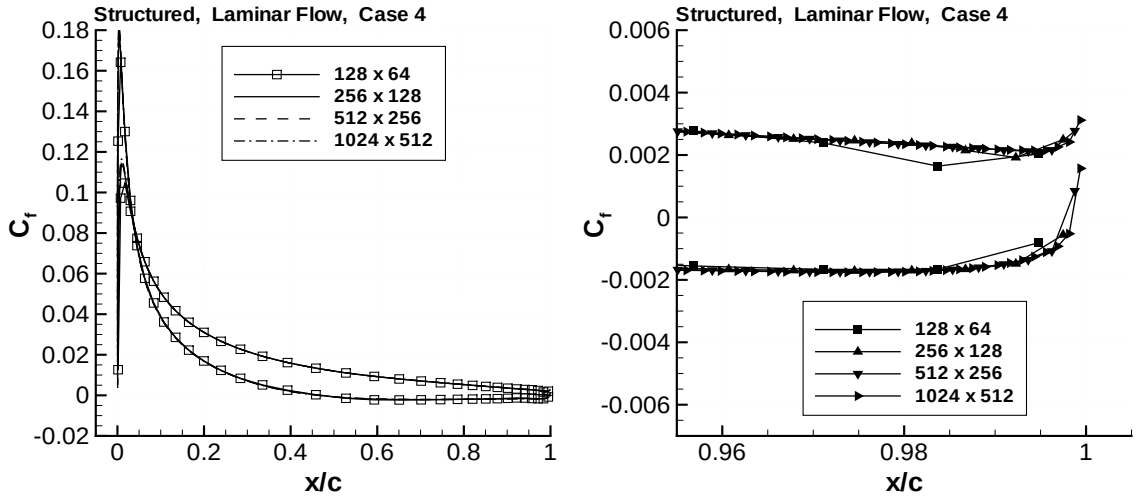
(a) (b)

Figure 32: Surface pressure distributions for Case 4 computed with structured grid method. (a) Surface pressures, (b) Trailing edge region.



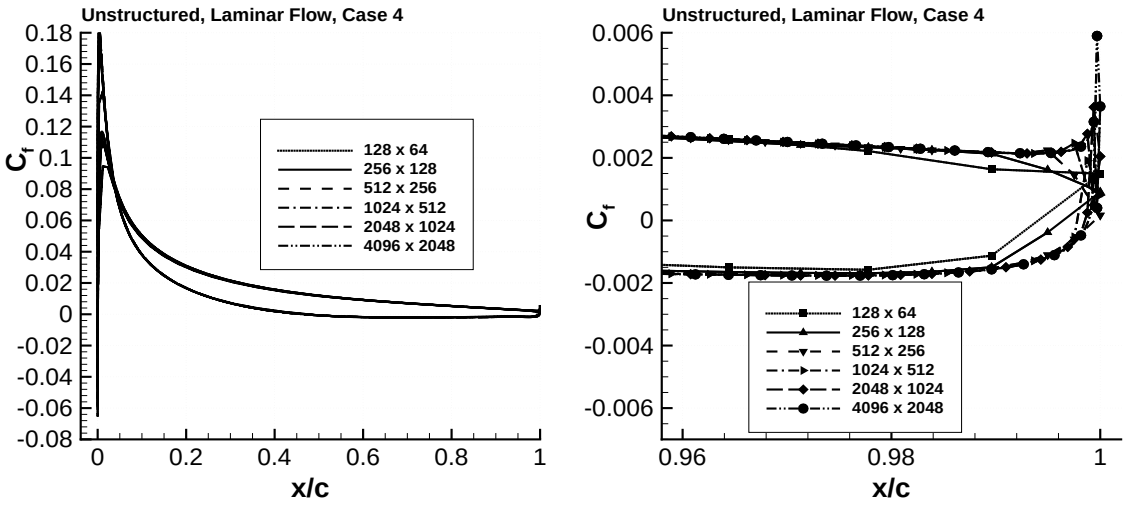
(a) (b)

Figure 33: Surface pressure distributions for Case 4 computed with unstructured grid method. (a) Surface pressures, (b) Trailing edge region.



(a) (b)

Figure 34: Surface skin-friction distributions for Case 4 computed with structured grid method. (a) Surface skin friction, (b) Trailing edge region.



(a) (b)

Figure 35: Surface skin-friction distributions for Case 4 computed with unstructured grid method. (a) Surface skin friction, (b) Trailing edge region.

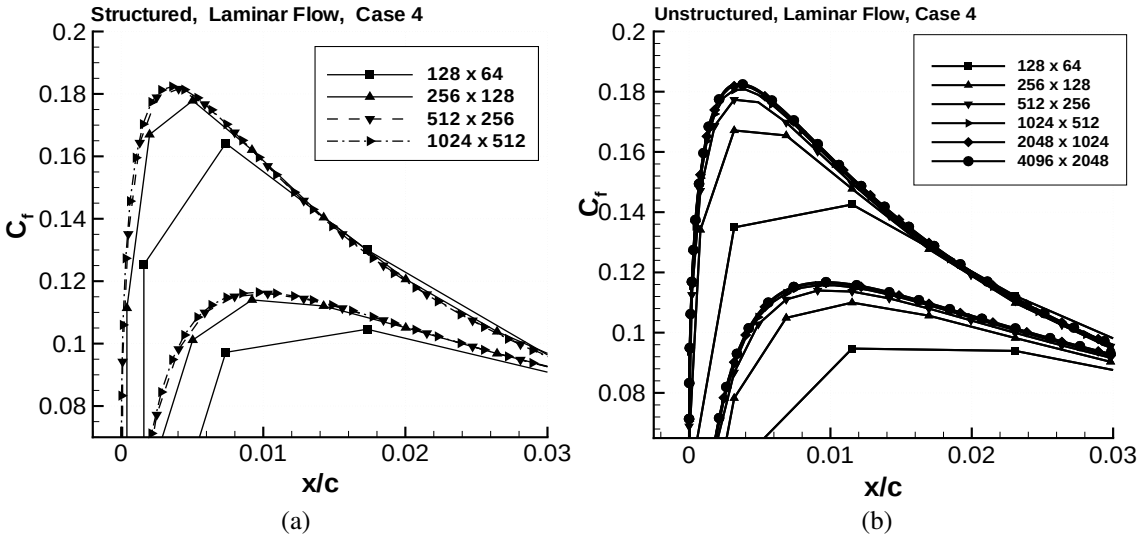


Figure 36: Leading edge skin-friction distributions for Case 4 computed with structured and unstructured grid methods. (a) Structured grid, (b) Unstructured grid.

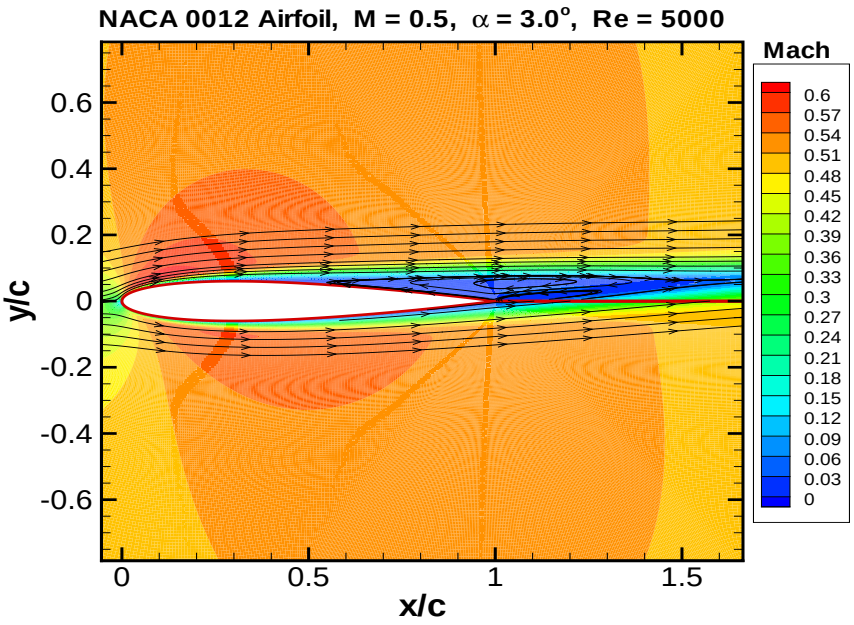


Figure 37: Mach contours and streamlines for laminar flow over NACA 0012 airfoil. Computed with structured grid method (Case 4, grid: 1024×512).

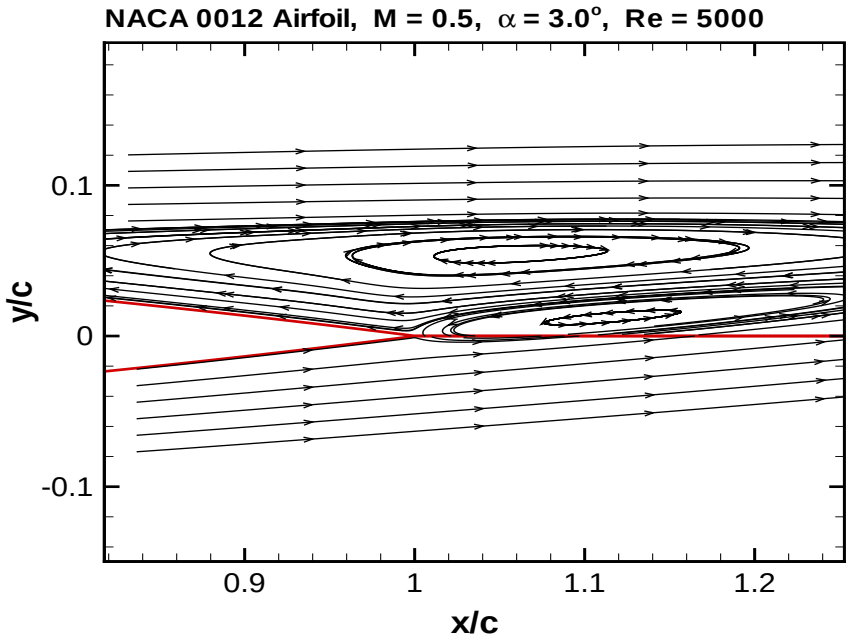


Figure 38: Blowup of streamline pattern in the airfoil trailing edge region. Computed with structured grid method (Case 4, grid: 1024×512).

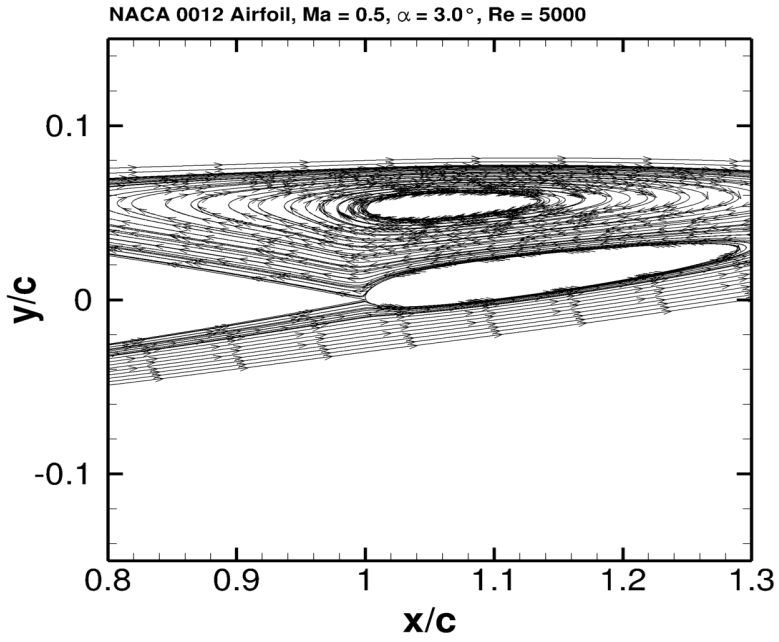


Figure 39: Blowup of streamline pattern in the airfoil trailing edge region. Computed with unstructured grid method (Case 4, grid: 4096×2048).

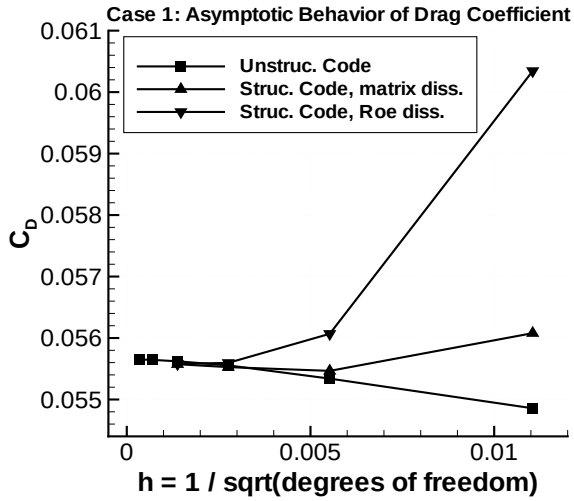
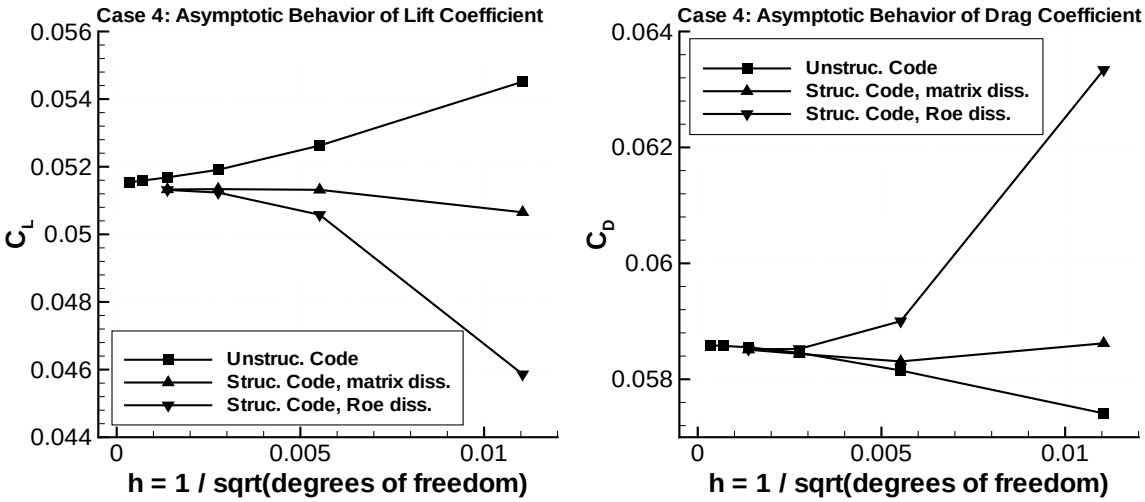
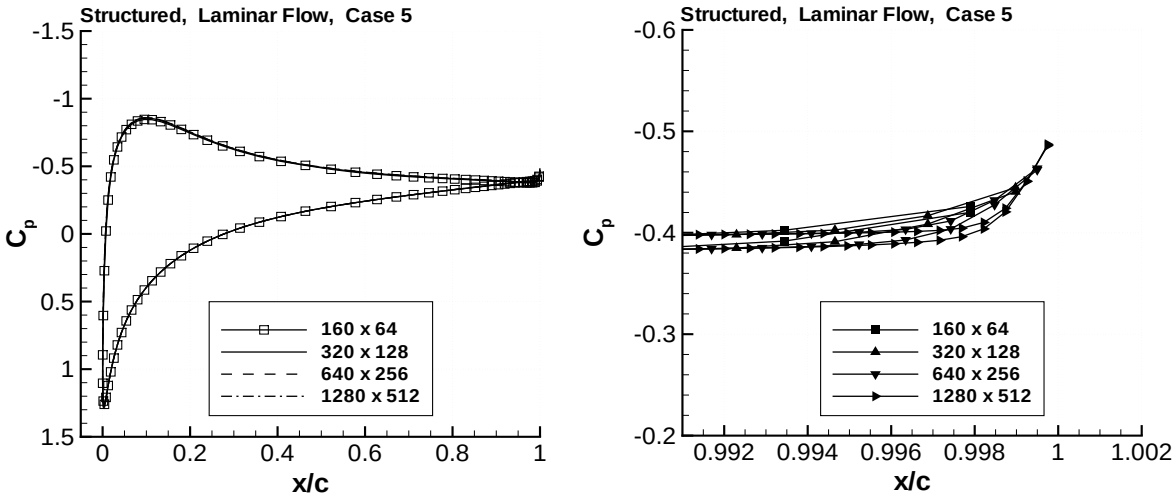


Figure 40: Grid convergence of for Case 1 ($\alpha = 0°$) computed with structured grid and unstructured grid methods.



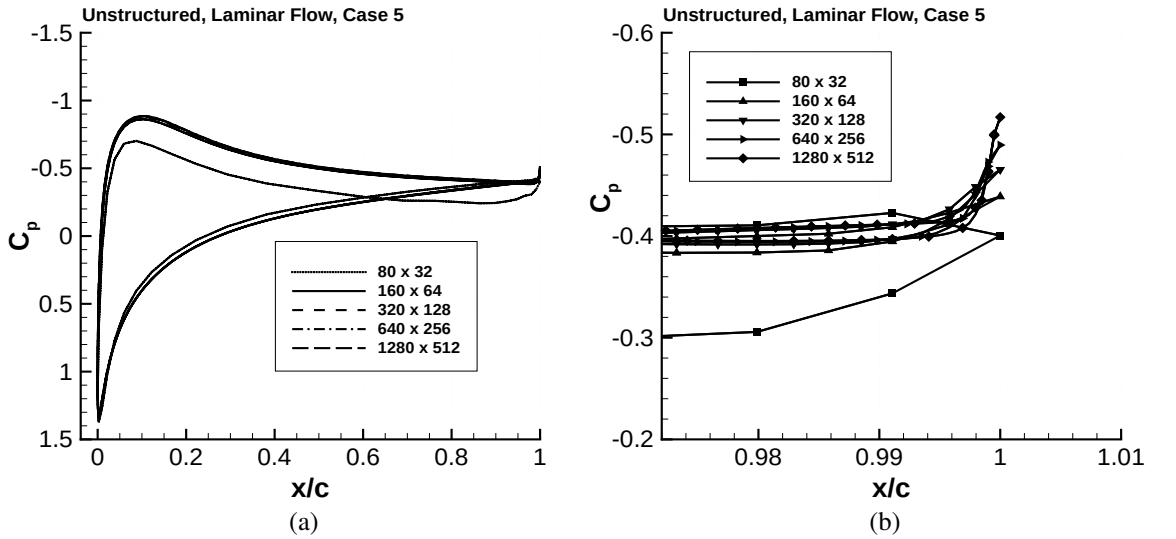
(a) (b)

Figure 41: Grid convergence for Case 4 ($\alpha = 3°$) computed with structured and unstructured grid methods. (a) C_L, (b) C_D.



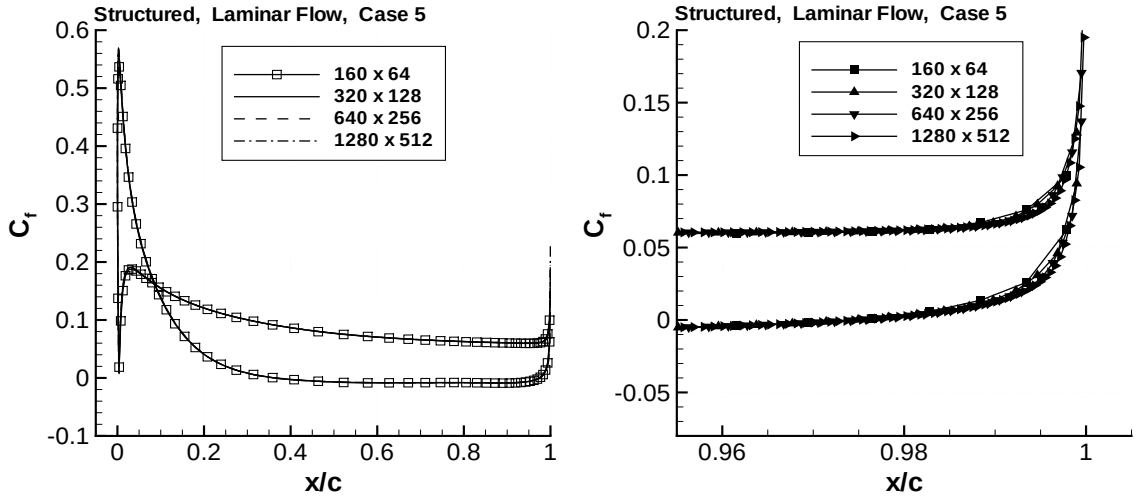
(a) (b)

Figure 42: Surface pressure distributions for Case 5 computed with structured grid method. (a) Surface pressures, (b) Trailing edge region.



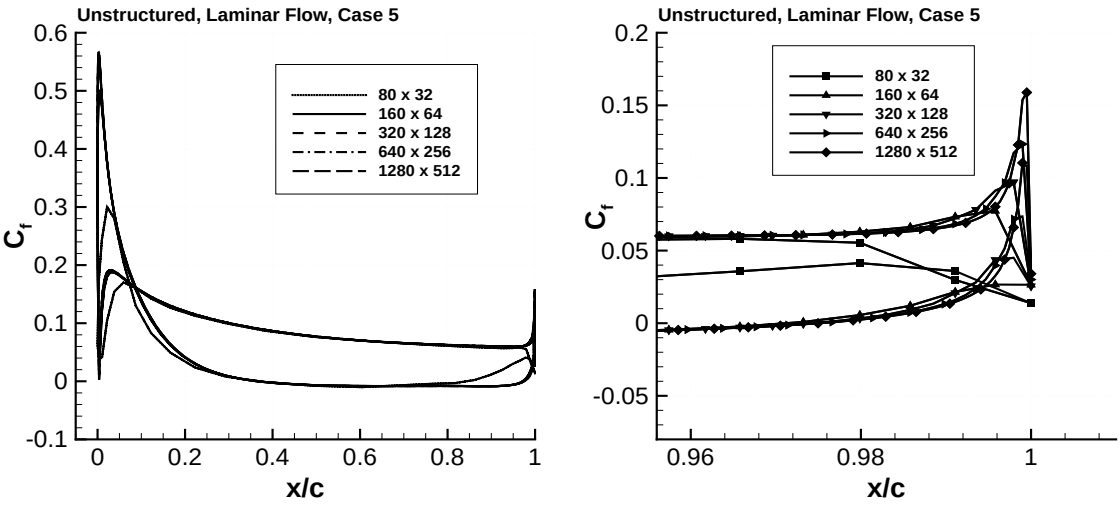
(a) (b)

Figure 43: Surface pressure distributions for Case 5 computed with unstructured grid method. (a) Surface pressures, (b) Trailing edge region.



(a) (b)

Figure 44: Surface skin-friction distributions for Case 5 computed with structured grid method. (a) Surface skin friction, (b) Trailing edge region.



(a) (b)

Figure 45: Surface skin-friction distributions for Case 5 computed with unstructured grid method. (a) Surface skin friction, (b) Trailing edge region.

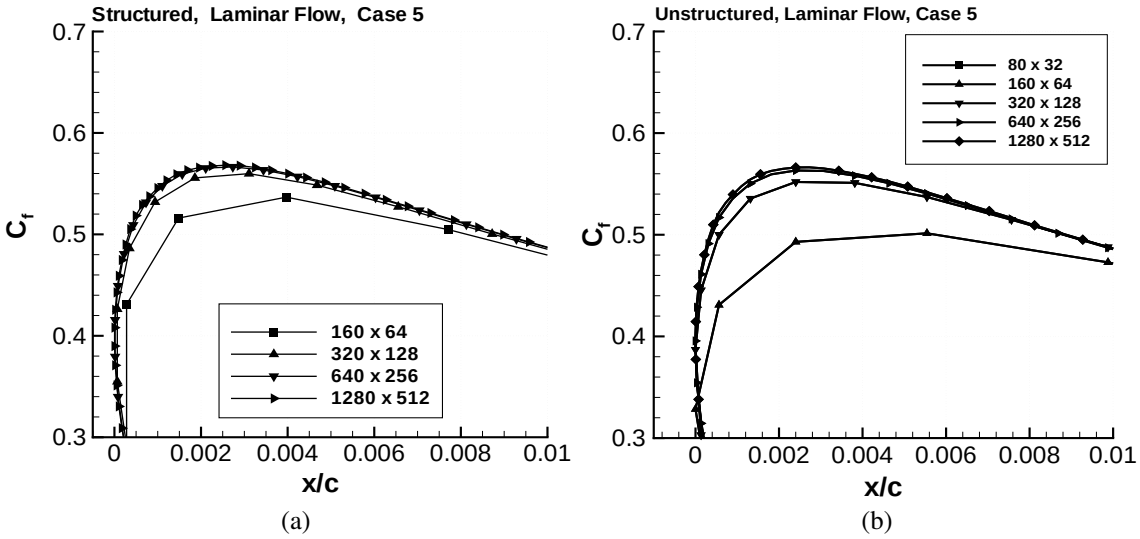


Figure 46: Leading edge skin-friction distributions for Case 5 computed with structured and unstructured grid methods. (a) Structured grid, (b) Unstructured grid.

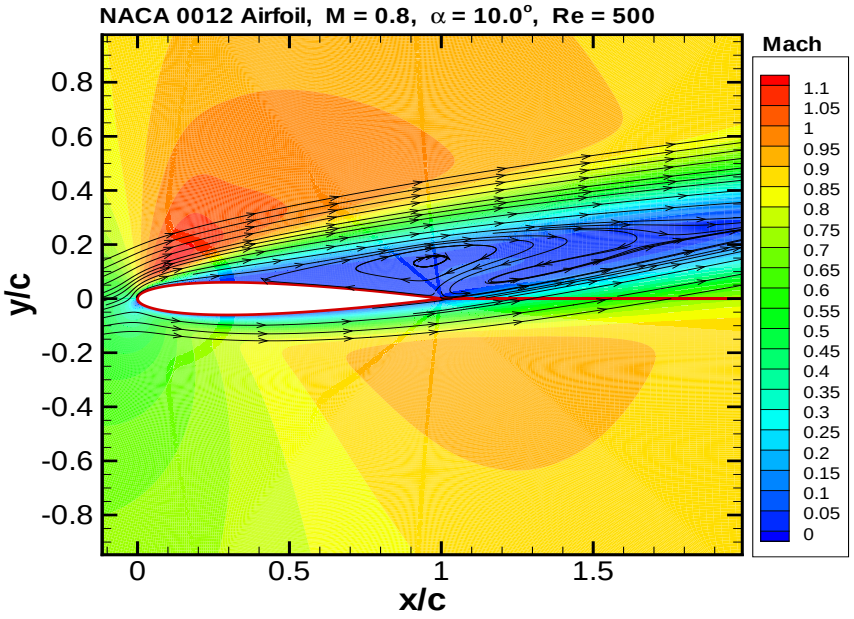


Figure 47: Mach contours and streamlines for laminar flow over NACA 0012 airfoil. Computed with structured grid method (Case 5, grid: 1280×512).

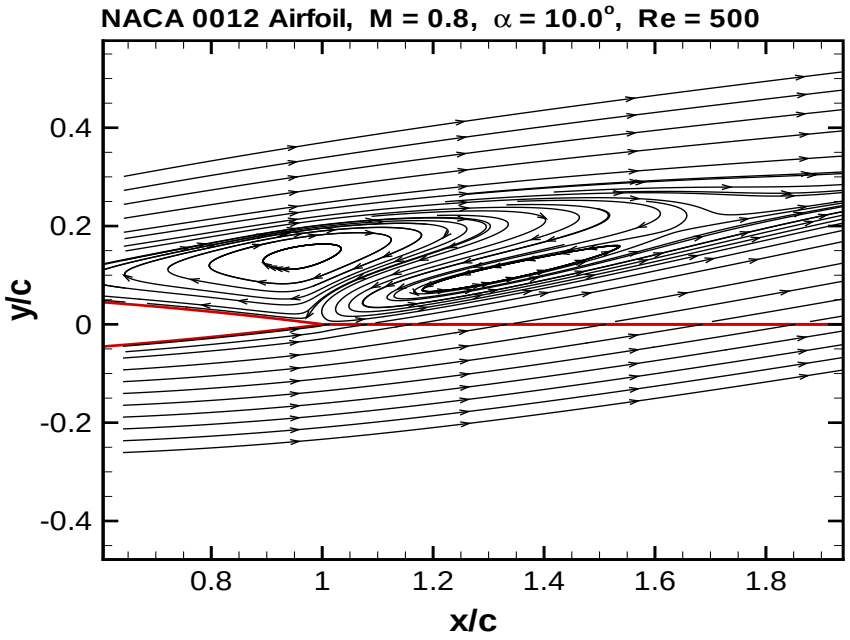


Figure 48: Blowup of streamline pattern in the airfoil trailing edge region. Computed with structured grid method (Case 5, grid: 1280×512).

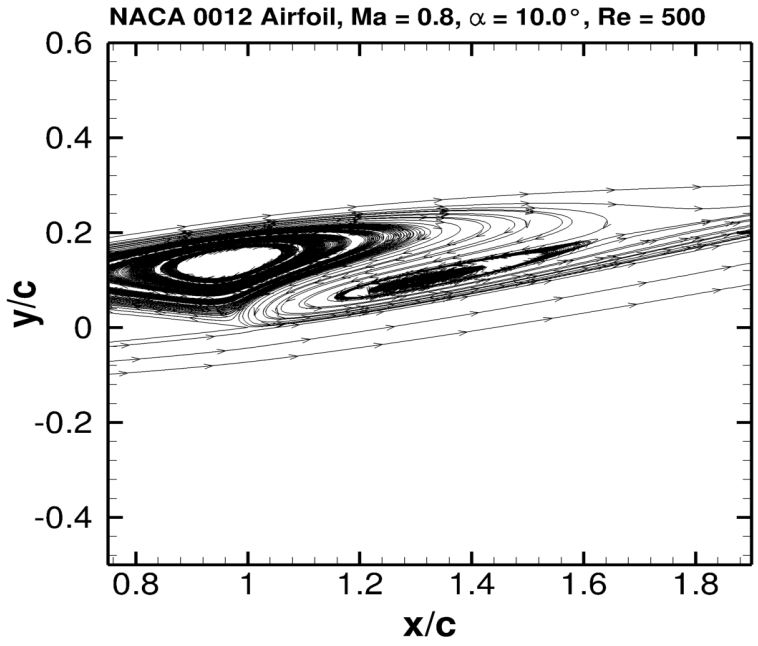


Figure 49: Blowup of streamline pattern in the airfoil trailing edge region. Computed with unstructured grid method (Case 5, grid: 1280×512).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-01-2016	Technical Memorandum	

4. TITLE AND SUBTITLE

Comparison of NACA 0012 Laminar Flow Solutions: Structured and Unstructured Grid Methods

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Swanson, R.C.; Langer, S.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

109492.02.07.01.03.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20560

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA-TM-2016-219003

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 01
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this paper we consider the solution of the compressible Navier-Stokes equations for a class of laminar airfoil flows. The principal objective of this paper is to demonstrate that members of this class of laminar flows have steady-state solutions. These laminar airfoil flow cases are often used to evaluate accuracy, stability and convergence of numerical solution algorithms for the Navier-Stokes equations. In recent years, such flows have also been used as test cases for high-order numerical schemes. While generally consistent steady-state solutions have been obtained for these flows using higher order schemes, a number of results have been published with various solutions, including unsteady ones. We demonstrate with two different numerical methods and a range of meshes with a maximum density that exceeds 8×10 grid points that steady-state solutions are obtained. Furthermore, numerical evidence is presented that even when solving the equations with an unsteady algorithm, one obtains steady-state solutions.

15. SUBJECT TERMS

Airfoils; Computational grid; Laminar flow; Numerical analysis; Steady state

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	61	(757) 864-9658