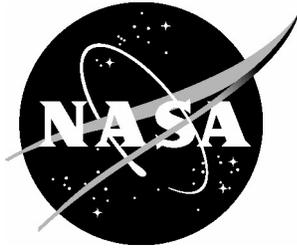


NASA/TM-2005-213909



Introduction of the ASP3D Computer Program for Unsteady Aerodynamic and Aeroelastic Analyses

John T. Batina
NASA Langley Research Center, Hampton, Virginia

December 2005

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.

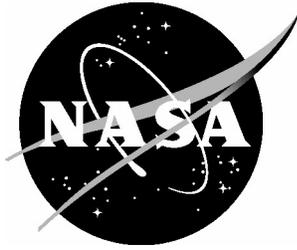
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with 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
7121 Standard Drive
Hanover, MD 21076-1320

NASA/TM-2005-213909



Introduction of the ASP3D Computer Program for Unsteady Aerodynamic and Aeroelastic Analyses

John T. Batina
NASA Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

December 2005

Available from:

NASA Center for AeroSpace Information (CASI)
7121 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

Introduction of the ASP3D Computer Program for Unsteady Aerodynamic and Aeroelastic Analyses

John T. Batina^φ
NASA Langley Research Center
Hampton, Virginia 23681

Abstract

A new computer program has been developed called ASP3D (Advanced Small Perturbation – 3D), which solves the small perturbation potential flow equation in an advanced form including mass-consistent surface and trailing wake boundary conditions, and entropy, vorticity, and viscous effects. The purpose of the program is for unsteady aerodynamic and aeroelastic analyses, especially in the nonlinear transonic flight regime. The program exploits the simplicity of stationary Cartesian meshes with the movement or deformation of the configuration under consideration incorporated into the solution algorithm through a planar surface boundary condition. The new ASP3D code is the result of a decade of developmental work on improvements to the small perturbation formulation, performed while the author was employed as a Senior Research Scientist in the Configuration Aerodynamics Branch at the NASA Langley Research Center. The ASP3D code is a significant improvement to the state-of-the-art for transonic aeroelastic analyses over the CAP-TSD code (Computational Aeroelasticity Program – Transonic Small Disturbance), which was developed principally by the author in the mid-1980s. The author is in a unique position as the developer of both computer programs to compare, contrast, and ultimately make conclusions regarding the underlying formulations and utility of each code. The paper describes the salient features of the ASP3D code including the rationale for improvements in comparison with CAP-TSD. Numerous results are presented to demonstrate the ASP3D capability. The general conclusion is that the new ASP3D capability is superior to the older CAP-TSD code because of the myriad improvements developed and incorporated.

Introduction

There has been considerable interest since the 1970s in developing computational fluid dynamics computer programs for the prediction of unsteady aerodynamic phenomena and the analysis of aeroelastic systems especially in the transonic flight regime.¹⁻³ A notable example of such a computer program is the XTRAN3S code⁴ developed by Boeing under United States Air Force sponsorship. The XTRAN3S code solves the transonic small disturbance (TSD) equation with a time-accurate alternating direction implicit (ADI) algorithm and coupled the flow equations with the structural equations of motion for simultaneous time integration for aeroelastic analyses. The code

^φ Retired Research Consultant; Associate Fellow, American Institute of Aeronautics and Astronautics; formerly employed as a Senior Research Scientist, Configuration Aerodynamics Branch, NASA Langley Research Center, Hampton, Virginia.

exploits the simplicity of stationary Cartesian meshes with the movement or deformation of the configuration under consideration incorporated into the solution algorithm through a planar surface boundary condition. However, several cross-derivative terms of the TSD equation were treated explicitly in the XTRAN3S ADI algorithm, which led to stability restrictions in applications of the code. The restrictions tended to be more severe for highly swept and/or tapered planforms⁵ because the explicit terms scale with the grid metrics related to the shearing of the grid that occurs with such planforms. Also, the code was not written in an optimal way to run efficiently on vector processors (supercomputers), which became the computer architecture of choice for such applications in the 1980s.

The CAP-TSD code⁶ was developed at NASA Langley Research Center in the mid-1980s to supersede the XTRAN3S code because of advances in computational fluid dynamics (CFD) technology that made the numerics in XTRAN3S obsolete. The CAP-TSD code solves the transonic small disturbance equation with a time-accurate approximate factorization (AF) algorithm,⁵ and like XTRAN3S, the flow equations are coupled with the structural equations of motion for simultaneous time-integration for aeroelastic analyses. The AF algorithm within the code is completely implicit, and thus, the stability restrictions experienced using the XTRAN3S code for swept/tapered planforms do not occur. The initial version of CAP-TSD was six times faster than XTRAN3S (version 1.5) on a per time step basis, and since larger time steps could be taken with CAP-TSD, the new capability produced a computational cost savings of approximately two orders of magnitude.⁶ Furthermore, the CAP-TSD code was developed to treat complete aircraft configurations involving multiple lifting surfaces and multiple bodies such as a fuselage. For example, steady and unsteady transonic calculations were performed using CAP-TSD for the F-16 fighter configuration, which was modeled using four lifting surfaces and two bodies.⁶

Over the years research has been conducted by several organizations including the NASA Langley Research Center to assess the general applicability and accuracy of the small perturbation computer programs in numerous unsteady aerodynamic and aeroelastic applications.⁷⁻¹⁰ These applications have revealed various inaccuracies inherent in the underlying small perturbation theories¹¹ including accuracy limitations near the leading edge and wing tip regions,⁷ shock waves that are inaccurately captured in terms of strength or location,¹⁰ and convergence difficulties attributable to certain computational and mathematical complications such as non-uniqueness of the potential flow equation.^{12,13} Consequently, the author conducted research over the last decade to examine the applicability of the small perturbation concept in general and the accuracy of the CAP-TSD code in specific. The author is the principal developer^{5,14,15} of the CAP-TSD code and is therefore in a unique position to make a critical assessment of the methodology contained therein. The objective of the present effort was to identify the sources of the inaccuracies and determine if improvements could be made to alleviate or eliminate them.

The first step in this effort was to determine the accuracy of the CAP-TSD code for the inviscid flow about a basic two-dimensional airfoil configuration. The NACA 0012

airfoil was selected because the shape is analytically defined and calculations could be performed first for non-lifting cases to eliminate any potential inaccuracies due to the modeling of the trailing wake. There are standard cases¹⁶ for the NACA 0012 airfoil whereby accurate solutions, obtained using computer codes that solve the full potential, Euler, and Navier-Stokes equations, have been published. These solutions were used for comparison purposes. The inviscid CAP-TSD calculations differed substantially from the accepted full potential and Euler solutions. For transonic cases, the shockwaves were poorly predicted in terms of strength and location. This was the case even when the shock-induced entropy and vorticity effects¹⁵ were included in the CAP-TSD calculation and comparisons were made with accepted Euler solutions. For subcritical cases where such effects are unnecessary, the pressure levels were poorly predicted in comparison with accepted full-potential solutions even when the complete pressure coefficient formula was selected within the code. This suggests that the underlying transonic small perturbation (TSP) potential flow theory in CAP-TSD is deficient in some respect, and the inclusion of entropy, vorticity, and even viscous effects is then only an academic exercise.

Consequently, a new advanced small perturbation (ASP) potential flow theory¹⁷ was developed by first determining the essential elements required to produce solutions as accurate as a full potential code with the small perturbation approach on a Cartesian grid. This level of accuracy was obtained by using a higher-order streamwise mass flux and a mass conserving surface boundary condition. Subsequent applications with these improvements showed very good agreement with all of the full potential cases considered. This was true even for cases at freestream Mach numbers as low as 0.3 and angles of attack as high as ten degrees, conditions that are normally considered to be outside the range of applicability of classical small perturbation theories.

The ASP theory was further developed to produce Euler-like solutions by incorporating a mass conserving calculation of the entropy jump across of the shock wave based on the higher-order streamwise mass flux of the ASP theory.¹⁷ Second-order terms in the trailing wake boundary condition were found to be not insignificant for lifting cases and were also incorporated. Subsequent calculations with these modifications showed very good agreement with all of the Euler cases that were considered. This was true for cases involving strong shock waves and multiple shock waves of disparate strengths.

Viscous effects were incorporated within the ASP formulation by coupling an integral boundary layer procedure to the outer inviscid flow calculation.¹⁷ The dissipation integral method¹⁸⁻²⁰ was implemented to model attached or shock-induced separated boundary layers. The capability involves solving the unsteady boundary layer and lag equations simultaneously with the outer potential flow solution, so that no interaction law coupling the inner and outer solutions is required. The combined solution procedure involves an implicit block tridiagonal inversion for all of the cells along the surface and its trailing wake. Exact formulas for edge quantities and exact boundary conditions are used along surfaces and wakes. Smoothing of edge quantities and limiters are not required for stability, and no arbitrary or free parameters are necessary to tune the

procedure. Calculations performed with the complete ASP capability including entropy, vorticity, and viscous effects showed good agreement with experimental data.

The ASP theory was shown to be mathematically more appropriate and computationally more accurate than the classical TSP theories.¹⁷ Consequently, the ASP theory has been used as the basis for a new computer program called ASP3D (Advanced Small Perturbation – 3D), which involves either AF1- or AF2-type approximate factorization algorithms and a FAS (full approximation scheme) multigrid procedure for the solution of the ASP potential flow equation. The ASP3D code can treat aircraft configurations involving multiple lifting surfaces with leading and trailing edge control surfaces and a fuselage. The purpose of the paper is to introduce the ASP3D computer program by reporting the underlying formulation with detailed descriptions of the governing equation, entropy and vorticity effects, viscous effects, surface and trailing wake boundary conditions, shock capturing options, solution algorithm, residual calculation, multigrid implementation, numerical accuracy, numerical stability, and structural equations of motion. In many cases comparisons with the CAP-TSD formulation are made to contrast the differences between the two computer programs. Results obtained using the ASP3D code are also presented for several cases to demonstrate various options and the general utility of the new program.

Overview of the CAP-TSD Code

The CAP-TSD code was developed by a team of researchers within the Unsteady Aerodynamics Branch at NASA Langley Research Center in 1986-1987. The team consisted of John T. Batina (principal developer), Robert M. Bennett (program manager), Samuel R. Bland, David A. Seidel, and Robert W. Neely. The code was based on 1970's and early 1980's potential flow CFD technology, involving an AF1-type approximate factorization finite-difference algorithm including entropy and vorticity effects. James T. Howlett added viscous effects later,^{21,22} based on the Green's lag-entrainment approach,²³ with further viscous modifications implemented by John W. Edwards.²⁴

The CAP-TSD code was released in February of 1988 and was quickly requested by numerous groups within industry, academia, and other government research laboratories. Early users of CAP-TSD included McDonnell-Douglas, Douglas Aircraft (Long Beach), General Dynamics, Rockwell International, Boeing, Northrop, and Lockheed. The code was applied to numerous cases and the efforts proved invaluable for code debugging and further development. The introduction of faster computers with more memory, however, proved to be a "double-edged sword." It allowed the use of finer grids (more grid points), but revealed various inaccuracies and inefficiencies largely attributable to the methodology embodied within the code.

Overview of the ASP3D Code

The ASP3D code was developed over the last decade by the author during his employment as a Senior Research Scientist within the Configuration Aerodynamics Branch at NASA Langley Research Center. The development of the code resulted from a

research study to determine the specific causes of the inaccuracies and inefficiencies inherent to CAP-TSD. That work revealed that the underlying small perturbation potential flow theory in CAP-TSD was deficient in several respects. An advanced small perturbation theory was developed to alleviate or eliminate the inaccuracies identified in CAP-TSD.¹⁷ The ASP theory was shown subsequently to be mathematically more appropriate and computationally more accurate than the classical TSP theories such as that embodied within CAP-TSD. Consequently, the ASP theory was used as the basis for the new ASP3D program, which in contrast with CAP-TSD, involves modern computational fluid dynamics concepts such as a finite volume spatial discretization, sharper shock capture, dissipation integral boundary layer modeling, alternative AF2-type approximate factorization algorithm, and FAS (full approximation scheme) multigrid solution convergence acceleration.

Advanced Small Perturbation Theory

In this section the essential elements of the ASP theory¹⁷ are described as background material, along with mathematical and computational comparisons with various TSP theories, selected alternative results, and experimental data.

- Governing Equation

It is first instructive to examine in detail the general small perturbation equation that is presented here in two dimensions for simplicity as

$$\frac{\partial f_0}{\partial t} + \frac{\partial f_1}{\partial x} + \frac{\partial f_3}{\partial z} = 0$$

The fluxes are defined as

$$f_0 = -A\phi_t - B\phi_x$$

$$f_1 = C + D\phi_x + E\phi_x^2 + F\phi_x^3$$

$$f_3 = \phi_z$$

The constant coefficients are defined as

$$A = M_\infty^2 \qquad B = 2M_\infty^2 \qquad C = 1 \qquad D = 1 - M_\infty^2$$

The remaining coefficients E and F are somewhat arbitrary depending upon the assumptions used in deriving the governing equation. When the equation is derived to match a small perturbation form of the shock condition the so-called *NASA Ames coefficients* result, given by

$$E = -\frac{1}{2}(\gamma + 1)M_\infty^2 \qquad F = 0$$

The so-called *NLR coefficients*, derived by taking a small perturbation approximation of a series expansion of the density, are defined by

$$E = -\frac{1}{2}[3 - (2 - \gamma)M_\infty^2]M_\infty^2 \quad F = 0$$

And the newly defined *ASP (Advanced Small Perturbation) coefficients*, derived by matching the exact sonic and stagnation conditions, are defined by

$$E = -\frac{1}{2}(\gamma + 1)M_\infty^2 \quad F = -\frac{1}{6}(\gamma + 1)M_\infty^2$$

An alternative streamwise flux f_1 was derived by an asymptotic expansion of the Euler equations by Williams.²⁵ This alternative flux is defined by

$$f_1 = (\gamma + 1)M_\infty^2 [1 + (\phi_x)_{sonic}] \left(VV_{sonic} - \frac{V^2}{2} \right)$$

where

$$V = \frac{\phi_x}{1 + \frac{\phi_x}{2 + (\phi_x)_{sonic}}} \quad \text{and} \quad V_{sonic} = \frac{[1 + (\phi_x)_{sonic}]^2 - 1}{2[1 + (\phi_x)_{sonic}]}$$

By design, Williams' flux has the exact sonic velocity and the correct shock jump condition. Therefore it is a good alternative to the TSP mass flux evaluated using the NLR or NASA Ames coefficients for cases involving shock waves. Consequently, Williams' flux is the default streamwise flux used within the CAP-TSD code. However as discussed below, Williams' flux is inaccurate near stagnation, which results in numerical difficulties near the leading edge.

To compare and contrast the various small perturbation formulations graphically, Figure 1 shows flux quantities as functions of streamwise velocity ($u = 1 + \phi_x$) for the ASP theory, the TSP theories with NLR and NASA Ames coefficients, and Williams' asymptotic expansion. The flux quantities were computed at a freestream Mach number of $M_\infty = 0.72$. Figure 1(a) shows the streamwise mass flux f_1 and Figure 1(b) shows the derivative of the streamwise mass flux $\partial f_1 / \partial \phi_x$. The derivative of the mass flux is important because it is related to the product of the local wave speeds given by

$$\frac{\partial f_1}{\partial \phi_x} = M_\infty^2 (a - u)(a + u) = M_\infty^2 (a^2 - u^2)$$

where a is the speed of sound. The sign of the derivative $\partial f_1 / \partial \phi_x$ indicates whether the local flow is subsonic (positive because $|u| < a$) or supersonic (negative because $|u| > a$). And hence, the sonic point is the velocity by which $\partial f_1 / \partial \phi_x = 0$.

At $u = 1$, which corresponds to the undisturbed or freestream flow of $\phi_x = 0$, the four small perturbation theories have identical values of the flux and its derivative, as expected. Near $u = 1$, which corresponds to small perturbations $|\phi_x| \ll 1$ where classical small perturbation theory is mathematically valid, the flux quantities are similar between the various theories with the greatest similarity occurring between the TSP theories with the NLR and NASA Ames coefficients. For larger perturbations corresponding to approximately $u < 0.8$ or $u > 1.2$, there are larger differences between the flux quantities of the small perturbation theories, especially for slower speeds $u \rightarrow 0$ corresponding to stagnation, and even more so for reverse flows $u < 0$.

Of the various small perturbation formulations shown in Figure 1, the ASP theory is the only formulation that has the correct form for the mass flux and its derivative across the entire velocity range. **The correct form for the mass flux** is an asymmetric function about $u = 0$ (stagnation) because the mass flux physically should have a similar magnitude but a negative sign for reverse flow as shown by the ASP theory curve in Figure 1(a). The TSP theories with the NLR or NASA Ames coefficients and Williams' formulation do not satisfy the above property, and hence, they are not applicable for cases involving reverse flow. Furthermore, they are inaccurate near stagnation, because those formulations have negative values for the mass flux at small positive values of u , with the greatest deviation from zero given by the Williams' flux.

The correct form for the derivative of the mass flux is a symmetric function about $u = 0$ (stagnation), as shown by the ASP curve in Figure 1(b). This is because the derivative should be positive for subsonic flow, corresponding to $|u| < a$, which includes reverse subsonic flow, and the derivative should be negative for supersonic flow, corresponding to $|u| > a$, including reverse supersonic flow. When the derivative of the mass flux is positive, the governing equation is of elliptic type, correctly describing subsonic flow. When the derivative is negative, the governing equation is hyperbolic, physically corresponding to supersonic flow. For the ASP theory this is true, independent of the direction of the flow. The other theories not only have the wrong form but they are inaccurate near stagnation and inaccurate for stronger supersonic flows. In fact the worst formulation in this regard is the Williams' flux, which explains why applications of CAP-TSD using the Williams' flux (default formulation) encounter convergence difficulties, especially on finer meshes.

Figure 2 shows an expanded view of the derivative of the mass flux as a function of the streamwise velocity for the $M_\infty = 0.72$ case of Figure 1 to illustrate the differences in sonic points predicted by the various small perturbation formulations. The sonic point is the velocity by which $\partial f_1 / \partial \phi_x = 0$. As discussed before, the ASP and Williams' flux formulations possess the exact sonic point by design as shown in Figure 2. The TSP theories with the NLR and NASA Ames coefficients predict values of the sonic velocity that are too high, with the highest value of the sonic velocity corresponding to the NASA Ames formulation. The TSP theory with the NASA Ames coefficients will consequently produce transonic solutions with the smallest supersonic zones and weakest shock waves in comparison with the other formulations. This is due to the flow remaining subsonic

until the local velocity exceeds the higher predicted value of the sonic velocity. The TSP theory with the NLR coefficients will generally produce transonic solutions with larger supersonic regions and stronger shocks in comparison with the TSP-NASA Ames theory, but the shock waves will not be as strong as those predicted correctly by either the ASP or Williams' formulations.

Flow calculations were performed to demonstrate the effects of the streamwise flux on the pressure coefficient distribution for a typical transonic case. These calculations are shown in Figure 3 from Batina¹⁷ for the ASP theory and the TSP theories with the NLR and NASA Ames coefficients. Results were not obtained using Williams' asymptotic flux because of numerical difficulties in the leading edge region that inhibited convergence. The case considered is that of the NACA 64A410 airfoil at an angle of attack of $\alpha = 0^\circ$ and $M_\infty = 0.72$, the same freestream Mach number used in Figures 1 and 2. As shown in Figure 3, the solution obtained using the TSP theory with the NASA Ames coefficients indicates that there is a smaller supersonic region and a weaker shock wave on the upper surface of the airfoil in comparison with the other two solutions as predicted by the theoretical analysis discussed earlier. The solution obtained using the TSP theory with the NLR coefficients has a slightly stronger shock wave, but the strongest shock wave is predicted by the ASP theory, also consistent with the theoretical analysis. The ASP pressure distribution is in very good agreement with a conservative full-potential (FP) calculation shown in Figure 4, reported by Jameson²⁶ for this case. A comparison of the ASP (Figure 3) and FP (Figure 4) pressure distributions demonstrates that the strength and position of the shock wave near approximately 63% chord on the upper surface of the airfoil are predicted accurately by the ASP formulation. Hence, the ASP mass flux is an essential element in obtaining FP-like solutions with a small perturbation formulation.

- Inviscid Irrotational Surface Boundary Conditions

The CAP-TSD code and other small perturbation codes typically use the lowest order classical small disturbance boundary condition to impose surface tangency given by

$$\phi_z = b_x - \alpha$$

where

$$b(x) - x\alpha = 0$$

represents the surface. With this boundary condition the velocity and mass flux vectors do not generally coincide, and neither vector is tangent to the surface.

Use of this boundary condition generally results in shock waves that are too weak and located forward in comparison with full potential results. Furthermore such solutions can be shown to be dependent on the local mesh density. Specifically, if the mesh is changed, such as using a finer or coarser mesh, the shock strength and location will be materially changed.

Requiring that the velocity vector be tangent to the surface results in

$$\phi_z = (1 + \phi_x)(b_x - \alpha)$$

which may be referred to as a “full potential” surface boundary condition, similar to what is imposed in full potential codes albeit on a body-fitted mesh. However this boundary condition is not consistent with the governing equation and generally leads to solutions involving shock waves that are too strong and located too far aft.

Requiring that the mass flux be tangent to the surface results in

$$\phi_z = f_1(b_x - \alpha)$$

where f_1 is the streamwise mass flux. This boundary condition is consistent with the governing equation (mass consistent) and its use produces solutions that are relatively mesh independent, meaning that the local mesh density does not appreciably affect the shock strength and location.

The ASP capability uses a higher-order mass conserving boundary condition to impose surface tangency given by

$$\phi_z = \frac{f_1}{g}(b_x - \alpha)$$

where

$$g = 1 + H\phi_x + \frac{H}{2}\phi_x^2 \quad \text{and} \quad H = -(\gamma - 1)M_\infty^2$$

Note that $g = M_\infty^2 a^2$. This boundary condition was found to be more accurate than the simpler one above, although the inclusion of the complete vertical flux g in the governing equation had a negligible effect on the solution.

The effects of surface boundary condition (BC) on pressure coefficient distribution are shown in Figure 5 for three forms of the surface BC including the lowest order form used in CAP-TSD labeled “slopes”, the full potential BC where the slopes are scaled by the streamwise velocity ($u = 1 + \phi_x$), and the mass consistent BC used in the ASP formulation. The calculations were performed for the same NACA 64A410 case presented in Figure 3 and the ASP streamwise flux was used throughout. Figure 5 shows that the use of the lowest order “slopes” BC produced a solution that has a shock wave that is too weak located forward of the correct position. Although not shown here, solutions obtained with the lowest order BC tend to be mesh-dependent because the BC is not mass conserving. Use of the FP BC produces a shock wave that is too strong and located too far aft. In contrast, the mass consistent BC (with the ASP mass flux) leads to the correct shock strength and location, as compared with the Jameson²⁶ full potential solution of Figure 6 (same as Figure 4; repeated for direct comparison). Hence, the mass consistent surface boundary condition is an essential element in obtaining FP-like solutions with a small perturbation formulation.

- Entropy, Vorticity, and Viscous Effects

Shock-generated entropy and vorticity effects are incorporated within the ASP theory using mass consistent modeling as described in detail below.¹⁷ In Reference 17 a mass conserving calculation of the entropy change across the shock wave and the inclusion of the second-order terms in the trailing wake were shown to be essential elements in producing Euler-like accuracy with the ASP formulation.

To demonstrate the applicability of **the ASP theory with entropy and vorticity** modeling, calculations were performed for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha = 1.25^\circ$. The ASP pressure coefficient distribution is presented in Figure 7 and an Euler solution computed by Anderson, et al.²⁷ with the CFL3D code is presented in Figure 8. The case involves a strong shock wave on the upper surface of the airfoil at approximately 64% chord and a weak shock wave on the lower surface near 34% chord. The difficulty of this case involves the difference in strength of the two shock waves. Comparisons of the two pressure distributions indicate that the ASP formulation including entropy and vorticity accurately predicts the strength and location of both of the shock waves, even though the shocks have disparate strengths.

Viscous effects are included within the ASP theory through a simultaneous implicit solution of the integral boundary layer and lag equations with the outer ASP potential flow as described in detail below.¹⁷ The resulting capability does not require the use of an interaction law because the equations are solved simultaneously. This is a stable approach that also does not require any smoothing or limiters. Turbulent closure is through the use of the dissipation integral relations of Drela,²⁰ which is applicable to attached and shock-induced separated flows.

Two cases are presented to demonstrate **the ASP viscous capability** corresponding to an attached boundary layer and a shock-induced flow separation. The attached flow case²⁸ is the RAE 2822 airfoil at $M_\infty = 0.676$, $\alpha = -2.25^\circ$, and $Re = 5.7 \times 10^6$. The calculations were performed using a CFL number of thirty for five hundred time steps, and the resulting ASP pressure coefficient distribution is compared with the experimental pressure data in Figure 9. The comparison shows very good agreement between the ASP pressures and the experimental data, thus demonstrating the accuracy of the ASP viscous capability.

The second case corresponds to separated flow on the upper surface of the NACA 0012 airfoil at $M_\infty = 0.775$, $\alpha = 2.05^\circ$, and $Re = 10^7$. The calculations were performed with a CFL number of fifteen for a total of one thousand time steps. The CFL number for the separated flow case was half of that used in the attached flow case, although it is still a relatively large value. The resulting ASP pressure coefficient distribution is shown in Figure 10, which is in good agreement with the experimental data,²⁹ especially in the strength and location of the upper surface shock wave. The shock induced separation along the upper surface of the airfoil ranges from the foot of the shock near 53% chord to the reattachment point at approximately 67% chord.

The ASP3D Computer Program

The ASP theory has been used as the basis for a new computer code called ASP3D (Advanced Small Perturbation – 3D), which involves either AF1- or AF2-type approximate factorization algorithms and a FAS (full approximation scheme) multigrid procedure for the solution of the ASP potential flow equation and associated boundary conditions. The ASP3D code can treat aircraft configurations involving multiple lifting surfaces with leading and trailing edge control surfaces, and a fuselage. The new code is described here along with conceptual and theoretical comparisons with the CAP-TSD code. The author is in a unique position as the developer of both computer codes to compare, contrast, and ultimately make conclusions regarding the underlying formulations and utility of each code.

- Governing Equation

The three-dimensional general small perturbation equation may be written in Cartesian coordinates as

$$\frac{\partial f_0}{\partial t} + \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} = 0$$

where the Cartesian fluxes are defined as

$$f_0 = -A\phi_t - B\phi_x$$

$$f_1 = C + D\phi_x + E\phi_x^2 + F\phi_x^3 + G\phi_y^2 + \frac{H}{2}\phi_x\phi_y^2$$

$$f_2 = \left(1 + H\phi_x + \frac{H}{2}\phi_x^2\right)\phi_y + F\phi_y^3$$

$$f_3 = \phi_z$$

with the constants defined as

$$A = M_\infty^2 \quad B = 2M_\infty^2 \quad C = 1 \quad D = 1 - M_\infty^2$$

$$E = -\frac{1}{2}(\gamma + 1)M_\infty^2 \quad F = -\frac{1}{6}(\gamma + 1)M_\infty^2 \quad G = \frac{1}{2}(\gamma - 3)M_\infty^2 \quad H = -(\gamma - 1)M_\infty^2$$

Figure 11 shows flux quantities as functions of streamwise velocity ($u = 1 + \phi_x$) for the ASP theory computed for freestream Mach numbers ranging from $M_\infty = 0.0$ to 1.2. Figure 11(a) shows the streamwise part of the mass flux f_1 and Figure 11(b) shows the derivative of the streamwise part of the mass flux $\partial f_1 / \partial \phi_x$. Figure 11(a) emphasizes that the ASP mass flux is an asymmetric function about $u = 0$ (stagnation) and the $M_\infty = 0.0$ result is a line with a slope of unity. If desired the positive offset in f_1 can be removed by replacing the constant $C = 1$ in the ASP mass flux by the constant $C = D - E + F$,

although this modification was not investigated. Of course the derivative is independent of the constant C. Figure 11(b) emphasizes that the derivative of the ASP flux is a symmetric function about $u = 0$ (stagnation) at any freestream Mach number.

The ASP3D code though solves the governing equation written in computational coordinates as

$$\frac{\partial \hat{f}_0}{\partial t} + \frac{\partial \hat{f}_1}{\partial \xi} + \frac{\partial \hat{f}_2}{\partial \eta} + \frac{\partial \hat{f}_3}{\partial \zeta} = 0$$

where the computational fluxes are defined as

$$\begin{aligned}\hat{f}_0 &= -\frac{A}{J} \phi_t - \frac{B}{J} \xi_x \phi_\xi \\ \hat{f}_1 &= \frac{|\nabla \xi|}{J} \left[\frac{\xi_x}{|\nabla \xi|} (f_1) + \frac{\xi_y}{|\nabla \xi|} (f_2) \right] \\ \hat{f}_2 &= \frac{|\nabla \eta|}{J} \left[\frac{\eta_y}{|\nabla \eta|} (f_2) \right] \\ \hat{f}_3 &= \frac{|\nabla \zeta|}{J} \left[\frac{\zeta_z}{|\nabla \zeta|} (f_3) \right]\end{aligned}$$

The various geometric quantities used in these equations are defined as

$$\begin{aligned}\frac{1}{J} &: \text{volume of the cell} \\ \frac{|\nabla \xi|}{J}, \frac{|\nabla \eta|}{J}, \frac{|\nabla \zeta|}{J} &: \text{areas of the faces in the } \xi, \eta, \zeta \text{ directions} \\ \frac{\xi_x}{|\nabla \xi|}, \frac{\xi_y}{|\nabla \xi|} &: \text{direction cosines in the } \xi\text{-direction} \\ \frac{\eta_y}{|\nabla \eta|} &: \text{direction cosines in the } \eta\text{-direction} \\ \frac{\zeta_z}{|\nabla \zeta|} &: \text{direction cosines in the } \zeta\text{-direction}\end{aligned}$$

With a typical cell defined as shown in Figure 12, the above geometric quantities are defined by the following exact formulas. For example, the cell volume is defined by

$$Volume \equiv \frac{1}{J} = \frac{1}{2}(\Delta x_i + \Delta x_o)\Delta y\Delta z$$

The cell interface areas are defined in the ξ -direction as

$$\left(\frac{|\nabla \xi|}{J}\right)_{i+1/2} = \Delta z \sqrt{\Delta x_d^2 + \Delta y^2} \quad \text{and} \quad \left(\frac{|\nabla \xi|}{J}\right)_{i-1/2} = \Delta z \sqrt{\Delta x_u^2 + \Delta y^2}$$

in the η -direction as

$$\left(\frac{|\nabla \eta|}{J}\right)_{j+1/2} = \Delta x_o \Delta z \quad \text{and} \quad \left(\frac{|\nabla \eta|}{J}\right)_{j-1/2} = \Delta x_i \Delta z$$

and in the ζ -direction as

$$\left(\frac{|\nabla \zeta|}{J}\right)_{k\pm 1/2} = \frac{1}{2}(\Delta x_i + \Delta x_o)\Delta y$$

The direction cosines in the ξ -direction are

$$\left(\frac{\xi_x}{|\nabla \xi|}\right)_{i+1/2} = \frac{\Delta y}{\sqrt{\Delta x_d^2 + \Delta y^2}}, \quad \left(\frac{\xi_x}{|\nabla \xi|}\right)_{i-1/2} = \frac{\Delta y}{\sqrt{\Delta x_u^2 + \Delta y^2}}$$

and

$$\left(\frac{\xi_y}{|\nabla \xi|}\right)_{i+1/2} = \frac{-\Delta x_d}{\sqrt{\Delta x_d^2 + \Delta y^2}}, \quad \left(\frac{\xi_y}{|\nabla \xi|}\right)_{i-1/2} = \frac{-\Delta x_u}{\sqrt{\Delta x_u^2 + \Delta y^2}}$$

in the η -direction are

$$\left(\frac{\eta_y}{|\nabla \eta|}\right)_{j\pm 1/2} = 1$$

and in the ζ -direction are

$$\left(\frac{\xi_z}{|\nabla \xi|}\right)_{k\pm 1/2} = 1$$

- Entropy and Vorticity Effects

Shock-generated entropy effects are incorporated within small perturbation codes such as CAP-TSD by first using the Prandtl relation³⁰ (shock jump condition) to determine the velocity downstream of the shock wave from the upstream and sonic velocities. The upstream and downstream velocities are then used in the Rankine-Hugoniot (R-H) shock relation³⁰ to determine the change in entropy across the shock wave.³¹ The resulting change in entropy is subsequently used in a Clebsch formulation³² to determine the vorticity downstream of the shock wave.^{33,34} The vorticity modifies the

calculation of the velocity field in the downstream region. This is a common procedure that has been used in various computer codes especially at the full-potential equation level. However, the approach as applied to the small perturbation equation, with any of the fluxes defined above including that of Williams,²⁵ does not conserve mass.

In contrast, the approach developed for the ASP3D program conserves mass by using a ratio of the streamwise fluxes evaluated using the upstream and downstream velocities. Specifically, the downstream perturbation velocity is first computed using the Prandtl relation (shock jump relation) given by

$$(\phi_x)_2 = \frac{[1 + (\phi_x)_{sonic}]^2}{1 + (\phi_x)_1} - 1$$

where the subscripts 1 and 2 represent stations that are upstream and downstream of the shock wave, respectively. The change in entropy across the shock is then computed using

$$\Delta s = (\gamma - 1) \left\{ 1 - f_1 [(\phi_x)_1] / f_1 [(\phi_x)_2] \right\}$$

For steady flows the entropy is held constant along gridlines downstream of shock waves. For unsteady flows the entropy is convected downstream using

$$\frac{\partial}{\partial t} \Delta s + \frac{\partial}{\partial x} \Delta s = 0$$

The fluxes downstream of the shock are subsequently modified according to

$$(f_1)_{nonisentropic} = \left[1 - \frac{\Delta s}{(\gamma - 1)} \right] (f_1)_{isentropic}$$

which conserves mass across the shock wave by design.

Similar to the CAP-TSD code, the ASP3D code uses a Clebsch formulation³² to compute the shock-generated vorticity. In brief, the streamwise velocities downstream of shocks are computed using

$$(\phi_x)_{rotational} = (\phi_x)_{irrotational} - \frac{\Delta s}{\gamma(\gamma - 1)M_\infty^2}$$

In the CAP-TSD code, when entropy and vorticity effects are included in the solution procedure no changes are made to the wake modeling because the first order term due to vorticity exactly cancels the first-order entropy term.¹⁵ However, the second-order terms were found to be not insignificant, and thus, they were incorporated into the ASP3D solution procedure. Hence when entropy and vorticity are included, the circulation is convected using

$$\Gamma_t + \Gamma_x = \frac{(\gamma - 1)M_\infty^2 + 1}{\gamma(\gamma + 1)M_\infty^2} \left\{ [(\Delta s)(\phi_x)]_{upper} - [(\Delta s)(\phi_x)]_{lower} \right\}$$

$$- \frac{1}{2} \left[1 - \frac{(\Delta s)_{upper}}{(\gamma - 1)} \right] \left(1 - M_\infty^2 \right) (\phi_x^2)_{upper}$$

$$+ \frac{1}{2} \left[1 - \frac{(\Delta s)_{lower}}{(\gamma - 1)} \right] \left(1 - M_\infty^2 \right) (\phi_x^2)_{lower}$$

where the subscripts “*upper*” and “*lower*” correspond to the values on the upper and lower surfaces of the trailing wake.

- Viscous Effects

As mentioned above, Howlett²¹ incorporated viscous effects into the CAP-TSD code. The methodology is the Green’s lag entrainment approach²³ for solving the steady boundary layer equations (even though the intended applications are for unsteady flows). Howlett originally developed the software within XTRAN3S, and the capability was initially applicable only for attached flows, but later the capability was extended to treat mildly separated flows.²² Edwards²⁴ inherited the viscous capability in CAP-TSD upon the retirement of Howlett from NASA, and the capability was extended by Edwards²⁴ to treat flows with more severe separation including shock-induced separation and naturally periodic flows with separating and reattaching boundary layers.

However, the viscous methodology incorporated into CAP-TSD by Howlett and Edwards is not strictly speaking applicable to separated flows. First, the matrix that needs to be inverted for solution of the viscous equations is ill conditioned as the flow nears separation, because the matrix has a zero determinant at separation onset. This occurs because one of the eigenvalues of the matrix changes sign for separated flow. Specifically, for attached flows, the three eigenvalues of the viscous equations are all positive, as discussed in more detail below, indicating that all of the characteristics are in the downstream or positive direction. For separated flows, one of the eigenvalues is negative, mathematically reflecting the fact that within the boundary layer there is some reverse flow (in the negative streamwise direction).

Similarly, the space-marching integration of the viscous equations of Howlett and Edwards, a multistage Runge-Kutta (R-K) integration, is inappropriate for separated flows because it is unconditionally unstable for such flows. Specifically, for attached flows, when all of the eigenvalues are positive, the integration is conditionally stable, and therefore it is an acceptable albeit inefficient explicit algorithm to integrate the viscous equations. But for separated flows with one negative eigenvalue, the downstream space marching R-K algorithm does not have a mathematical domain of dependence that includes the physical domain of dependence, and hence, the R-K algorithm is

unconditionally unstable. This can be demonstrated easily with a Von Neumann stability analysis of the governing equations.

The method of Howlett²² produced numerical instabilities when used for separated flow cases since it is unconditionally unstable for such flows, until the transition to separation was arbitrarily set to downstream of the shockwave. Then, of course, it makes no sense physically. Edwards²⁴ uses the same unconditionally unstable R-K algorithm, but has added various smoothings, limiters, and filters, including an unnecessary interaction equation involving active control elements to couple the inner viscous solution with the outer inviscid solution, instead of implementing a numerically stable algorithm to integrate the equations. The interaction law involves free parameters that affect the solution accuracy, as demonstrated in Ref. 24, and it can be shown to be an extreme filter that smoothes all of the high frequency diverging waves produced by the unstable R-K integration. This is one of several reasons why the method incorporated within the CAP-TSD code by Edwards never converges.

Furthermore, the turbulent closure relations of the Green's lag entrainment approach have not been shown in a rigorous way to be applicable to separated flows. Melnik³⁵ extended his attached flow Green's lag entrainment method within the GRUMFOIL³⁶ code to treat mildly separated flows, by making some ad hoc modifications to the closure relations. Howlett²² incorporated similar modifications within CAP-TSD. Edwards²⁴ has used these modifications also in his capability wherein the standard Green's turbulent closure relations are used for attached flow, some undefined set of closure relations are used for separated flows, and a linear (and not physical) interpolation is used somewhere in between. Unfortunately, these researchers have not published the detailed specifics on what closure relations they have used for separated flows, and therefore no one can independently verify what they have done.

In contrast, the dissipation integral method developed by Whitfield and coworkers^{18,19} was implemented within the ASP3D code to model attached or separated boundary layers. The dissipation integral approach represents a significant improvement over the Green's lag entrainment method such as that in CAP-TSD because the closure relations were derived by a detailed fitting of velocity profiles obtained through experiments for both attached and separated flows. Hence, the closure relations are continuous functions applicable to either attached or separated flows, and there is no need for any nonphysical interpolation between disparate sets of closure relations. Drela²⁰ has refined the dissipation integral method, wherein the closure relations depend not only upon the local Mach number and shape factor, but also on the momentum thickness Reynolds number.

The ASP3D viscous method involves solving the unsteady boundary layer equations simultaneously with the outer potential flow solution so that no interaction law coupling the inner and outer solutions is required. The combined solution procedure involves an implicit block tridiagonal inversion for all of the cells along the surface and trailing wake. Unlike the CAP-TSD viscous capability,²⁴ the ASP capability uses exact formulas for edge quantities and exact boundary conditions along surfaces and wakes. Smoothing of

edge quantities and limiters are not required for stability, and no arbitrary or free parameters are necessary to tune the procedure.

Specifically in the ASP3D viscous capability, the exact edge quantities are used for the calculation of velocity u_e , density ρ_e , temperature T_e , local Mach number M_e , and the coefficient of viscosity μ_e defined by

$$\begin{aligned}
 u_e &= 1 + \phi_x \\
 \rho_e &= \left[1 - \frac{\gamma-1}{2} M_\infty^2 (2\phi_x + \phi_x^2) \right]^{\frac{1}{\gamma-1}} \\
 T_e &= \left[1 - \frac{\gamma-1}{2} M_\infty^2 (2\phi_x + \phi_x^2) \right] \\
 M_e &= \frac{M_\infty (1 + \phi_x)}{\left[1 - \frac{\gamma-1}{2} M_\infty^2 (2\phi_x + \phi_x^2) \right]^{1/2}} \\
 \mu_e &= (T_e)^{3/2} \frac{1 + (S/T_\infty)}{(T_e) + (S/T_\infty)}
 \end{aligned}$$

where S is Sutherland's coefficient and T_∞ is the freestream temperature. These equations are in contrast with the edge quantities contained within CAP-TSD wherein the density, temperature, and local Mach number relations were linearized as

$$\begin{aligned}
 \rho_e &= 1 - M_\infty^2 \phi_x \\
 T_e &= 1 - (\gamma-1) M_\infty^2 \phi_x \\
 M_e &= M_\infty \left[1 + \left(1 + \frac{\gamma-1}{2} M_\infty^2 \right) \phi_x \right]
 \end{aligned}$$

and the velocity and coefficient of viscosity are defined as above.²²

For the ASP3D capability, the time-dependent integral boundary layer (IBL) and lag equations may be written as a system of equations in the form

$$[A] \frac{\partial}{\partial t} \begin{Bmatrix} \theta \\ H_k \\ c_\tau^{1/2} \end{Bmatrix} + [B] \frac{\partial}{\partial x} \begin{Bmatrix} \theta \\ H_k \\ c_\tau^{1/2} \end{Bmatrix} + [C] \begin{Bmatrix} \phi_{xx} \\ \phi_{xt} \\ \phi_{tt} \end{Bmatrix} + \{S\} = \{0\}$$

where the independent variables are the momentum thickness θ , the incompressible shape parameter H_k , and the square root of the shear stress coefficient c_τ . The matrix $[B]$ has three eigenvalues that indicate the nature of the boundary layer. For attached boundary layers ($H_k < H_0$) all three eigenvalues are positive, as shown in Figure 13, indicating that all of the characteristics are in the downstream or positive direction. The shape parameter H_0 is defined as²⁰

$$\left. \begin{array}{ll} H_0 = 3 + 400/Re_\theta & \text{if } Re_\theta > 400 \\ H_0 = 4 & \text{if } Re_\theta \leq 400 \end{array} \right\} \text{ where } Re_\theta = \frac{\rho_e u_e \theta}{\mu_e} Re$$

At separation onset ($H_k = H_0$) one of the eigenvalues becomes zero. And for separated flows ($H_k > H_0$), one of the eigenvalues becomes negative, mathematically reflecting the fact that within the boundary layer there is some reverse flow (in the negative streamwise direction).

For solution, the integral boundary layer and lag equations are first premultiplied by $[A]^{-1}$ to diagonalize the time term as

$$\frac{\partial}{\partial t} \begin{Bmatrix} \theta \\ H_k \\ c_\tau^{1/2} \end{Bmatrix} + [D] \frac{\partial}{\partial x} \begin{Bmatrix} \theta \\ H_k \\ c_\tau^{1/2} \end{Bmatrix} + [E] \begin{Bmatrix} \phi_{xx} \\ \phi_{xt} \\ \phi_{tt} \end{Bmatrix} + \{F\} = \{0\}$$

The resulting IBL system is then linearized in a simple way and cast into the so-called delta-form for implicit solution as

$$\Delta x \begin{Bmatrix} \Delta \theta \\ \Delta H_k \\ \Delta c_\tau^{1/2} \end{Bmatrix} + \Delta t [G] \begin{Bmatrix} \Delta \theta \\ \Delta H_k \\ \Delta c_\tau^{1/2} \end{Bmatrix} = -\Delta x \Delta t \{R\}$$

where the viscous residual is defined as

$$\{R\} = [D] \frac{\partial}{\partial x} \begin{Bmatrix} \theta \\ H_k \\ c_\tau^{1/2} \end{Bmatrix} + [E] \begin{Bmatrix} \phi_{xx} \\ \phi_{xt} \\ \phi_{tt} \end{Bmatrix} + \{F\}$$

The system of equations is directly coupled with the ASP equation for the outer potential flow for simultaneous implicit solution using a block tridiagonal matrix inversion procedure. This is done only for the cells adjacent to the upper and lower sides of the lifting surfaces and their wakes where the boundary layer is located. All other cells do not involve the boundary layer, and hence, only the ASP equation is solved in those cells. Dissipation integral relations are used for turbulent closure, similar to those reported by Drela.²⁰

- **Surface Boundary Condition**

In CAP-TSD the lowest order linear surface tangency condition is used including entropy, vorticity, and viscous effects. In ASP3D the complete mass-consistent surface boundary condition including entropy, vorticity, and viscous effects is given by

$$\phi_z = \frac{f}{g}(b_x - \alpha) + \phi_y b_y \pm \frac{1}{g}(f\delta^*)_x + b_t + \delta_t^*$$

where

$$f = C + D(\phi_x)_{rotational} + E(\phi_x^2)_{rotational} + F(\phi_x^3)_{rotational} - A\phi_y^2 + \frac{H}{2}\phi_y^2(\phi_x)_{rotational}$$

and

$$g = 1 + H(\phi_x)_{rotational} + \frac{H}{2}(\phi_x^2)_{rotational} + \frac{H}{2}(\phi_y^2)$$

The surface is represented by

$$b(x, y, t) \pm \delta^*(x, t) - x\alpha = 0$$

where the \pm is for the upper and lower surfaces, respectively, since the boundary layer displacement thickness δ^* is defined as a positive quantity on both surfaces.

Note that the spanwise slopes b_y are included in the complete surface boundary condition for three-dimensional applications, which may be important for swept wings near the leading edge and also in the wing tip region. Furthermore, the streamwise and spanwise slopes are computed internal to ASP3D from the ordinates of the wing geometry by simple second-order accurate central finite differences, effectively the same as is done in advanced codes with body fitted meshes. This eliminates the differentiation of the spline of the wing geometry normally performed as a preprocessing step to determine the airfoil slopes. It thus makes the input to the ASP3D code simpler than most other small perturbation codes such as CAP-TSD.

- **Wake Boundary Condition**

Viscous effects are included in the wake modeling in ASP3D through the boundary condition

$$\Delta\phi_z = \pm \frac{1}{g}(f\delta^*)_x + \delta_t^*$$

across the wake where f and g were defined previously. The boundary condition is incorporated numerically using

$$(\phi_z)_{upper} = \frac{\phi_k - (\phi_{k-1} + \Gamma)}{z_k - z_{k-1}} + \frac{1}{2} \left[\frac{1}{g}(f\delta^*)_x + \delta_t^* \right]$$

and

$$(\phi_z)_{lower} = \frac{(\phi_k - \Gamma) - \phi_{k-1}}{z_k - z_{k-1}} - \frac{1}{2} \left[\frac{1}{g} (f\delta^*)_x + \delta_t^* \right]$$

- Shock Capturing

The AF1 algorithm in the CAP-TSD code uses the Engquist-Osher³⁷ entropy-satisfying type-dependent switching to capture shock waves and correctly model supersonic regions of the flow including sonic lines. Shock waves are captured usually with two interior states and the solutions are first-order accurate at supersonic points. The second-order-accurate capability within the code is not correct and therefore should not be used.

The AF1 and AF2 algorithms of the ASP3D code use one of three approaches to model supersonic regions of the flow including: 1) Godunov,³⁸ 2) Engquist-Osher³⁷ (E-O), and 3) Murman-Cole³⁹ (M-C) type-dependent switches. The first two switches satisfy an entropy inequality by design. This means that their use precludes the possibility of computing nonphysical expansion shocks that can occur with the Murman-Cole switch, especially in two-dimensional applications. With Godunov³⁸ switching, shocks are captured sharply, usually with only one interior state. Therefore, the Godunov switch is preferable over the E-O and M-C switches, and it is the default switch within the ASP3D code. The E-O switch is available for comparisons with CAP-TSD, and the M-C switch was included for completeness and historical reasons.

Specifically, the solution of the governing equation involves a difference of biased fluxes given by

$$\Delta f = \tilde{f}_{i+1/2} - \tilde{f}_{i-1/2}$$

The biased flux in the *Godunov*³⁸ *switch* is defined as

$$\tilde{f}_{i+1/2} = f_{i+1/2} - \left[\varepsilon_{i+1/2} + (1 - \varepsilon_{i+1/2}) \varepsilon_i \right] (f_{i+1/2} - f_{sonic}) + \varepsilon_{i-1/2} \varepsilon_i (f_{i-1/2} - f_{sonic})$$

where the sonic value of the flux is defined as

$$f_{sonic} = C + D(\phi_x)_{sonic} + E(\phi_x^2)_{sonic} + F(\phi_x^3)_{sonic}$$

and the epsilons are either zero or one, depending upon whether the cell i or cell interfaces $i+1/2$ and $i-1/2$, are subsonic or supersonic, respectively. The difference of biased fluxes then becomes

$$\Delta f = \left[f_{i+1/2} - f_{i-1/2} \right] - \left[\varepsilon_{i+1/2} + (1 - \varepsilon_{i+1/2}) \varepsilon_i \right] \times \left[f_{i+1/2} - f_{sonic} \right]$$

$$\begin{aligned}
& + \left[\varepsilon_{i-1/2} + (1 - \varepsilon_{i-1/2}) \varepsilon_{i-1} + \varepsilon_{i-1/2} \varepsilon_i \right] \times \left[f_{i-1/2} - f_{sonic} \right] \\
& - \left[\varepsilon_{i-3/2} \varepsilon_{i-1} \right] \times \left[f_{i-3/2} - f_{sonic} \right]
\end{aligned}$$

The Godunov switch satisfies an entropy inequality to eliminate expansion shocks, as suggested by the inclusion of the sonic flux. The Godunov approach samples the velocity field and the shock speed to determine the biased flux, which results in a sharp capture of shock waves with typically only one interior state in inviscid calculations.

The biased flux in the *Engquist-Osher*³⁷ *switch* is defined by

$$\tilde{f}_{i+1/2} = f_{i+1/2} - \varepsilon_{i+1/2} (f_{i+1/2} - f_{sonic}) + \varepsilon_{i-1/2} (f_{i-1/2} - f_{sonic})$$

where the sonic flux and epsilons are defined as above. The difference of biased fluxes then becomes

$$\begin{aligned}
\Delta f &= \left[f_{i+1/2} - f_{i-1/2} \right] \\
& - \varepsilon_{i+1/2} \times \left[f_{i+1/2} - f_{sonic} \right] \\
& + 2\varepsilon_{i-1/2} \times \left[f_{i-1/2} - f_{sonic} \right] \\
& - \varepsilon_{i-3/2} \times \left[f_{i-3/2} - f_{sonic} \right]
\end{aligned}$$

The E-O switch satisfies an entropy inequality as suggested by the inclusion of the sonic flux. The E-O approach samples the velocity field only (and not the shock speed) and consequently captures shock waves with two interior states in contrast with the sharper shocks of the Godunov approach.

The biased flux in the *Murman-Cole*³⁹ *switch* is defined by

$$\tilde{f}_{i+1/2} = f_{i+1/2} - \varepsilon_i f_{i+1/2} + \varepsilon_{i-1} f_{i-1/2}$$

where the epsilons are defined as above. The difference of biased fluxes then becomes

$$\begin{aligned}
\Delta f &= \left[f_{i+1/2} - f_{i-1/2} \right] \\
& - \varepsilon_i \times \left[f_{i+1/2} \right] \\
& + (\varepsilon_i + \varepsilon_{i-1}) \times \left[f_{i-1/2} \right] \\
& - \varepsilon_{i-1} \times \left[f_{i-3/2} \right]
\end{aligned}$$

The M-C switch does not satisfy an entropy inequality since the biased flux does not use the sonic flux, and thus use of the M-C switch may result in the computation of nonphysical expansion shocks. However, this typically happens only in rare two-dimensional cases. The M-C approach samples the shock speed only (and not the velocity field) but nonetheless captures shock waves with only one interior state, though the Godunov switch is the preferred approach.

- Solution Algorithm

The CAP-TSD code involves a cell-vertex finite-difference discretization of the transonic small disturbance equation and associated boundary conditions. The leading edges and tips of lifting surfaces are located between grid lines because of singularities that occur near the leading edges and along the wake edges for lifting cases. The small disturbance equation is solved numerically using an AF1-type approximate factorization algorithm developed by Batina.⁵ Details of the CAP-TSD algorithm development including comprehensive finite-difference formulas are reported by Batina.⁴⁰ In brief, the AF1 algorithm in CAP-TSD was derived by first writing the transonic small disturbance equation in general form as

$$R(\phi^{n+1}) = 0$$

where R is the residual evaluated at time level $n+1$. The solution is given by the linearization

$$R(\phi^*) + \left(\frac{\partial R}{\partial \phi} \right)_{\phi=\phi^*} \Delta\phi = 0$$

where

$$\Delta\phi = \phi^{n+1} - \phi^*$$

The two terms are evaluated using the currently available value of the velocity potential beginning with

$$\phi^* = \phi^n$$

For unsteady calculations where time accuracy is required, subiterations may be performed to drive

$$\phi^* \rightarrow \phi^{n+1}$$

The AF1 algorithm may be written as

$$L_{\xi} L_{\eta} L_{\zeta} \Delta\phi = -R(\phi)$$

where the product of operators on the left hand side is an approximate factorization of the second term of the linearization above. The equation is solved in a three-sweep procedure wherein each operator is applied individually as

$$L_{\xi} \Delta \bar{\phi} = -R(\phi^*)$$

$$L_{\eta} \Delta \bar{\bar{\phi}} = \Delta \bar{\phi}$$

$$L_{\zeta} \Delta \phi = \Delta \bar{\bar{\phi}}$$

and the sweeps are performed independently in streamwise, spanwise, and vertical directions. The solution is advanced with a constant time step, which is selected for numerical stability. Specifically, the size of the time step is selected by trial and error so that shock waves in the flow do not move more than one grid point per step. If that condition is violated, the solution becomes unstable, and the calculations diverge. This is the so-called moving shock instability inherent with AF1-type schemes as applied to the TSD equation.⁴¹

The ASP3D code involves a modern cell-centered finite-volume discretization, which allows exact planform treatment including leading edges, wing tips, and control surface edges. The advanced small perturbation equation is solved numerically using either an AF1 approximate factorization algorithm or an AF2-type⁴¹ algorithm. The AF1 algorithm within the ASP3D code is the cell-centered finite-volume version of the AF1 algorithm developed for CAP-TSD, which is defined in general terms above. The finite volume version of the AF1 algorithm is also susceptible to the moving shock instability if shock waves move more than one cell per time step.

The AF2 algorithm within the ASP3D code is described in general by

$$L_{\zeta} L_{\eta} L_{\xi} \Delta \phi = -R(\phi)$$

The general form of the AF2 algorithm appears to be similar to that of the AF1 algorithm but there are two fundamental differences. First, the operators are applied in the reverse order (here, vertical and spanwise sweeps are performed first, with the streamwise sweep performed last) and a streamwise implicit temporal damping term is included on the right hand side of the equation to eliminate the occurrence of the moving shock instability. The AF2 algorithm is defined specifically by

$$\left[\left(\frac{\bar{\alpha}}{J} \right) - \delta_{\xi} \hat{g}_3 \delta_{\zeta} \right] \Delta \bar{\phi}_i = -\omega R + \frac{1}{J} \left(\frac{\xi_x}{\Delta \tau} + \psi \frac{3\xi_x}{2\Delta t} B \right) \Delta \bar{\bar{\phi}}_{i-1}$$

$$\left[\left(\frac{\bar{\alpha}}{J} \right) - \delta_{\eta} \hat{g}_2 \delta_{\eta} \right] \Delta \bar{\bar{\phi}}_i = \left(\frac{\bar{\alpha}}{J} \right) \Delta \bar{\phi}_i$$

$$\left[\left(\frac{\bar{\alpha}}{J} \right) - (\hat{g}_1)_{i+1/2} \bar{\delta}_{\xi} \right] \Delta \phi_i = \left(\frac{\bar{\alpha}}{J} \right) \Delta \bar{\bar{\phi}}_i$$

where the flux Jacobians are defined by

$$\hat{g}_1 = \frac{|\nabla\xi|}{J} \left\{ \frac{\xi_x}{|\nabla\xi|} \left[\xi_x \left(\frac{\partial f_1}{\partial \phi_x} \right) + \xi_y \left(\frac{\partial f_1}{\partial \phi_y} \right) \right] + \frac{\xi_y}{|\nabla\xi|} \left[\xi_x \left(\frac{\partial f_2}{\partial \phi_x} \right) + \xi_y \left(\frac{\partial f_2}{\partial \phi_y} \right) \right] \right\}$$

$$\hat{g}_2 = \frac{|\nabla\eta|}{J} \left\{ \frac{\eta_y}{|\nabla\eta|} \left[\eta_y \left(\frac{\partial f_2}{\partial \phi_y} \right) \right] \right\}$$

$$\hat{g}_3 = \frac{|\nabla\xi|}{J} \left\{ \frac{\xi_z}{|\nabla\xi|} \xi_z \right\}$$

When the fluxes are substituted into the above equations the flux Jacobians become

$$\hat{g}_1 = \frac{|\nabla\xi|}{J} \left\{ \frac{\xi_x}{|\nabla\xi|} \left[\xi_x \left(D + 2E\phi_x + 3F\phi_x^2 + \frac{H}{2}\phi_y^2 \right) + \xi_y (2G\phi_y + H\phi_x\phi_y) \right] \right. \\ \left. + \frac{\xi_y}{|\nabla\xi|} \left[\xi_x (1 + \phi_x) H\phi_y + \xi_y \left(1 + H\phi_x + \frac{H}{2}\phi_x^2 + 3F\phi_y^2 \right) \right] \right\}$$

$$\hat{g}_2 = \frac{|\nabla\eta|}{J} \left\{ \frac{\eta_y}{|\nabla\eta|} \left[\eta_y \left(1 + H\phi_x + \frac{H}{2}\phi_x^2 + 3F\phi_y^2 \right) \right] \right\}$$

$$\hat{g}_3 = \frac{|\nabla\xi|}{J} \left\{ \frac{\xi_z}{|\nabla\xi|} \xi_z \right\}$$

Although not obvious, the temporal damping term is treated implicitly since it involves information from the second intermediate sweep but at the previous grid plane $i - 1$, and

$$\bar{\alpha} = \frac{\xi_x}{\Delta\tau} + \psi \left(\frac{2}{\Delta t^2} A + \frac{3\xi_x}{2\Delta t} B \right)$$

The parameter ψ is set equal to zero for steady applications and set equal to unity for unsteady applications. The parameter ω is an over-relaxation factor. The algorithm involves both physical (Δt) and computational ($\Delta\tau$) time steps. The computational step is selected for rapid convergence and it is usually assigned a large value. Hence the solution can be advanced using either a constant computational time step or a constant CFL number (variable time step). The physical time step is used only for unsteady applications, and it is selected based on the physics of the problem being considered.

The AF2 approach is the preferred algorithm. The AF2 scheme is more robust than the AF1 scheme since it cannot fail due to the moving shock instability by design, since

the temporal damping term eliminates the instability. Hence convergence to steady state is achieved with large time steps or large CFL numbers, independent of the moving shock instability that plagues the AF1 algorithm. The AF2 scheme also may be used for unsteady calculations when local iteration is applied. Therefore, the physical time step may be selected based on the physics of the problem being solved, rather than on numerical stability considerations. In either steady or unsteady applications, the temporal damping term vanishes in the steady-state limit or within the convergence process of performing subiterations (for unsteady calculations).

- Residual Calculation

The CAP-TSD code does not use the right hand side of the equations to be solved, termed the residual, as the measure of solution convergence. Instead it uses the L_2 -norm of the change in velocity potential from one time step to the next given by

$$\Delta\phi = \phi^{n+1} - \phi^*$$

This is not the best measure of solution convergence because if the solution “hangs” somewhere in the flow field, which means that the solution is no longer converging there, then the change in velocity potential would be zero but the solution residual is not zero.

A better measure of the convergence of the solution is given by the residual, which is defined by the right hand side of the equations to be solved, such as

$$L_{\xi}L_{\eta}L_{\zeta}\Delta\phi = -R(\phi)$$

As the solution converges, the residual R is driven to zero. If the solution “hangs” the residual will not be zero and the user will know it immediately. The ASP3D code uses the L_2 -norm of the residual R as the measure of convergence because it correctly assesses the performance of the algorithm in solving the governing equation and associated boundary conditions.

- Multigrid Implementation

The AF1 algorithm is not appropriate for multigrid implementation because the rapid rate of convergence of the multigrid procedure would result in the moving shock instability. The AF2 algorithm of the ASP3D code is amenable to multigrid implementation because the temporal damping term prevents the moving shock instability from occurring. Therefore the multigrid procedure⁴² may be used for steady or iterative unsteady applications with ASP3D. The procedure is implemented in FAS (Full Approximation Scheme) form,⁴³ which is applicable to nonlinear governing equations. As with any multigrid procedure, high-frequency errors are damped on the fine mesh and low-frequency errors are damped on the coarser meshes. Either V- or W-cycles may be selected as suggested by Figure 14 and full multigrid is available. W-cycles (Figure 14(b)) have been found to be slightly more effective for solution convergence, although W-cycles are slightly more expensive than V-cycles (Figure 14(a)).⁴⁴

Specifically, as diagrammed in Figure 14, the calculation on the finest mesh (N) smoothes high frequency errors using the AF2 algorithm written in general form

$$L_N(\phi_N) = R_N$$

Subsequent calculations on the next coarser mesh ($N-1$) involve fine mesh residual injection

$$L_{N-1}(\phi_{N-1}) = R_{N-1} - \hat{I}_N^{N-1}(R_N) + L_{N-1}I_N^{N-1}(\phi_N)$$

where the I functions are restriction operators for transferring the velocity potential and residual from the fine mesh to the coarse mesh. The procedure is generally repeated for additional coarse meshes ($N-2, N-3, \dots$).

The restriction operator⁴⁴ for the velocity potential is a volume-weighted operator defined by

$$\phi_{N-1} = I_N^{N-1}\phi_N = \frac{\sum V\phi_N}{\sum V}$$

and the restriction operator⁴⁴ for residuals is defined by

$$\hat{I}_N^{N-1}R_N = \sum R_N$$

where the summations above are taken over all of the fine mesh cells that make up the coarse mesh cell. The interpolation operator for the velocity potential that is required to transfer information back from coarse to fine mesh is a trilinear interpolation operator defined symbolically by

$$\phi_N \leftarrow \phi_N + I_{N-1}^N\phi_{N-1}$$

- Numerical Accuracy

The grid metrics in the CAP-TSD code are calculated to be consistent with the finite differencing of the velocities as reported by Batina.⁴⁰ This is important, since if the code were a full potential code rather than one that solves the small disturbance potential equation, a uniform flow would be an exact solution of the difference equations. The velocities are computed using second-order accurate central differences, which are exact for a linear test function on a uniform grid. Generally, the CAP-TSD solutions are second-order accurate at subsonic points and first-order accurate at supersonic points. The second-order-accurate supersonic capability contained within the code is conceptually incorrect and therefore should not be used.

In the ASP3D code, geometric quantities such as areas, volumes, and direction cosines are computed using exact formulas. The velocity components are computed using similar formulas as used in CAP-TSD, but in ASP3D, they are exact for a linear

test function on a general grid. Generally speaking, the ASP3D algorithm is second-order-accurate within subsonic cells and first-order accurate within supersonic cells, although a second-order-accurate supersonic capability is available as a user-selected option.

Also in CAP-TSD, the $\phi_{,xt}$ term in the governing equation is computed using first-order-accurate backward differencing for stability reasons, similarly to that within the XTRAN3S code. However, for unsteady applications at higher reduced frequencies encountered with control surfaces or higher wave number phenomena such as Kutta waves, this differencing may be too dispersive. In contrast, first-, second-, and third-order-accurate backward differencing operators are available for the $\phi_{,xt}$ term in ASP3D as user-selected options to improve the accuracy for such applications.

- **Numerical Stability**

As discussed previously, with the CAP-TSD code the step size for steady or unsteady applications must be selected iteratively to determine a value that avoids the moving shock instability. Although the AF1 algorithm developed for CAP-TSD is unconditionally stable based on a linear stability analysis, the moving shock instability limits practical applications to step sizes that allow the shock waves in the solution to propagate at most one grid point per time step.

The ASP3D code with the AF2 algorithm is generally more stable than the CAP-TSD code, in part, because the moving shock instability is avoided. Thus large time steps or large CFL numbers may be selected for rapid convergence to steady state. For unsteady applications, the physical time step may be selected for the accuracy of the problem under consideration independent of numerical stability issues, and a large computational step size or CFL number may be selected for rapid convergence of the iterative procedure.

- **Structural Equations of Motion**

Similar to the CAP-TSD code, the flow equations of ASP3D are coupled with the structural equations of motion for simultaneous time-integration for aeroelastic analysis. Specifically, such analysis involves the coupling of the aerodynamics with the structural characteristics of the configuration under consideration.⁸ The resulting equations of motion for a time domain or time-marching aeroelastic solution are based on the aircraft natural vibration modes. These equations are integrated in time along with the finite volume solution of the flow field in ASP3D. Initial conditions for each structural mode are specified and free decay transients are computed. Initial velocities in one or more modes, rather than displacements, have been found to be superior in avoiding nonphysical, strictly numerical transients and their possibly associated instabilities. Aeroelastic stability is then deduced from the free decay records or time histories. This is a fairly standard procedure for aeroelastic analyses with the small perturbation computer codes, and thus, the interested reader is referred to Cunningham, et al.⁸ or Bennett, et al.⁹ for further details including equations and representative calculations.

Results and Discussion

Representative ASP3D results are presented to demonstrate application of the new code. Several cases were considered including: (1) multigrid calculations for the NACA 64A410 airfoil, (2) full-multigrid calculations for the NACA 0012 airfoil, (3) steady and unsteady calculations for the F-5 fighter wing⁴⁵ and (4) viscous calculations for the NACA RM L51F07 wing-fuselage configuration.⁴⁶ Most of the calculations are for steady flow applications. Unsteady and aeroelastic applications using ASP3D will be published elsewhere.

- NACA 64A410 Airfoil Results

Flow calculations were performed for airfoil configurations using a 257 x 129 Cartesian finite volume mesh that had a fifty chord extent as shown in Figure 15. The spacing at the leading and trailing edges is $\Delta x = 0.005c$, as shown in Figures 16(a) and 16(b), respectively, which is generally considered small for transonic small perturbation applications. There is only one cell in the spanwise direction for airfoil applications, which effectively produces a two-dimensional solution. Multigrid calculations were performed using this mesh (termed Mesh 6) and five subsets of the mesh (Meshes 1 through 5) determined by simply deleting every other grid line. This is done automatically within the ASP3D code and thus is not something that the user needs to do during mesh generation. The six meshes are shown in Figures 17(a) to 17(f) including the total number of points in each direction for each mesh. The smallest mesh, for example, is a very coarse 9 x 5 mesh, which has only a few cells in the far field. Corresponding near field views of these six meshes are shown in Figures 18(a) through 18(f) with the airfoil slit in the center of each plot. The individual figures illustrate the relative mesh density near the airfoil surface.

The first case considered is that of the NACA 64A410 airfoil at an angle of attack of $\alpha = 0^\circ$ and $M_\infty = 0.72$, the same freestream Mach number used in Figures 1 and 2, and the same case as shown in Figures 3 and 5, whereby the ASP streamwise mass flux and mass conserving surface boundary conditions were used. Multigrid calculations were performed using twenty W-cycles on the six meshes shown in Figures 17 and 18, a CFL number of ten, and an over-relaxation factor (ω) of 1.33. The convergence history from these calculations is shown in Figure 19(a) and the number of supersonic points is shown in Figure 19(b). The solution is converged over seven orders of magnitude in twenty multigrid cycles, as measured by the L_2 -norm of the residual, and the number of supersonic points reaches its final value of 410 in just five cycles. The ASP2D pressure coefficient distribution is shown in Figure 20, which indicates that there is a shock wave of moderate strength on the upper surface of the airfoil. This pressure distribution is in very good agreement with a conservative full-potential (FP) calculation shown in Figure 21, reported by Jameson²⁶ for this case. A comparison of the ASP2D (Figure 20) and FP (Figure 21) pressure distributions demonstrates that the strength and position of the shock wave at approximately 63% chord on the upper surface of the airfoil are predicted accurately by the ASP2D code. Also, the ASP2D code captured the shock wave with only one interior state due to the sharp shock capturing ability of the Godunov scheme.

- NACA 0012 Airfoil Results

The second case considered is that of the NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2^\circ$. This case involves a stronger shock wave on the upper surface of the airfoil, so the full multigrid capability was exercised. Here, ten W-cycles of the inviscid irrotational multigrid acceleration were performed using the coarsest four meshes (Meshes 1 to 4) of those shown in Figures 17 and 18. The resulting solution was interpolated to the next finest mesh (Mesh 5) to perform another ten cycles of multigrid, and that result was interpolated to the finest mesh (Mesh 6) for twenty additional cycles of multigrid acceleration. In these calculations the CFL number was twelve and the over-relaxation factor (ω) was set equal to 1.4. The resulting convergence history and number of supersonic points are shown in Figures 22(a) and 22(b), respectively. The convergence history indicates that the multigrid procedure reduced the residual approximately three-and-a-half orders of magnitude for the first two sets of calculations where the coarser meshes were used, and the final multigrid procedure reduced the residual nearly seven orders of magnitude. The number of supersonic points shows the rapid convergence of each of the three multigrid procedures to 40, 148, and finally 585 supersonic points. The final value of 585 was attained after only four multigrid cycles using all six meshes.

The corresponding ASP2D pressure coefficient distribution is presented in Figure 23. These pressures indicate that there is a relatively strong shock wave on the upper surface of the NACA 0012 airfoil near 57% chord that is captured with only one interior state. There are also no overshoots or undershoots near the shock, and the pressure distribution is very smooth everywhere else including in the leading edge region where stagnation occurs. Furthermore, the ASP2D pressures are in very good agreement with a full potential solution for this case computed by Jameson,⁴⁰ as shown in Figure 24. A comparison of the ASP2D (Figure 23) and FP (Figure 24) pressure distributions demonstrates that the strength and position of the shock wave at approximately 57% chord on the upper surface of the airfoil are predicted accurately by the ASP2D code.

- F-5 Fighter Wing Results

The F-5 fighter wing⁴⁵ has a leading edge sweep of 31.92 degrees, an aspect ratio of 1.578, and a taper ratio of 0.283. The wing has a NACA 65A004.8 airfoil section that has a drooped leading edge. The wing was modeled for ASP3D applications using a 97 x 25 x 65 finite volume mesh as shown in Figure 25. A near field view of the 97 x 25 planform mesh is shown in Figure 25(a) and a view of the 97 x 65 root sectional mesh is shown in Figure 25(b). The meshes were generated using a program that was created to construct meshes specifically for use by the ASP3D code, and as such, it can generate meshes for aircraft configurations with multiple lifting surfaces, with multiple leading and trailing edge controls, and a fuselage. The mesh generator automatically creates the input file for ASP3D, which makes it user friendly. The meshes were generated using an extension of the polynomial mesh generation procedure developed by Bland.⁴⁷ Here though, the equations were generalized to the physical domain rather than the computational domain, which allows the creation of three-dimensional meshes about swept tapered planforms with the ability to control the spacing along the leading and

trailing edges. This is especially an issue in the outboard region of the wing near the tip. For example, Figures 26(a) and 26(b) show near field views of the planform mesh near the wing tip leading and trailing edges, respectively, for the F-5 wing. The figures show that the mesh was generated with a uniform spacing about the leading and trailing edges in the streamwise direction, and in fact, the spacing is also uniform across the tip in the spanwise direction. The planform mesh also emphasizes that all edges of the lifting surface are modeled with grid lines for ASP3D application. This is in contrast with meshes generated for CAP-TSD application, which cannot have grid lines on leading edges or the edge of the wake (wing tip) because of singularities that occur along these lines. Details of the mesh generation procedures developed for the ASP3D computer program will be published elsewhere.

To demonstrate the multigrid procedures for a three dimensional case, calculations were performed using this 97 x 25 planform mesh (termed Mesh 4) and three subsets of the mesh (Meshes 1 through 3) determined by simply deleting every other grid line. This is done automatically within the ASP3D code and thus is not something that the user needs to do during mesh generation. The four planform meshes are shown in Figures 27(a) to 27(d) including the total number of points in each direction for each mesh. The smallest mesh, for example, is a very coarse 13 x 4 mesh, which has only a few cells on the surface of the wing. Similarly, near field views of the corresponding four sectional meshes at the root of the F-5 wing are shown in Figures 28(a) through 28(d) with the airfoil slit in the center of each plot. The individual figures illustrate the relative mesh density near the airfoil surface along the wing root.

Inviscid multigrid calculations for the F-5 wing at $M_\infty = 0.897$ and $\alpha = -0.004^\circ$ are presented in Figure 29. The convergence history is shown in Figure 29(a) and the number of supersonic points is shown in Figure 29(b). The convergence history indicates that the residual was reduced nearly seven orders of magnitude with twenty-five cycles of multigrid acceleration. The number of supersonic points indicates that the final value of 1192 points was achieved in only eight cycles. The resulting pressure coefficient distributions are shown in Figure 30(a) for the upper surface of the wing and in Figure 30(b) for the lower surface. For this case there is a weak shock wave on the upper surface of the wing in the outboard region near the wing tip. There is additional supersonic flow on the lower surface near the leading edge where a large flow expansion occurs because of the drooped nose of the NACA 65A004.8 airfoil.

Comparisons of ASP3D pressure coefficient distributions with the NLR experimental data⁴⁵ for the F-5 wing are presented in Figure 31. Figure 31(a) shows pressure comparisons for the subsonic freestream case of $M_\infty = 0.897$ and $\alpha = -0.004^\circ$, which is the case presented in Figures 29 and 30, and Figure 31(b) shows pressure comparisons for a supersonic freestream case of $M_\infty = 1.328$ and $\alpha = -0.005^\circ$. The ASP3D pressures were interpolated to the eight span stations where the data were measured ranging from $\eta = 0.174$ in the inboard region of the wing to $\eta = 0.939$ near the wing tip. Both figures indicate that the ASP3D pressure distributions are in very good agreement with the experimental data including the leading edge and wing tip regions.

To demonstrate the effects of viscosity for a more difficult case involving a stronger shock wave, inviscid and viscous calculations were performed (both using the entropy and vorticity effects) for the F-5 fighter wing at $M_\infty = 0.946$, $\alpha = -0.004^\circ$, and $Re = 5.89 \times 10^6$. These results are presented in Figure 32, with the inviscid pressure distributions shown in Figure 32(a) and the viscous pressure distributions shown in Figure 32(b). The inviscid pressure distributions (Figure 32(a)) show generally good agreement with the experimental pressure data⁴⁵ except in the regions of the shock waves on the upper and lower surfaces of the wing. This is expected because of the neglect of viscous effects. The viscous pressure distributions (Figure 32(b)) are similar to the inviscid pressures and thus also agree well with the data, although the shock waves are slightly weaker and located more forward on the wing, and hence, they are in better agreement with the experimental shock results.

To demonstrate the iterative multigrid capability for time-dependent applications, unsteady calculations were performed for the F-5 fighter wing pitching about the root midchord with an amplitude of $\alpha_i = 0.109^\circ$ and a reduced frequency based on root semichord of $k = 0.237$. In these calculations the freestream Mach number was $M_\infty = 0.897$ and the mean angle of attack was $\alpha_0 = -0.004^\circ$, the same conditions used in Figures 29, 30, and 31(a). Here though, the wing was forced to oscillate harmonically in pitch for four cycles of motion to ensure periodicity of the flow field with the resulting lift and moment coefficients shown in Figure 33(a). The calculations were performed using only sixteen steps per cycle of motion, corresponding to a very large time step of $\Delta t = 1.433$, wherein the multigrid procedure was used at each time step to reduce the solution residual by two orders of magnitude as shown in Figure 31(b). By iterating on the solution at each time step, the linearization and factorization errors inherent in the AF2 algorithm are reduced to an acceptable level ($\sim 10^{-4}$) to produce a time accurate solution. The technique is analogous to performing subiterations per time step with CAP-TSD to ensure time accuracy, but the iterative multigrid procedure in ASP3D is cheaper computationally because the iteration is performed on coarser grids within the overall multigrid capability.

The resulting unsteady pressure coefficient distributions, normalized by the amplitude of motion and interpolated to the span stations where the NLR experimental pressure data were measured, are presented in Figure 34. The unsteady pressure distributions on the upper surface of the wing are shown in Figure 34(a) and the unsteady pressure distributions on the lower surface of the wing are shown in Figure 34(b). The ASP3D pressure distributions were computed from the fourth cycle of motion and the real and imaginary parts of the pressures are the in-phase and out-of-phase components of the pressure time histories, respectively. The unsteady pressures on the upper surface of the wing agree reasonably well with the experimental data⁴⁵ except near the shock wave. Here the magnitude of the pressure pulse due to the movement of the shock wave is slightly over predicted in the ASP3D calculation because of the neglect of viscous effects. On the lower surface of the wing (Figure 34(b)) the unsteady pressures are in very good agreement with the experimental data, with relatively uniform agreement with the data from the wing root to the wing tip.

- NACA RM L51F07 Wing-Fuselage Results

Fuselage modeling was implemented within the ASP3D code in a similar fashion as was developed originally in the XTRAN3S code by the author⁴⁸ and implemented also in CAP-TSD.⁶ Similar to that of Mason, et al.,⁴⁹ the fuselage is modeled by applying the fuselage surface tangency boundary condition on a constant rectangular cross-section computational surface rather than on the true fuselage surface as illustrated in Figure 35. This is done for simplicity rather than attempting to model the exact fuselage surface with a Cartesian mesh. The computational fuselage surface extends from the upstream grid boundary to the downstream grid boundary, the rectangular cross section of which has maximum dimensions that approximate the maximum fuselage diameter. This typically occurs in the wing-fuselage junction region. The fuselage boundary condition then simplifies to two planar boundary conditions shown in the upper right portion of Figure 35, applied along the top/bottom and side surfaces of the computational fuselage surface. These simplified fuselage boundary conditions are analogous to the lifting surface boundary condition that is applied on the mean plane of the surface rather than on the true surface.

To account for the spatial differences between true and computational fuselage surfaces, slender body theory corrections are applied.⁴⁹ Separate corrections are imposed for fuselage thickness and angle of attack, which modify the boundary conditions as shown in Figure 35. The correction applied to the fuselage thickness terms is derived by representing thickness by a source distribution with strength proportional to the rate of change of fuselage area. By requiring that the net source strength across the true and computational surfaces be equivalent at a given cross section, the thickness terms are scaled by the ratio of arc lengths along the true and computational fuselage surfaces as graphically defined in Figure 35. Similarly the correction applied to the fuselage angle of attack term is derived by representing angle of attack by a doublet distribution with strength proportional to cross sectional area. By equating doublet strengths, the angle of attack term is then scaled by a ratio of cross sectional areas as also graphically indicated in Figure 35.

Computationally, the fuselage boundary conditions are applied along all of the cell interfaces that lie on the computational fuselage surface as shown in Figure 36. The figure also indicates the cell interfaces where wing and symmetry plane boundary conditions are imposed. Hence to model the fuselage, the vertical and spanwise sweeps of the AF2 algorithm within ASP3D are modified to include the top/bottom and side surface fuselage boundary conditions, respectively. No flow is computed within cells that are inside the computational fuselage by solving the trivial equation.

To demonstrate the ASP3D fuselage modeling capability, calculations were performed for the NACA RM L51F07 wing-fuselage configuration. The geometry of the NACA RM L51F07 wing-fuselage configuration⁴⁶ is defined in Figure 37. The wing has a leading edge sweep of 46.76 degrees (quarter chord sweep of 45 degrees), an aspect ratio of 4.0, and a taper ratio of 0.6. The wing has a NACA 65A006 airfoil section. The fuselage is an axisymmetric body with fineness ratio of 12. The model was sting

mounted for wind tunnel testing and pressures were measured along five chords of the wing as well as on the fuselage. The NACA RM L51F07 configuration was modeled for ASP3D applications using a 129 x 29 x 75 finite volume mesh. A near field view of the 129 x 29 planform mesh is shown in Figure 38. The mesh was generated in the streamwise direction by clustering points near the wing, and there is a slight clustering near the fuselage nose as well as where the fuselage is attached to the sting. The mesh was generated in the spanwise direction by clustering cells near the fuselage and near the wing tip.

Viscous calculations were performed including the entropy and vorticity effects for the NACA RM L51F07 configuration at $M_\infty = 0.93$, $\alpha = 2.0^\circ$, and $Re = 10^7$. The resulting ASP3D pressure coefficient contour lines are shown in Figure 39(a) and the pressure coefficient distributions on the wing are shown in Figure 39(b). These pressures indicate that there is a shock wave across the upper surface of the wing that begins inboard across the top of the fuselage near the wing root trailing edge. The shock wave is swept slightly aft in the outboard direction from the wing root trailing edge to the wing tip leading edge, as shown in both Figures 39(a) and 39(b). The pressure coefficients on the lower surface suggest that the flow there is smooth and subcritical.

The ASP3D results are presented again for this case in Figure 40 including comparisons with the experimental data.⁴⁶ The wing pressure comparisons are shown in Figure 40(a) and the fuselage pressure comparisons are shown in Figure 40(b). The ASP3D pressure distributions on the wing were interpolated to the five span stations where experimental data were measured ranging from $\eta = 0.2$ near the wing fuselage junction to $\eta = 0.95$ near the wing tip. The ASP3D pressure distributions on the fuselage were directly computed along the fuselage centerline for comparison with the experimental data measured at $\eta = 0.0$. The wing pressure comparisons of Figure 40(a) show good overall agreement in pressure levels and in the prediction of the strength and location of the upper surface shock wave. The largest differences between computation and experiment occur in the inboard region at $\eta = 0.2$, which may be because there is no boundary layer on the fuselage (only on the wing). The fuselage pressure comparisons of Figure 40(b) also show good agreement with the data, although the calculated shock wave is a little sharper than the wing shock wave due to the neglect of viscous effects on the fuselage.

Conclusions

A new computer program has been developed called ASP3D (Advanced Small Perturbation – 3D), which solves the small perturbation potential flow equation in an advanced form including mass-consistent surface and trailing wake boundary conditions, and entropy, vorticity, and viscous effects. The purpose of the program is for unsteady aerodynamic and aeroelastic analyses, especially in the nonlinear transonic flight regime. The program exploits the simplicity of stationary Cartesian meshes with the movement or deformation of the configuration under consideration incorporated into the solution algorithm through a planar surface boundary condition. The new ASP3D code is the result of a decade of developmental work on improvements to the small perturbation

formulation, performed while the author was employed as a Senior Research Scientist in the Configuration Aerodynamics Branch at the NASA Langley Research Center. The ASP3D code is a significant improvement to the state-of-the-art for transonic aeroelastic analyses over the CAP-TSD code (Computational Aeroelasticity Program – Transonic Small Disturbance), which was developed principally by the author in the mid-1980s. The author is in a unique position as the developer of both computer programs to compare, contrast, and ultimately make conclusions regarding the underlying formulations and utility of each code. The paper described the salient features of the ASP3D code including the rationale for improvements in comparison with CAP-TSD.

The ASP3D code involves a modern cell-centered finite-volume discretization, which allows exact planform treatment including leading edges, wing tips, and control surface edges. It can treat aircraft configurations involving multiple lifting surfaces with leading and trailing edge control surfaces, and a fuselage. The advanced-small-perturbation potential flow equation is solved numerically using either an AF1- or an AF2-type approximate factorization algorithm. The AF1 algorithm within the ASP3D code is the cell-centered finite-volume version of the AF1 finite-difference algorithm developed for CAP-TSD. However, the AF1 algorithm is susceptible to the so-called moving shock instability if shock waves move more than one grid point or cell per time step. In contrast, the AF2 scheme has a temporal damping term that eliminates the moving shock instability by design, and thus, the AF2 algorithm is more robust than the AF1 scheme. Thus, with the AF2 approach, convergence to steady state is achieved with large time steps or large CFL numbers. And because the moving shock instability does not occur, the AF2 algorithm of the ASP3D code is also amenable to multigrid implementation. Therefore, a multigrid procedure was developed within ASP3D that may be used for steady or iterative unsteady applications. The procedure was implemented in FAS (Full Approximation Scheme) form, which is applicable to nonlinear governing equations. The multigrid procedure may be used for unsteady calculations when local iteration is applied. Therefore, the physical time step may be selected for the accuracy of the problem under consideration independent of numerical stability issues, and a large computational step size or CFL number may be selected for rapid convergence of the iterative procedure.

Results were presented for the NACA 64A410 and NACA 0012 airfoils, the F-5 fighter wing, and the NACA RM L51F07 wing-fuselage configuration to demonstrate various features and capabilities of the ASP3D code. Inviscid irrotational calculations for the NACA 64A410 and NACA 0012 airfoils demonstrated the rapid convergence to steady state achieved using the multigrid capability. The resulting pressure coefficient distributions showed very good agreement with published full potential results especially in terms of shock strength and location. Inviscid and viscous calculations for the F-5 fighter wing demonstrated the accuracy of the ASP3D code for subsonic and supersonic freestream conditions. Unsteady calculations for the F-5 wing demonstrated the use of an iterative multigrid capability that allows the use of very large physical time steps. The resulting unsteady pressure distributions showed good agreement with the experimental data even though the calculations involved only sixteen steps per cycle of motion. Calculations for the NACA RM L51F07 wing-fuselage configuration demonstrated

application of ASP3D to a more complex configuration than airfoils and wings. The resulting viscous pressure coefficient distributions on the wing and the fuselage showed good agreement with the experimental data. The comparisons were for a transonic case involving a shock wave on the upper surface of the wing. Many of the cases considered either physically involved flow conditions or numerically involved finer meshes or larger time steps than would be considered typically using the CAP-TSD program. Therefore the general conclusion is that the new ASP3D capability is superior to the older CAP-TSD code because of the myriad improvements developed and incorporated.

Acknowledgement

The NASA Langley Research Center supported the work contained herein while the author was employed as a Senior Research Scientist in the Configuration Aerodynamics Branch.

References

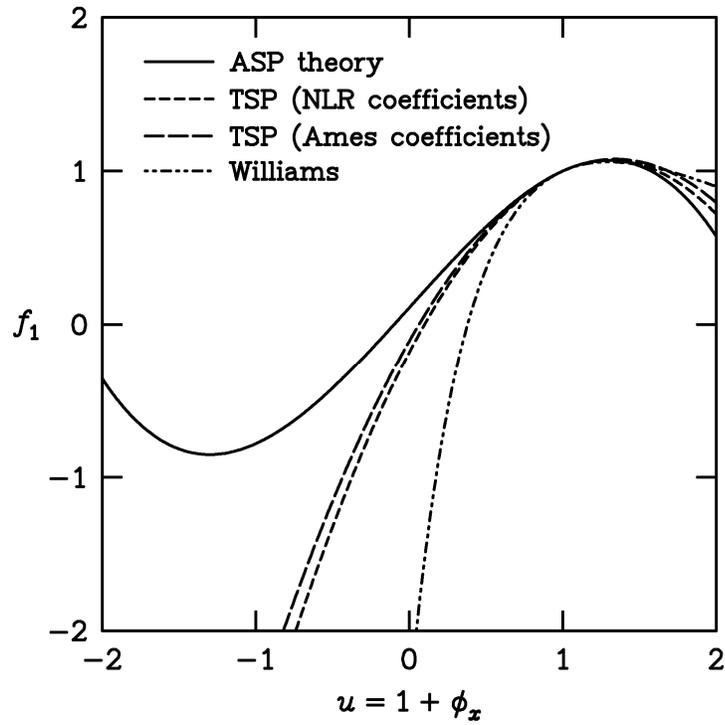
1. Edwards, J. W.; and Thomas, J. L.: Computational Methods for Unsteady Transonic Flows, AIAA Paper No. 87-0107, January 1987.
2. Edwards, J. W.; and Malone, J. M.: Current Status of Computational Methods for Transonic Unsteady Aerodynamics and Aeroelastic Applications, AGARD CP-507, March 1992, pp. 1-1 to 1-24.
3. Schuster, D. M.; Liu, D. D.; and Huttshell, L. J.: Computational Aeroelasticity: Success, Progress, Challenge, AIAA Paper No. 2003-1725, April 2003.
4. Borland, C. J.; and Rizzetta, D. P.: Nonlinear Transonic Flutter Analysis, *AIAA Journal*, Vol. 20, November 1982, pp. 1606-1615.
5. Batina, J. T.: An Efficient Algorithm for Solution of the Transonic Small-Disturbance Equation, AIAA Paper No. 87-0109, January 1987; also *Journal of Aircraft*, Vol. 25, July 1988, pp. 598-605.
6. Batina, J. T.; Seidel, D. A.; Bland, S. R.; and Bennett, R. M.: Unsteady Transonic Flow Calculations for Realistic Aircraft Configurations, AIAA Paper No. 87-0850, April 1987; also *Journal of Aircraft*, Vol. 26, January 1989, pp. 21-28.
7. Bennett, R. M.; Bland, S. R.; Batina, J. T.; Gibbons, M. D.; and Mabey, D. G.: Calculation of Steady and Unsteady Pressures on Wings at Supersonic Speeds with a Transonic Small-Disturbance Code, AIAA Paper 87-0851, April 1987.
8. Cunningham, H. J.; Batina, J. T.; and Bennett, R. M.: Modern Wing Flutter Analysis by Computational Fluid Dynamics Methods, ASME Paper No. 87-WA/Aero-9, December 13-18, 1987; also *Journal of Aircraft*, Vol. 25, October 1988, pp. 962-968.

9. Bennett, R. M.; Batina, J. T.; and Bland, S. R.: Wing-Flutter Calculations with the CAP-TSD Unsteady Transonic Small-Disturbance Program, AIAA Paper No. 88-2347, April 1988; also *Journal of Aircraft*, Vol. 26, September 1989, pp. 876-882.
10. Gibbons, M. D.: Aeroelastic Calculations using CFD for a Typical Business Jet Model, NASA CR-4753, September 1996.
11. Gibbons, M. D.: personal communication, October 20, 1994.
12. Steinhoff, J.; and Jameson, A.: Multiple Solutions of the Transonic Potential Flow Equation, AIAA Paper No. 81-1019, June 1981.
13. Williams, M. H.; Bland, S. R.; and Edwards, J. W.: Flow Instabilities in Transonic Small Disturbance Theory, NASA TM 86251, January 1985.
14. Batina, J. T.: Unsteady Transonic Algorithm Improvements for Realistic Aircraft Applications, AIAA Paper No. 88-0105, January 1988; also *Journal of Aircraft*, Vol. 26, February 1989, pp. 131-139.
15. Batina, J. T.: Unsteady Transonic Small-Disturbance Theory Including Entropy and Vorticity Effects, AIAA Paper No. 88-2278, April 1988; also *Journal of Aircraft*, Vol. 26, June 1989, pp. 531-538.
16. Jones, D. J.: Reference Test Cases and Contributors, *Test Cases for Inviscid Flow Field Methods*, AGARD-AR-211, May 1985.
17. Batina, J. T.: Advanced Small Perturbation Potential Flow Theory for Unsteady Aerodynamic and Aeroelastic Analyses, NASA/TM-2005-213908, 2005.
18. Whitfield, D. L.: Analytical Description of the Complete Two-Dimensional Turbulent Boundary-Layer Velocity Profile, AEDC-TR-77-79, September 1977; also AIAA Paper No. 78-1158, July 1978.
19. Swafford, T. W.: Analytical Approximation of Two-Dimensional Separated Turbulent Boundary-Layer Velocity Profiles, AEDC-TR-79-99, October 1980.
20. Drela, M.: Two-Dimensional Transonic Aerodynamic Design and Analysis Using the Euler Equations, GTL Report No. 187, February 1986.
21. Howlett, J. T.: Efficient Self-Consistent Viscous-Inviscid Solutions for Unsteady Transonic Flow, AIAA Paper No. 85-0482, January 1985.
22. Howlett, J. T.: Calculation of Unsteady Transonic Flows With Mild Separation by Viscous-Inviscid Interaction, NASA TP 3197, June 1992.

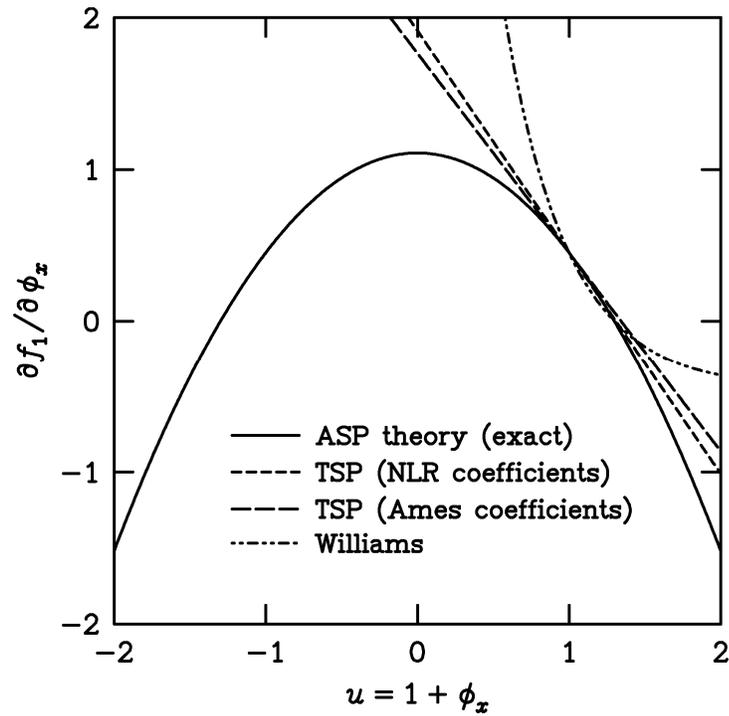
23. Green, J. E.; Weeks, D. J.; and Brooman, J. W. F.: Prediction of Turbulent Boundary Layers and Wakes in Compressible Flow By a Lag-Entrainment Method, British Aeronautical Research Council Reports and Memoranda No. 3791, January 1973.
24. Edwards, J. W.: Transonic Shock Oscillations Calculated with a New Interactive Boundary Layer Coupling Method, AIAA Paper No. 93-0777, January 1993.
25. Fuglsang, D. F.; and Williams, M. H.: Non-Isentropic Unsteady Transonic Small Disturbance Theory, AIAA Paper 85-0600, April 1985.
26. Jameson, A.: Transonic Flow Calculations, Lecture Series 87, Computational Fluid Dynamics, von Karman Institute for Fluid Dynamics, March 15-19, 1976.
27. Anderson, W. K.; Thomas, J. L.; and van Leer, B.: Comparison of Finite-Volume Flux-Vector Splittings for the Euler Equations, AIAA Paper No. 85-0122, January 1985.
28. Cook, P. H.; McDonald, M. A.; and Firmin, M. C. P.: Aerofoil RAE 2822 – Pressure Distribution, and Boundary Layer and Wake Measurements, *Experimental Data Base for Computer Program Assessment*, AGARD-AR-138, May 1979.
29. McDevitt, J. B.; and Okuno, A. F.: Static and Dynamic Pressure Measurements on a NACA 0012 Airfoil in the Ames High Reynolds Number Facility, NASA TP-2485, June 1985.
30. Kuethe, A. M.; and Chow, C.-Y.: *Foundations of Aerodynamics: Bases of Aerodynamic Design*, John Wiley and Sons, New York, Third Edition, 1976.
31. Hafez, M.; and Lovell, D.: Entropy and Vorticity Corrections for Transonic Flows, AIAA Paper No. 83-1926, July 1983.
32. Clebsch, A.: Uber die Integration der Hydrodynamischen Gleichungen, *J. Reine Angew. Math.*, Vol. 57, 1959, pp. 1-10.
33. Grossman, B.: The Computation of Inviscid Rotational Gasdynamic Flows Using an Alternate Velocity Decomposition, AIAA Paper No. 83-1900, July 1983.
34. Dang, T. Q.; and Chen, L. T.: An Euler Correction Method for Two- and Three-Dimensional Transonic Flows, AIAA Paper No. 87-0522, January 1987.
35. Melnik, R. E.; and Brook, J. W.: The Computation of Viscid/Inviscid Interaction on Airfoils With Separated Flow, *Third Symposium on Numerical and Physical*

- Aspects of Aerodynamic Flows*, California State University, January 1985, pp. 1-21 to 1-37.
36. Mead, H. R.; and Melnik, R. E.: GRUMFOIL: A Computer Code for the Viscous Transonic Flow Over Airfoils, NASA CR 3806, October 1985.
 37. Engquist, B. E.; and Osher, S. J.: Stable and Entropy Satisfying Approximations for Transonic Flow Calculations, *Mathematics of Computation*, Vol. 34, No. 149, January 1980, pp. 45-75.
 38. Godunov, S.: A Finite Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics, *Mathematics of the USSR*, Sbornik, Vol. 47, 1959.
 39. Murman, E. M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods, Proceeds of the AIAA Computational Fluid Dynamics Conference, July 1973, pp. 27-40.
 40. Batina, J. T.: A Finite-Difference Approximate-Factorization Algorithm for Solution of the Transonic Small-Disturbance Equation, NASA TP 3129, January 1992.
 41. Ballhaus, W. F.; Jameson, A.; and Albert, J.: Implicit Approximate-Factorization Schemes for Steady Transonic Flow Problems, *AIAA Journal*, Vol. 16, June 1978, pp. 573-579.
 42. Jameson, A.: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA Paper No. 79-1458, July 1979.
 43. Brandt, A.: Multi-Level Adaptive Solutions to Boundary-Value Problems, *Mathematics of Computation*, Vol. 31, April 1977, pp. 333-390.
 44. Anderson, W. K.: Implicit Multigrid Algorithms for the Three-Dimensional Flux Split Euler Equations, Doctoral Thesis, Mississippi State University, Mississippi, August 1986.
 45. Tijdeman, H.; Van Nunen, J. W. G.; Kraan, A. N.; Persoon, A. J.; Poestkoke, R.; Roos, R.; Schippers, P.; and Siebert, C. M.: Transonic Wind Tunnel Tests on an Oscillating Wing with External Stores, AFFDL-TR-78-194, December 1978.
 46. Loving, D. L.; and Estabrooks, B. B.: Transonic-Wing Investigation in the Langley 8-Foot High-Speed Tunnel at High Subsonic Mach Numbers and at a Mach Number of 1.2, Analysis of Pressure Distribution of Wing-Fuselage Configuration Having a Wing of 45° Sweepback, Aspect Ratio 4, Taper Ratio 0.6, and NACA 65A006 Airfoil Section, NACA RM L51F07, September 1951.

47. Bland, S. R.: Interactive Grid Generation Program for CAP-TSD, NASA TM 102705, October 1990.
48. Batina, J. T.: Unsteady Transonic Flow Calculations for Wing-Fuselage Configurations, AIAA Paper No. 86-0862, May 1986; also *Journal of Aircraft*, Vol. 23, December 1986, pp.897-903.
49. Mason, W. H.; Mackenzie, C.; Stern, M.; Ballhaus, W. F.; and Frick, J.: An Automated Procedure for Computing the Three-Dimensional Transonic Flow Over Wing-Body Combinations, Including Viscous Effects, AFFDL-TR-77-122, Vol. I, February 1978.



(a) Streamwise mass flux f_1 .



(b) Derivative of streamwise mass flux $\partial f_1 / \partial \phi_x$.

Figure 1 - Mass flux quantities as functions of streamwise velocity u for various small perturbation theories at $M_\infty = 0.72$.

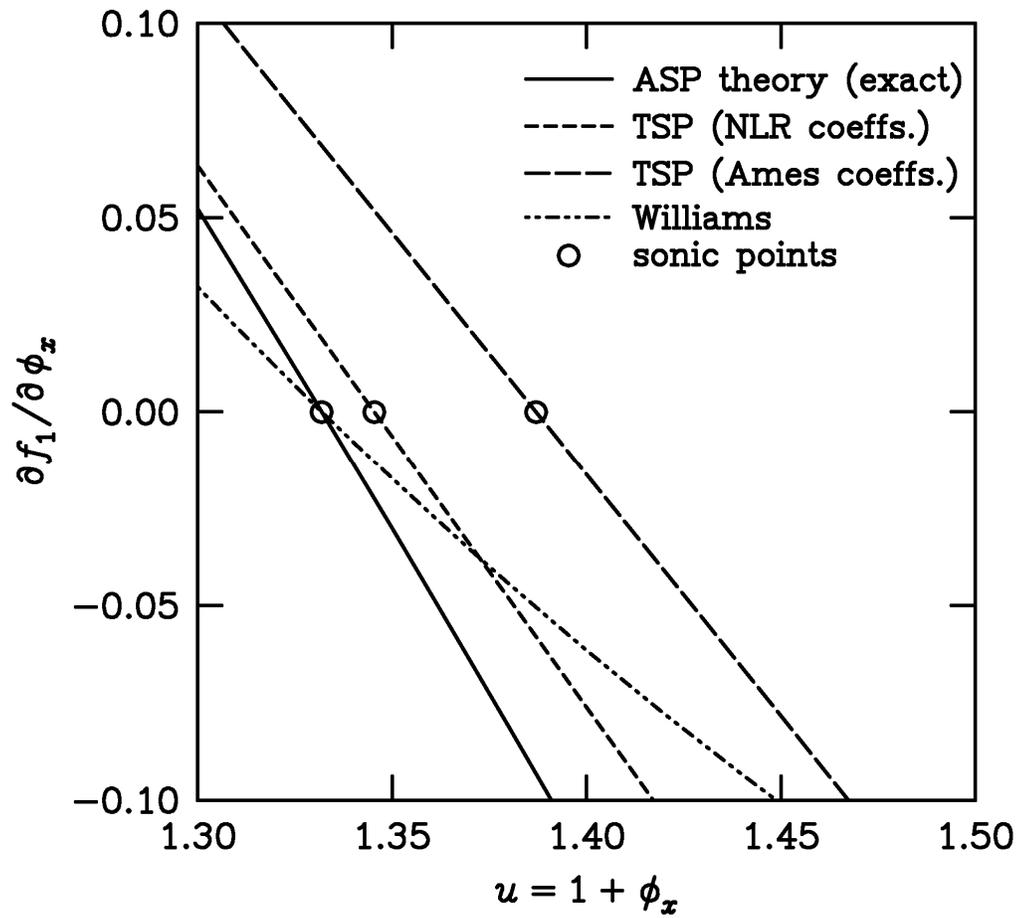


Figure 2 - Sonic point ($\partial f_1 / \partial \phi_x = 0$) as a function of streamwise velocity u for various small perturbation theories at $M_\infty = 0.72$.

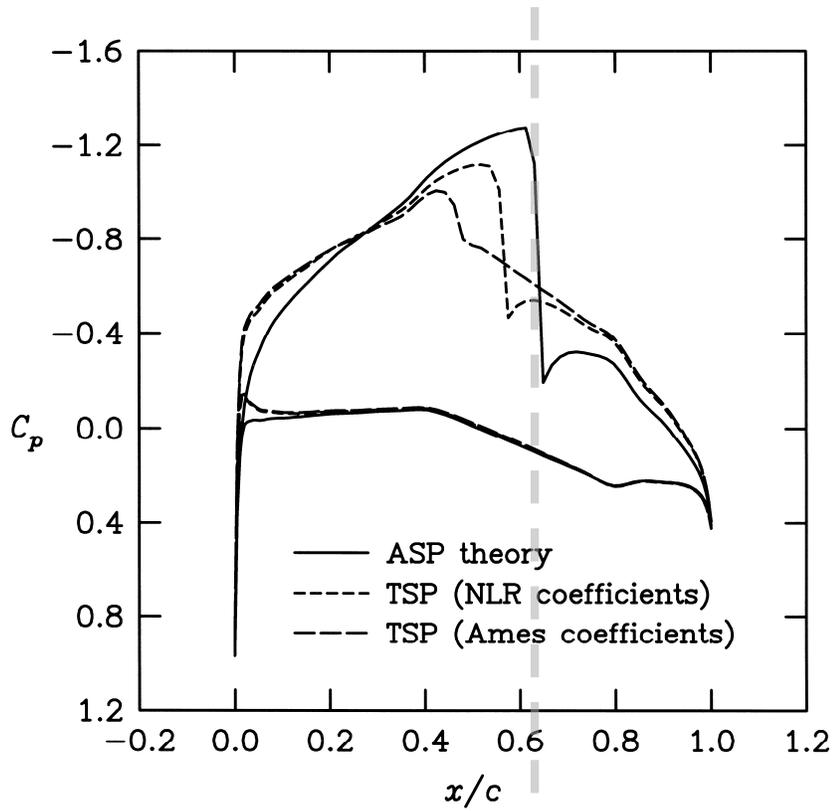


Figure 3 - Effects of small perturbation streamwise flux on pressure coefficient distribution for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$.

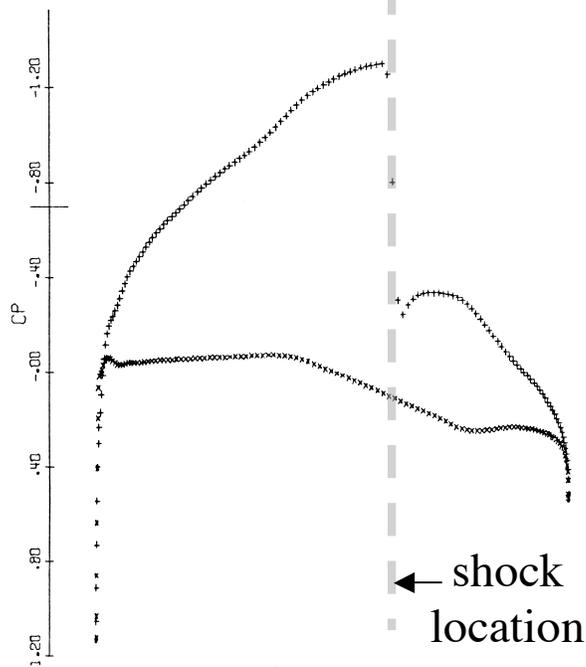


Figure 4 – Full potential pressure coefficient distribution computed by Jameson²⁶ for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$.

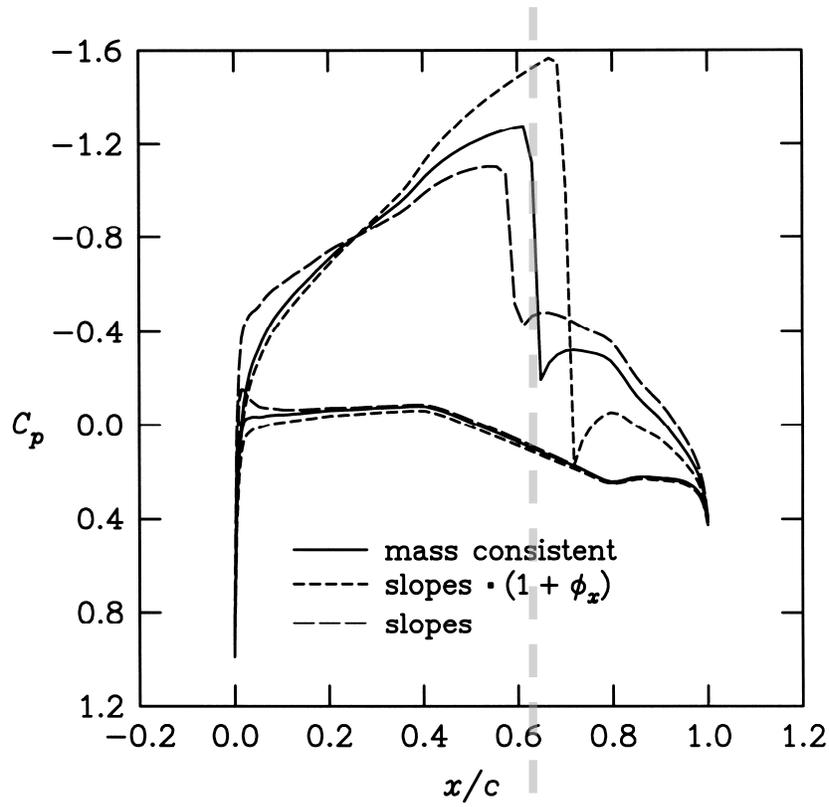


Figure 5 - Effects of small perturbation surface boundary condition on pressure coefficient distribution for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$.

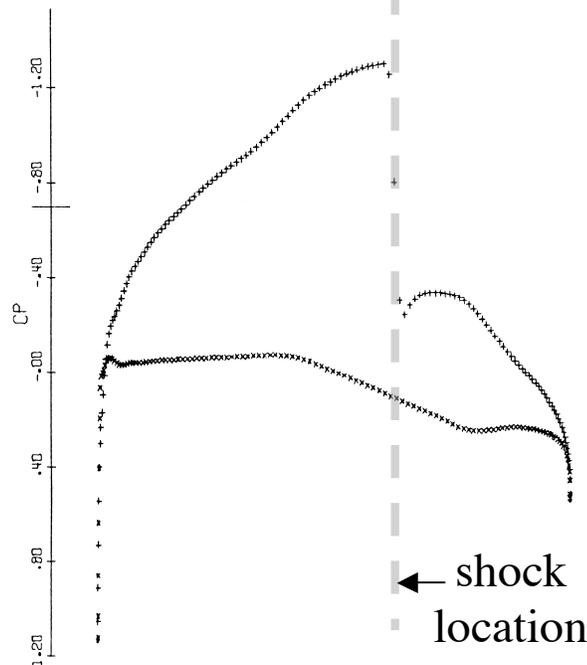


Figure 6 – Full potential pressure coefficient distribution computed by Jameson²⁶ for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$ (same as Figure 7).

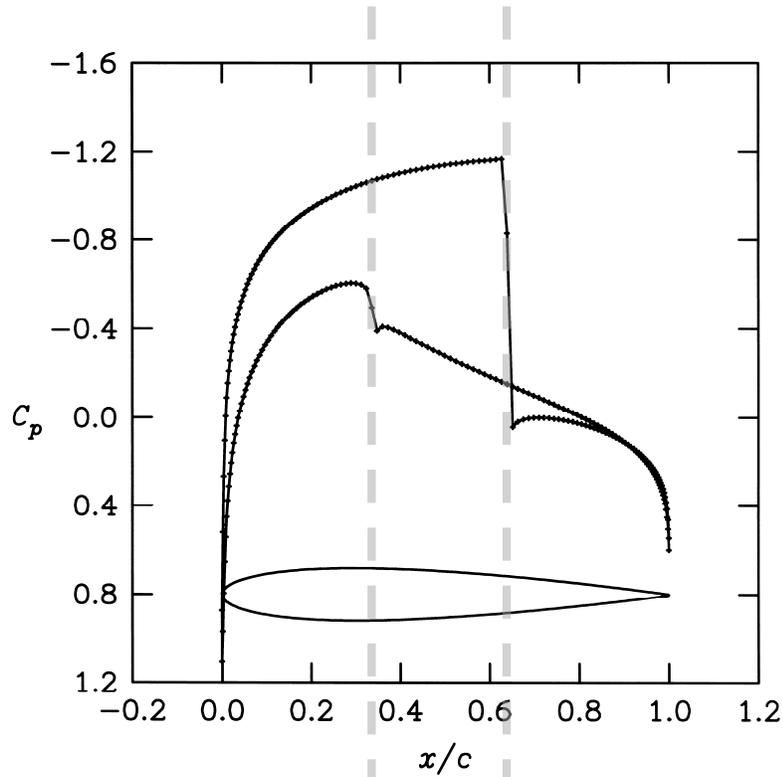


Figure 7 - Pressure coefficient distribution for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha = 1.25^\circ$ computed using ASP theory including entropy and vorticity.

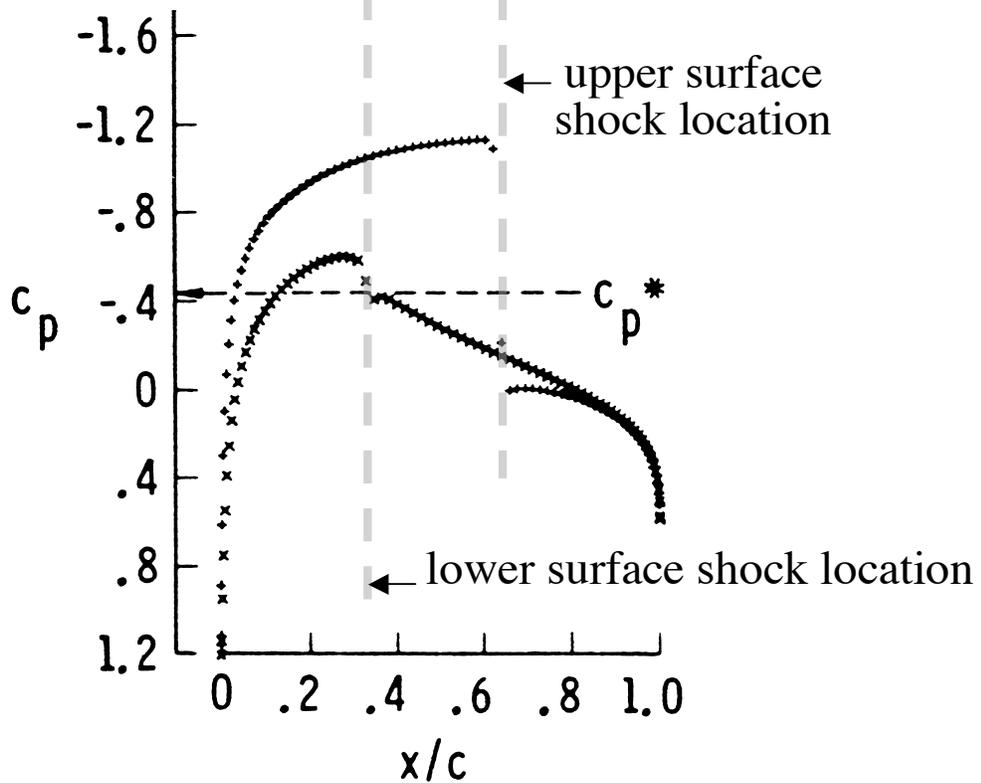


Figure 8 – Euler (CFL3D) pressure coefficient distribution computed by Anderson, et al.²⁷ for the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha = 1.25^\circ$.

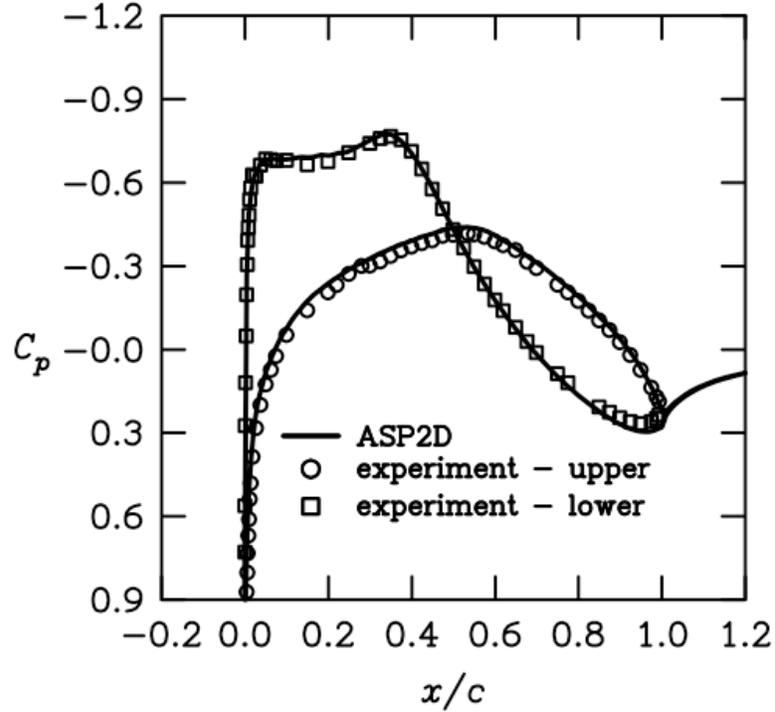


Figure 9 – ASP pressure coefficient distribution for the RAE 2822 airfoil at $M_\infty = 0.676$, $\alpha = -2.25^\circ$, and $Re = 5.7 \times 10^6$ (attached flow case).

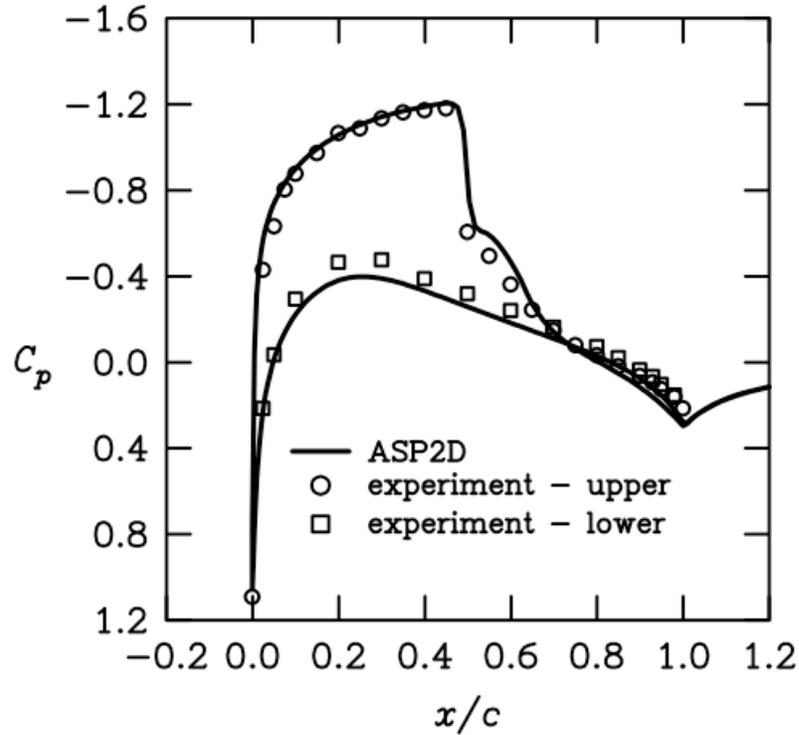
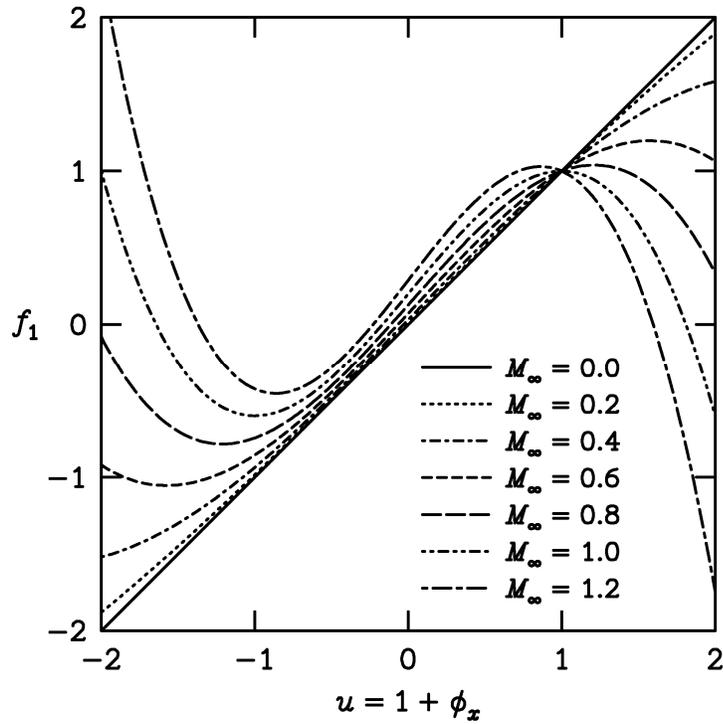
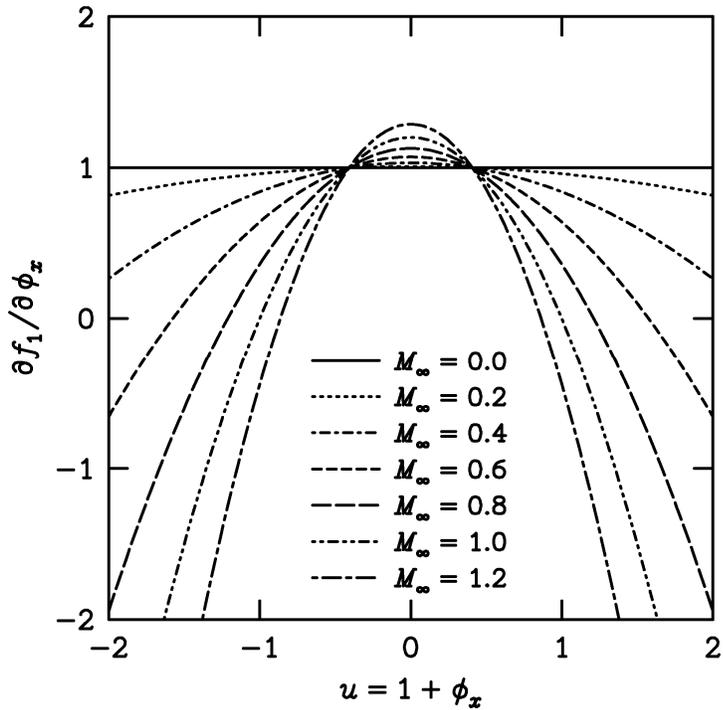


Figure 10 – ASP pressure coefficient distribution for the NACA 0012 airfoil at $M_\infty = 0.775$, $\alpha = 2.05^\circ$, and $Re = 10^7$ (separated flow case).



(a) Streamwise mass flux f_1 .



(b) Derivative of streamwise mass flux $\partial f_1 / \partial \phi_x$.

Figure 11 – ASP mass flux quantities as functions of streamwise velocity u for a range of values of freestream Mach number.

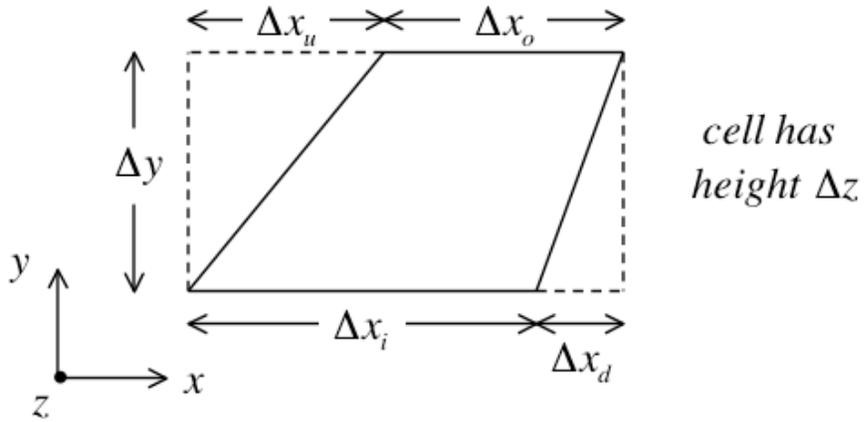


Figure 12 – Definition of control volume (cell) with associated dimensions.

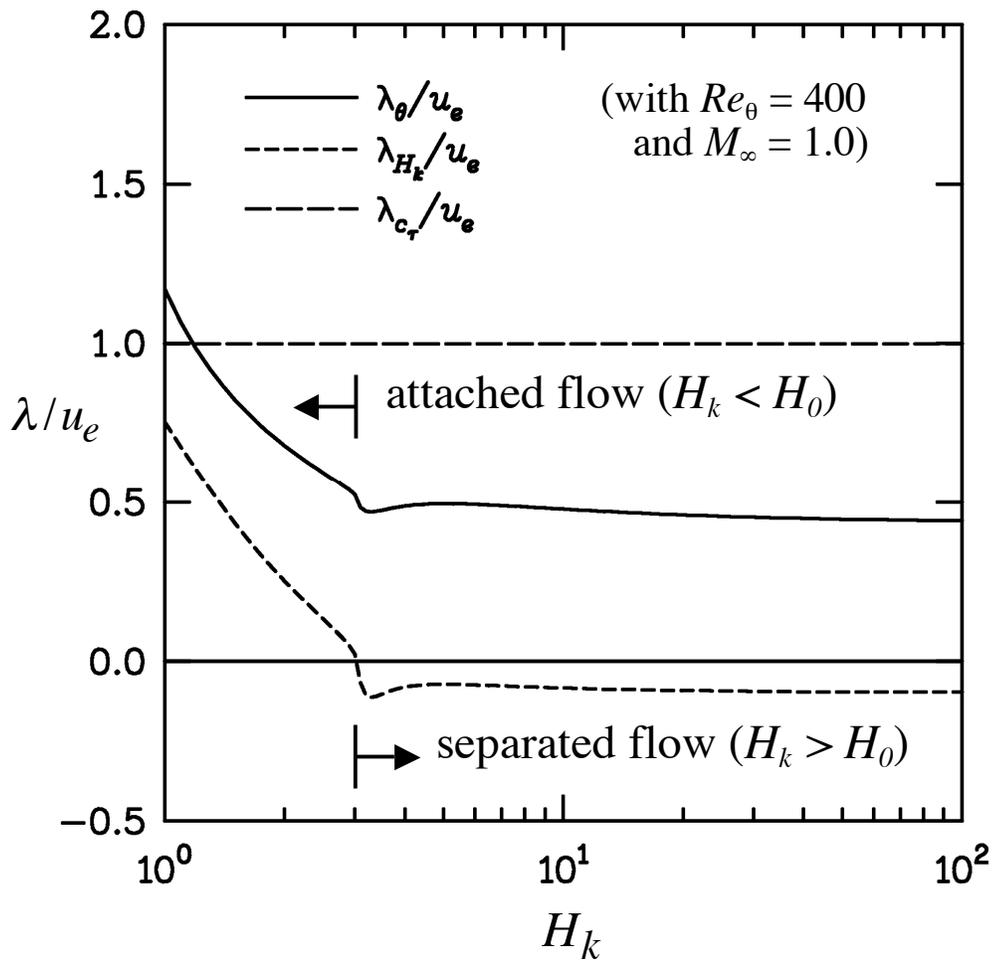
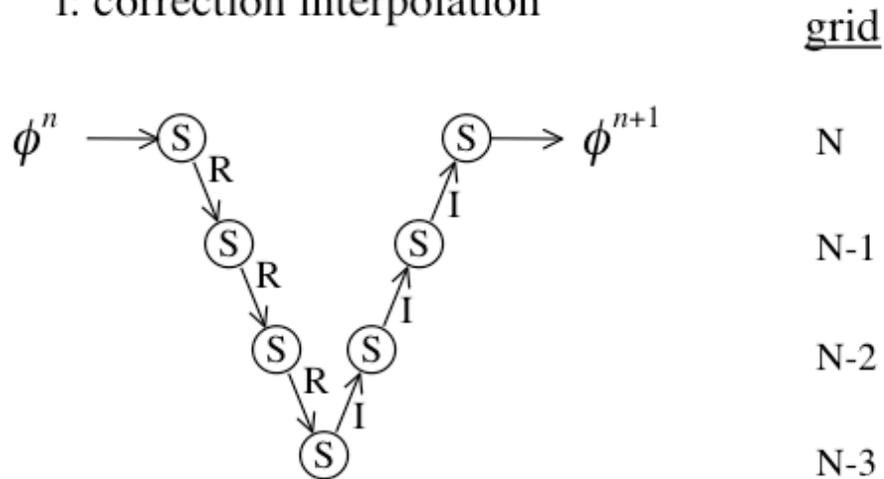


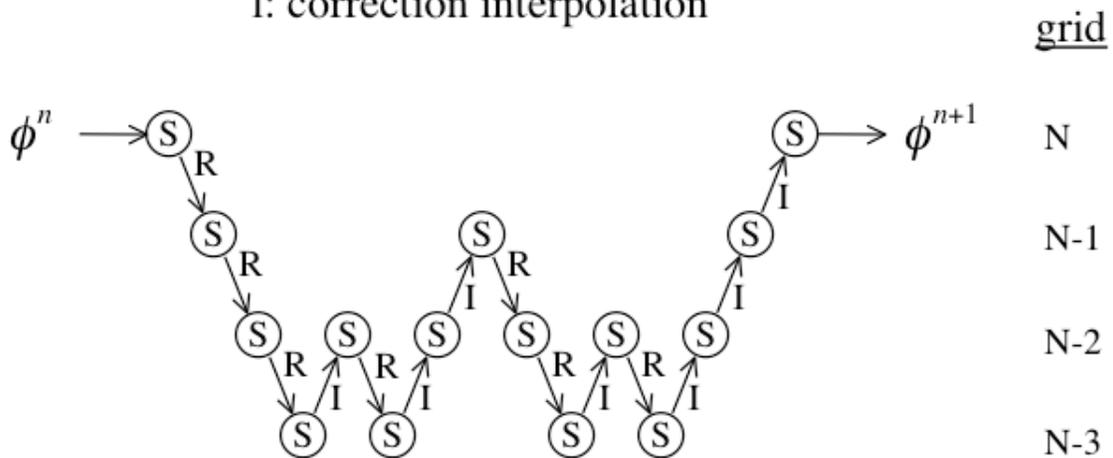
Figure 13 – Normalized eigenvalues of the integral boundary layer and lag equations as functions of the incompressible shape parameter H_k for $Re_\theta = 400$ and $M_\infty = 1.0$.

S: smoothing (AF2 algorithm)
 R: restrict ϕ and residual R
 I: correction interpolation



(a) Details of V-cycle calculations.

S: smoothing (AF2 algorithm)
 R: restrict ϕ and residual R
 I: correction interpolation



(b) Details of W-cycle calculations.

Figure 14 – ASP3D multigrid implementation.

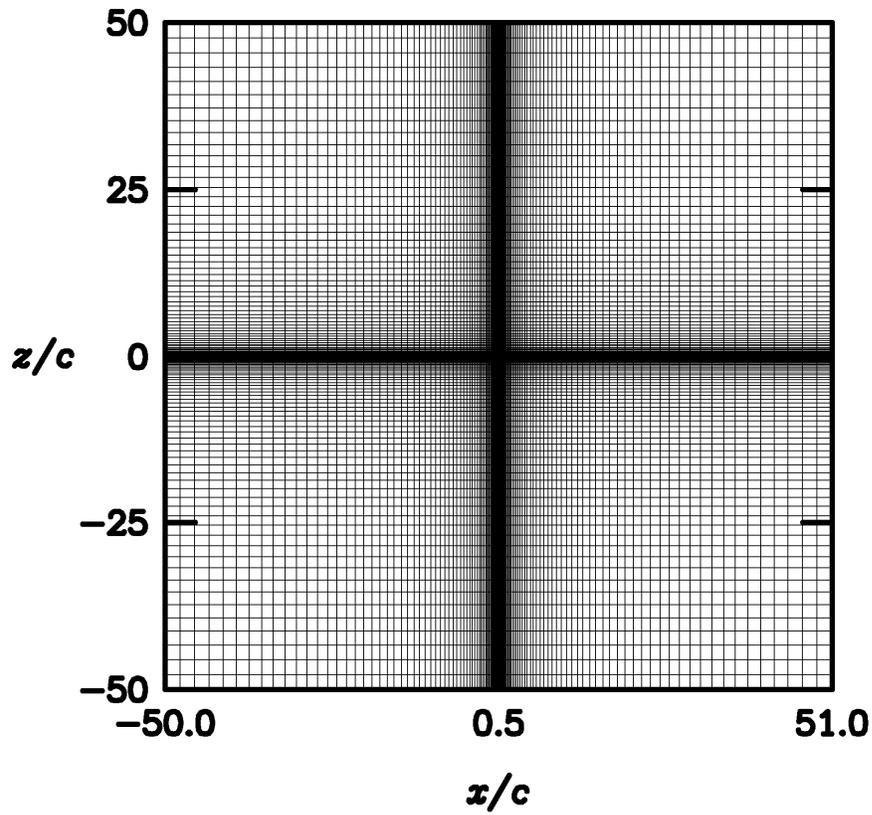
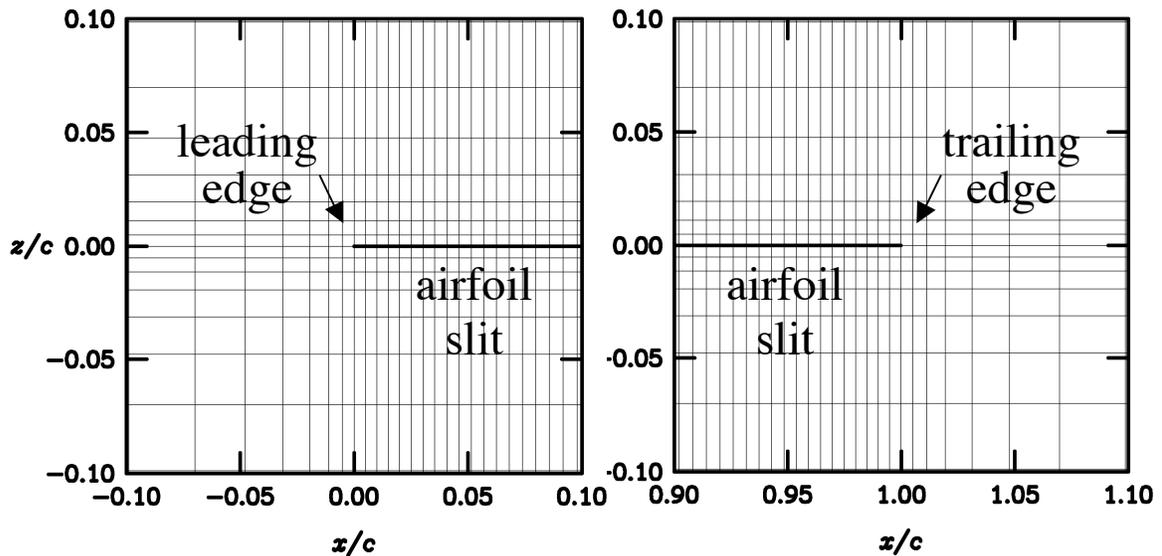


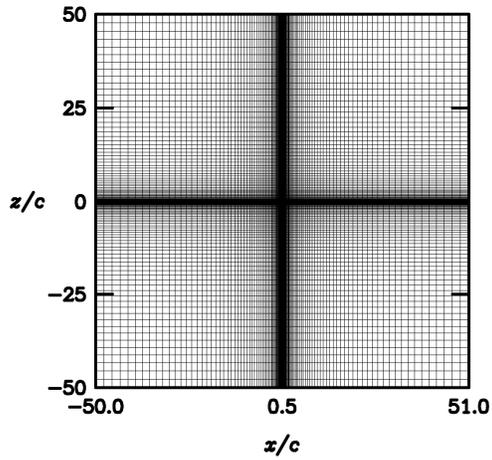
Figure 15 – Cartesian finite volume sectional mesh (257 x 129) with fifty-chord extent for airfoil applications with the ASP2D code.



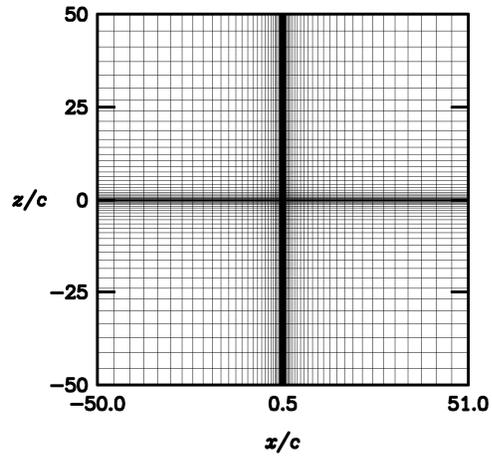
(a) Leading edge region.

(b) Trailing edge region.

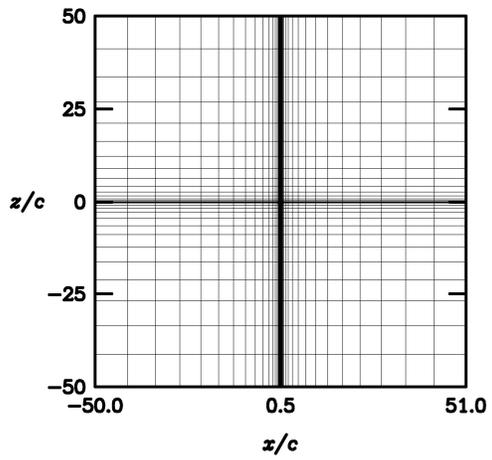
Figure 16 – Near field views of the Cartesian finite volume mesh for airfoil applications with the ASP2D code.



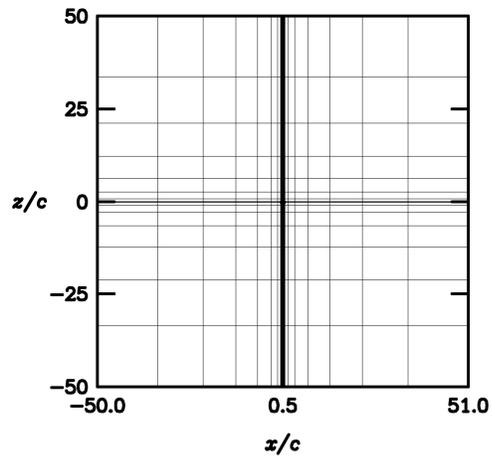
(a) Mesh 6 (257 x 129).



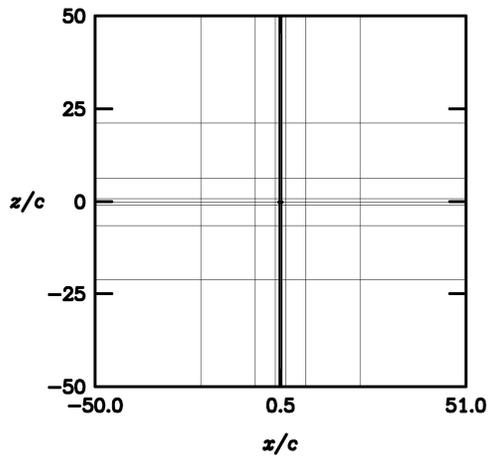
(b) Mesh 5 (129 x 65).



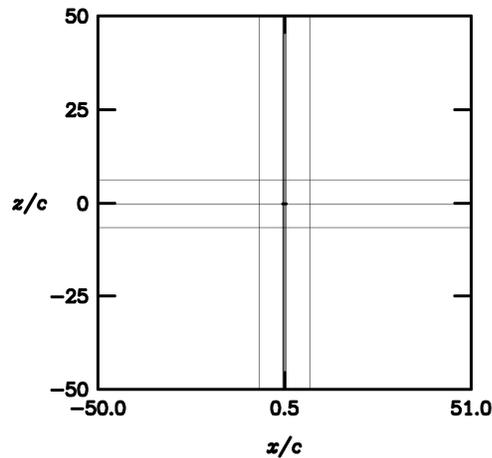
(c) Mesh 4 (65 x 33).



(d) Mesh 3 (33 x 17).

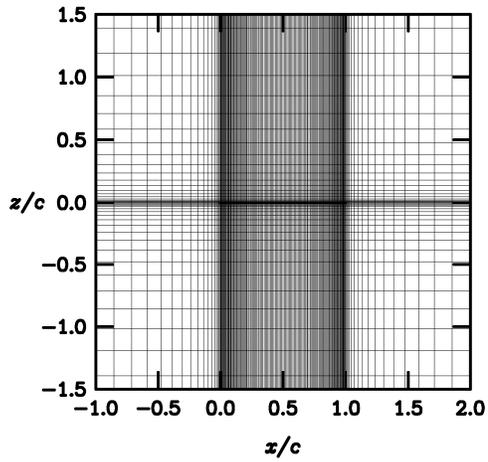


(e) Mesh 2 (17 x 9).

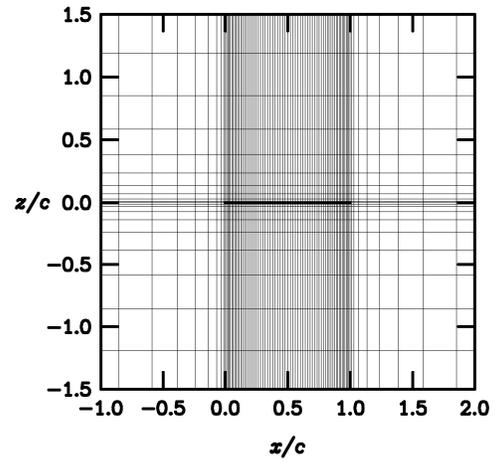


(f) Mesh 1 (9 x 5).

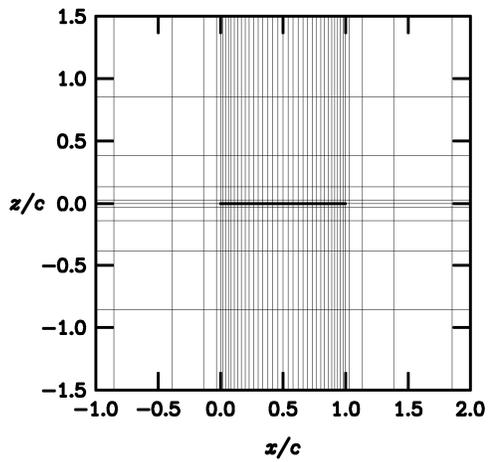
Figure 17 – Cartesian finite volume sectional meshes with fifty-chord extent for airfoil applications with the ASP2D multigrid capability.



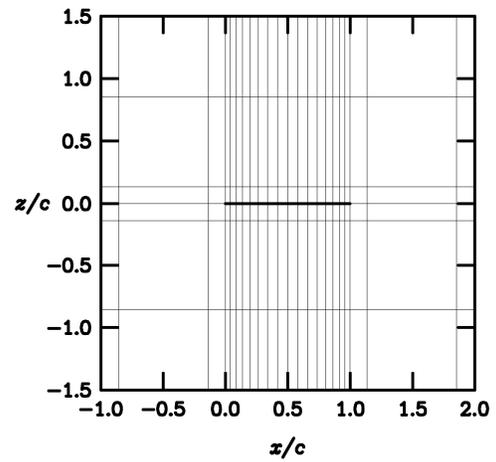
(a) Mesh 6 (257 x 129).



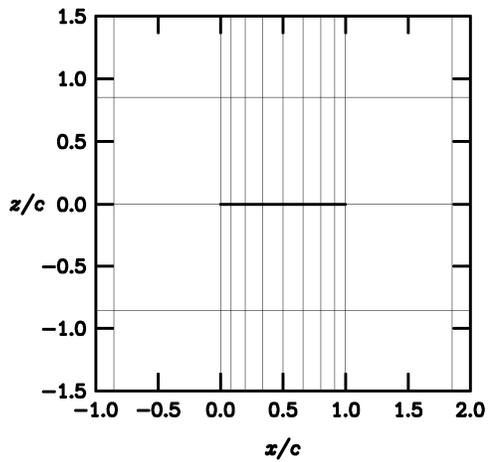
(b) Mesh 5 (129 x 65).



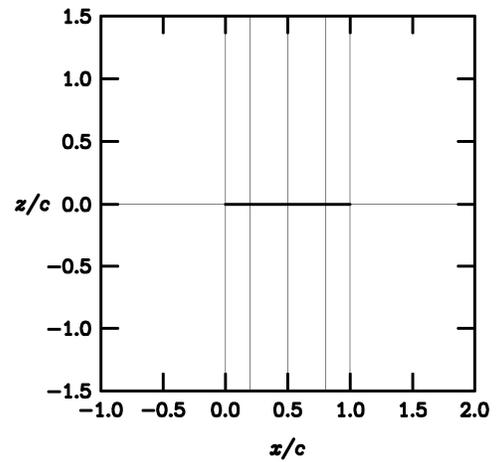
(c) Mesh 4 (65 x 33).



(d) Mesh 3 (33 x 17).

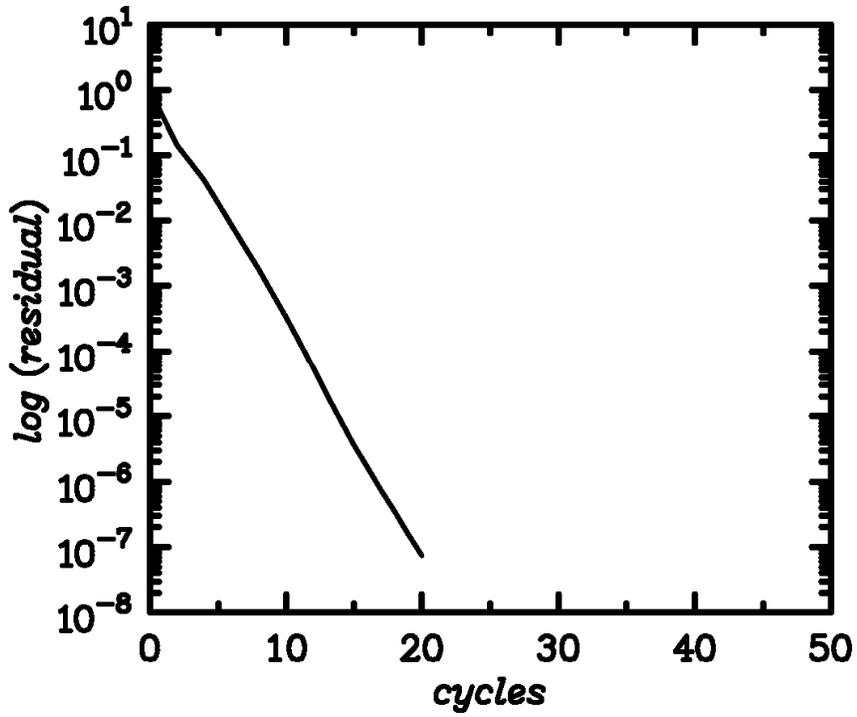


(e) Mesh 2 (17 x 9).

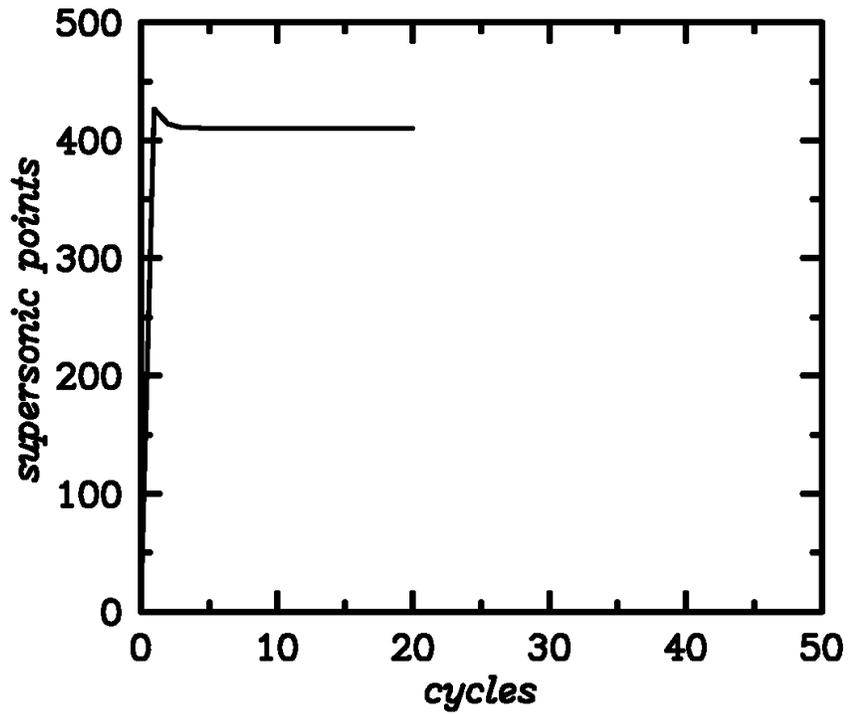


(f) Mesh 1 (9 x 5).

Figure 18 – Near field view of the Cartesian finite volume sectional meshes for airfoil applications with the ASP2D multigrid capability.



(a) Convergence history.



(b) Number of supersonic points.

Figure 19 – ASP2D multigrid calculations for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$.

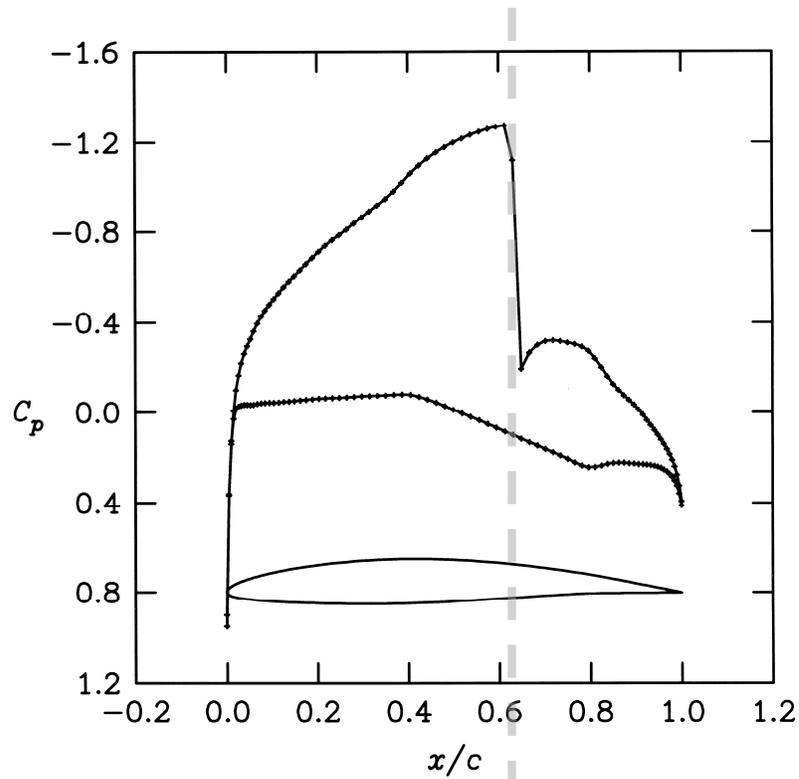


Figure 20 - ASP2D pressure coefficient distribution for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$.

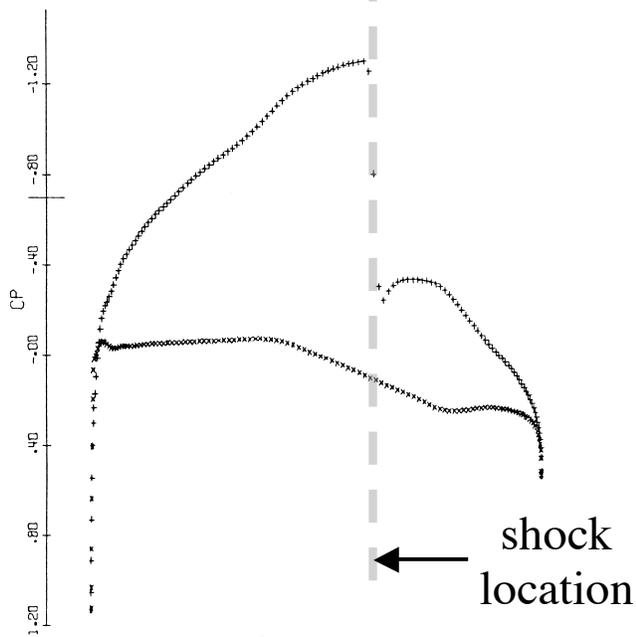
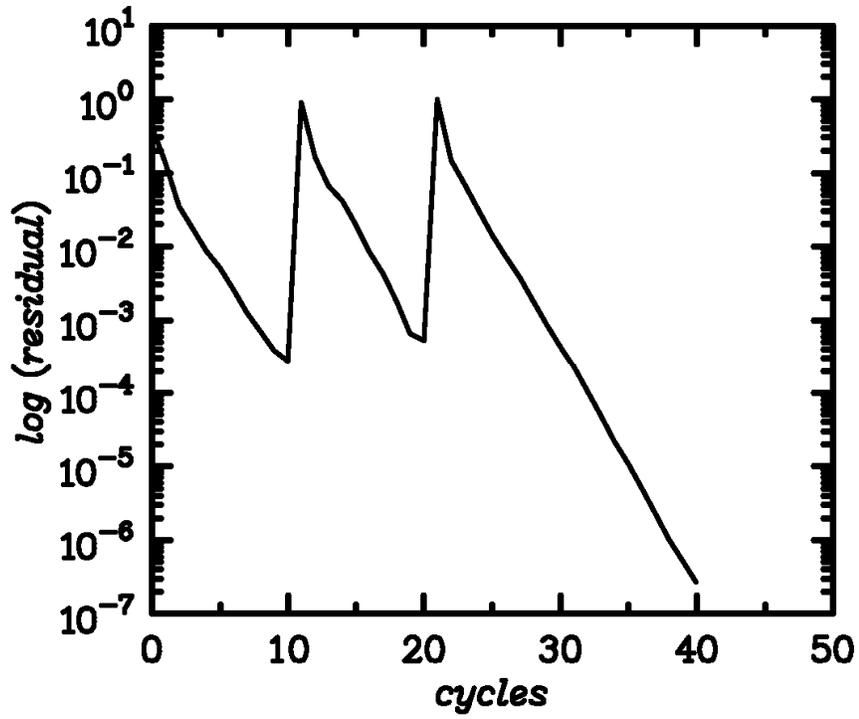
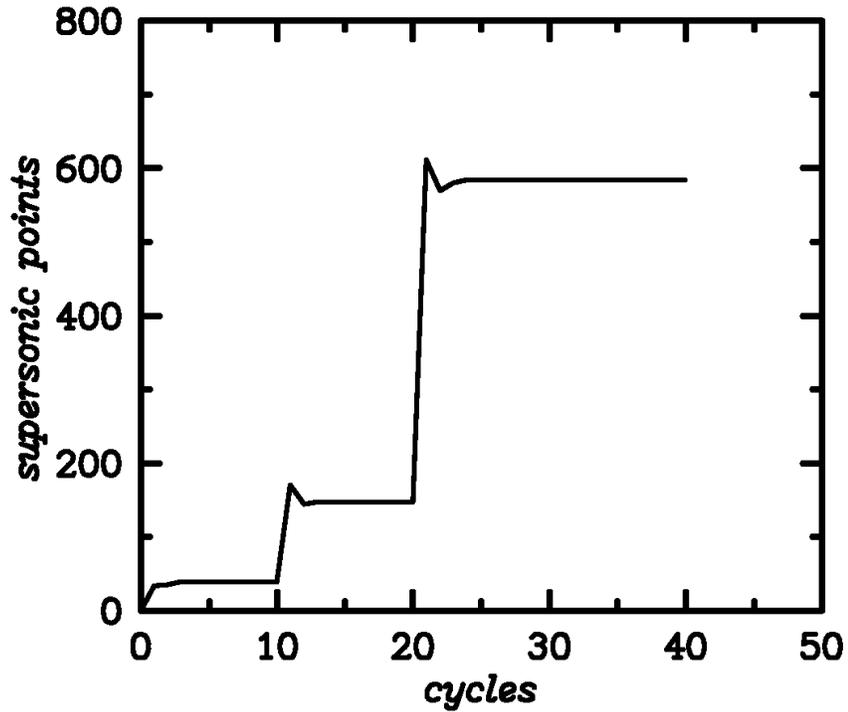


Figure 21 – Full potential pressure coefficient distribution computed by Jameson²⁶ for the NACA 64A410 airfoil at $M_\infty = 0.72$ and $\alpha = 0^\circ$.



(a) Convergence history.



(b) Number of supersonic points.

Figure 22 – ASP2D full multigrid calculations for the NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2^\circ$.

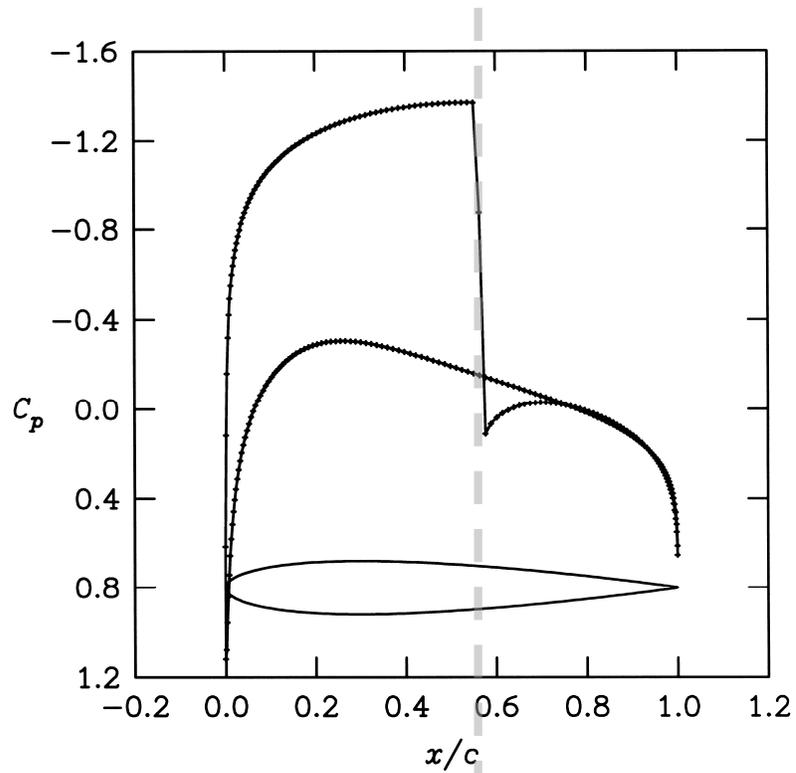


Figure 23 - ASP2D pressure coefficient distribution for the NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2^\circ$.

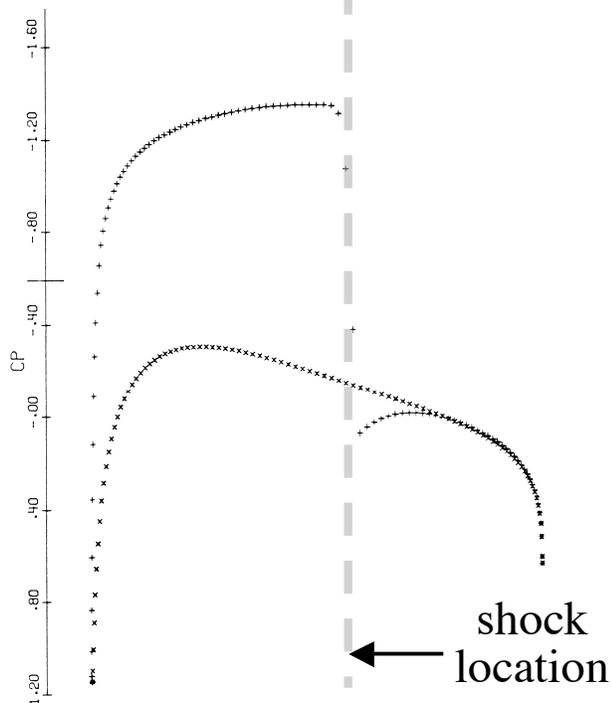
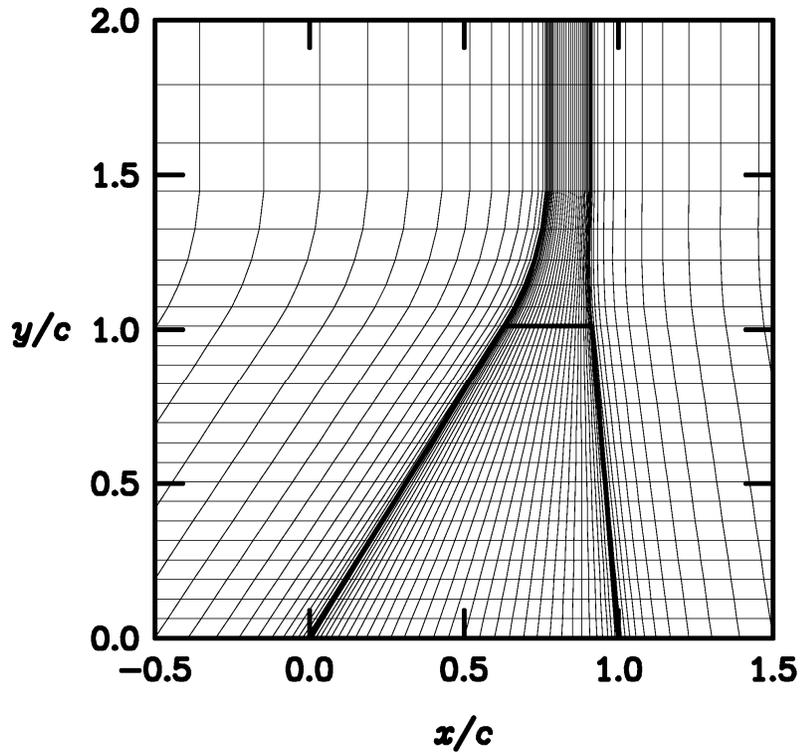
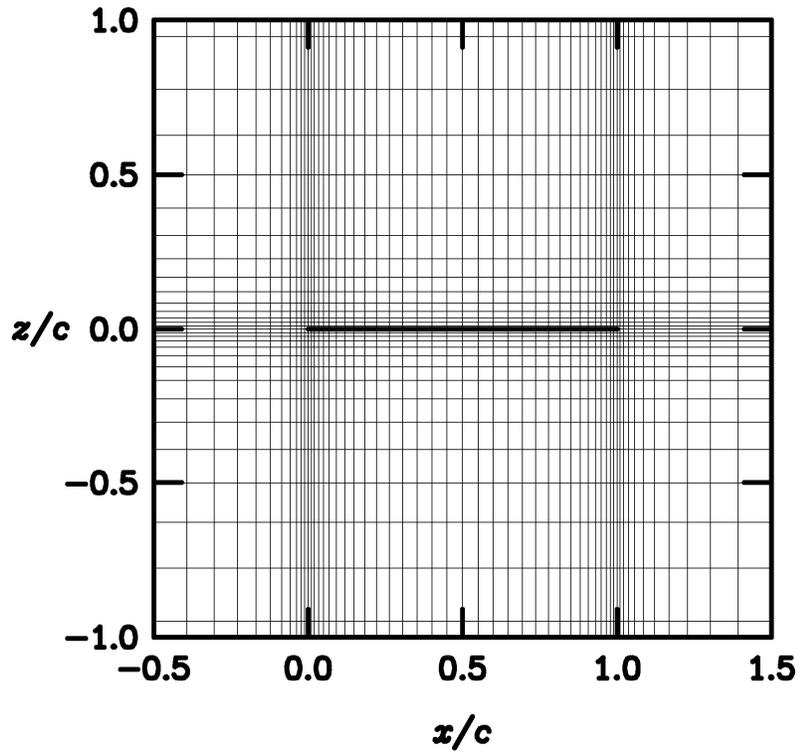


Figure 24 – Full potential pressure coefficient distribution computed by Jameson⁴⁰ for the NACA 0012 airfoil at $M_\infty = 0.75$ and $\alpha = 2^\circ$.

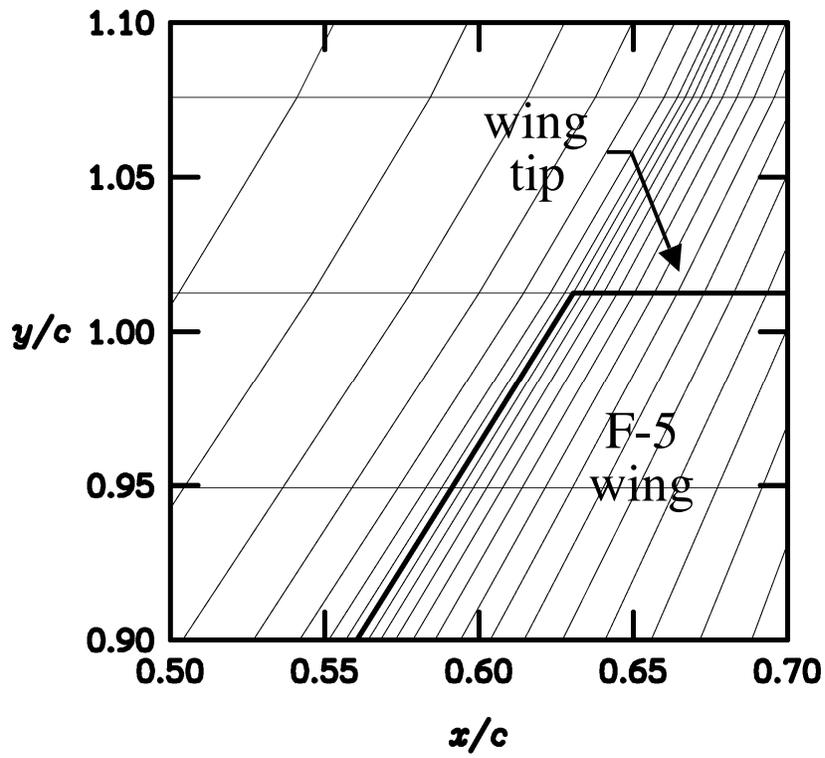


(a) Near field view of 97 x 25 planform mesh.

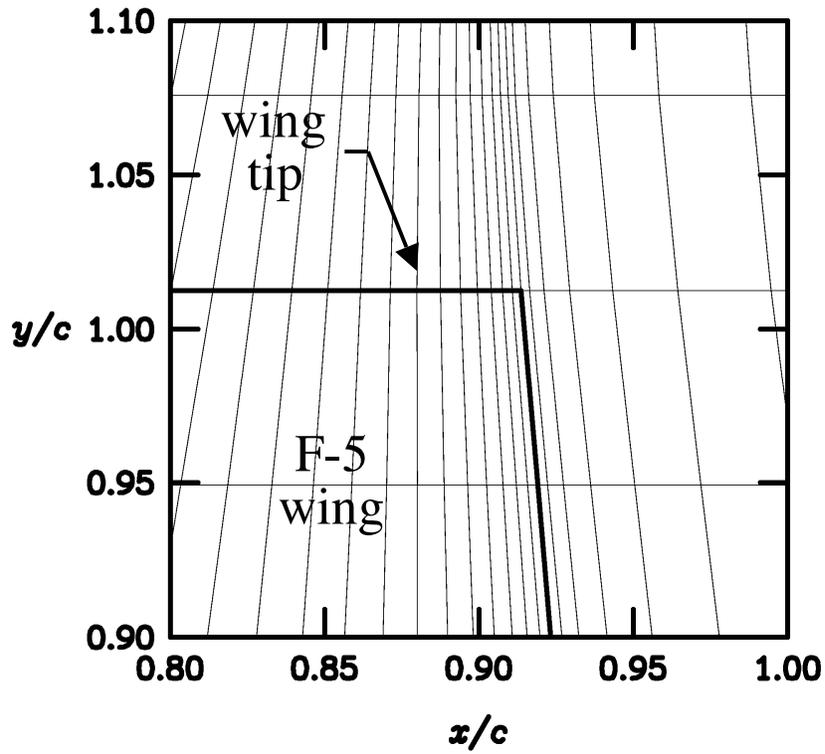


(b) Near field view of 97 x 65 root sectional mesh.

Figure 25 – Finite volume meshes for the F-5 fighter wing.

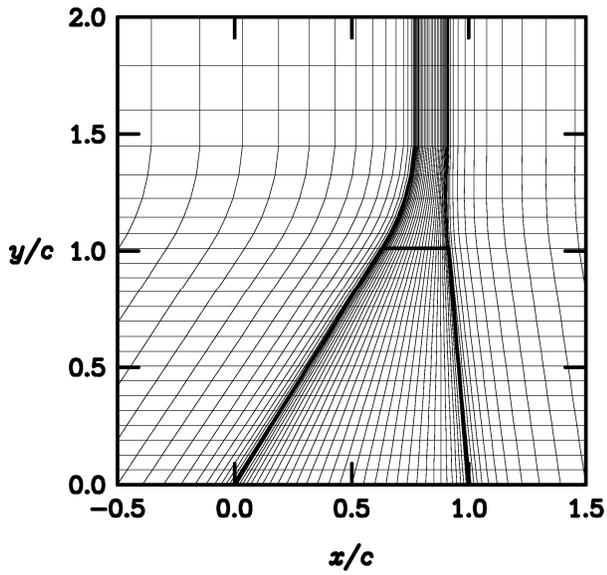


(a) Wing tip leading edge region.

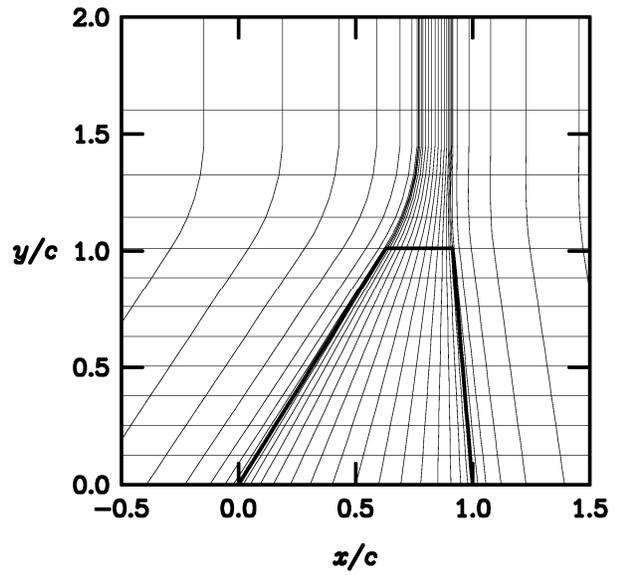


(b) Wing tip trailing edge region.

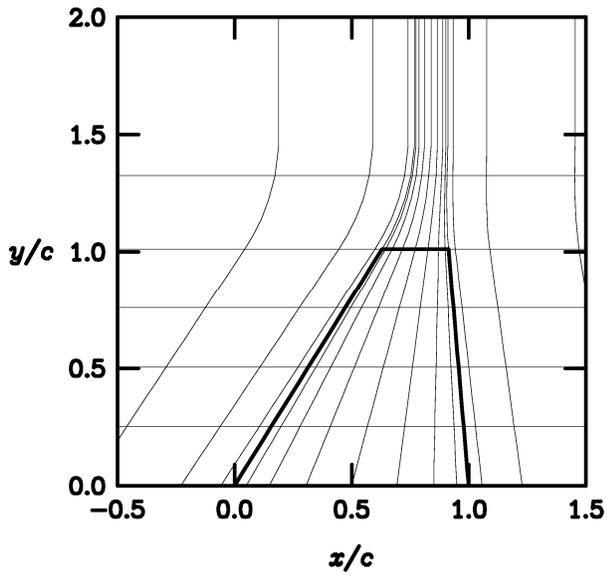
Figure 26 – Near field views of the planform mesh of the F-5 fighter wing.



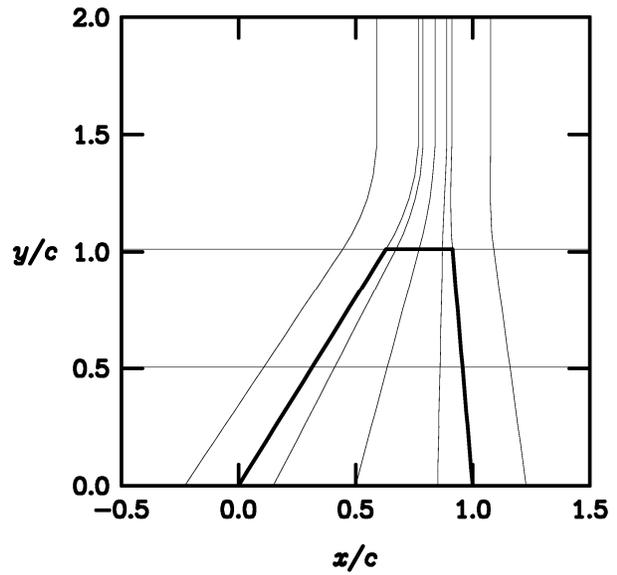
(a) Mesh 4 (97 x 25).



(b) Mesh 3 (49 x 13).

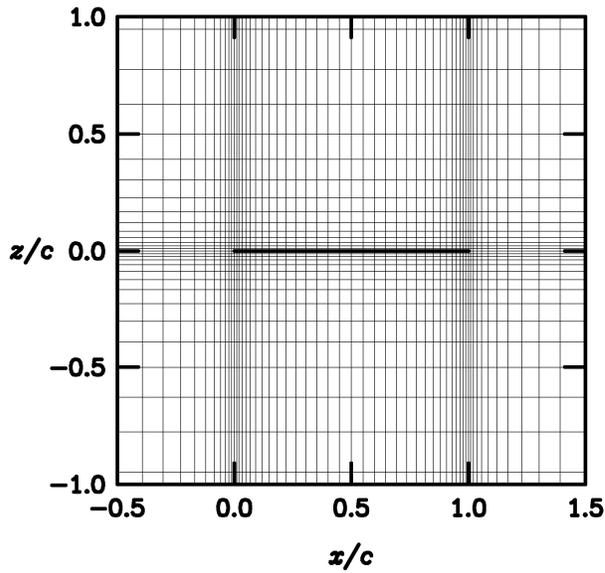


(c) Mesh 2 (25 x 7).

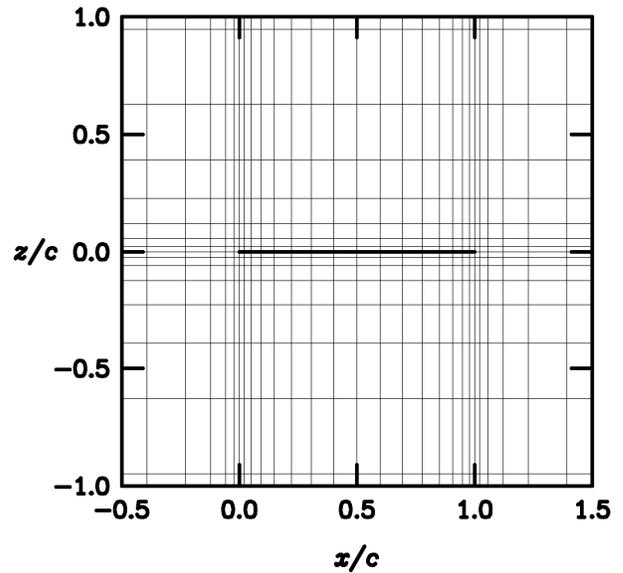


(d) Mesh 1 (13 x 4).

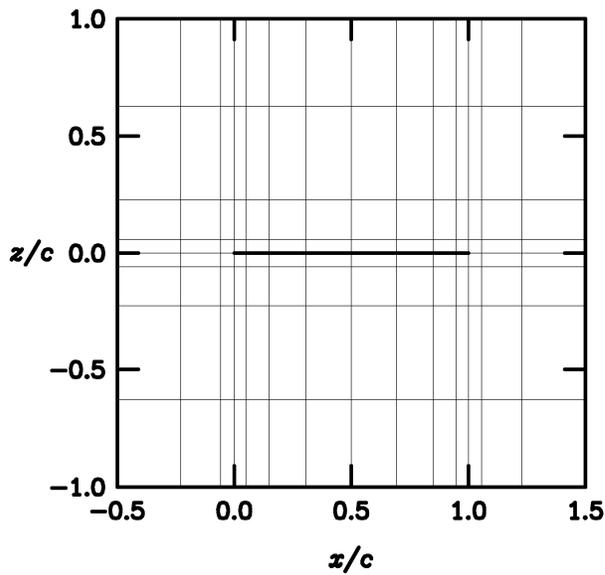
Figure 27 – Near field view of finite volume planform meshes for the F-5 fighter wing for ASP3D multigrid calculations.



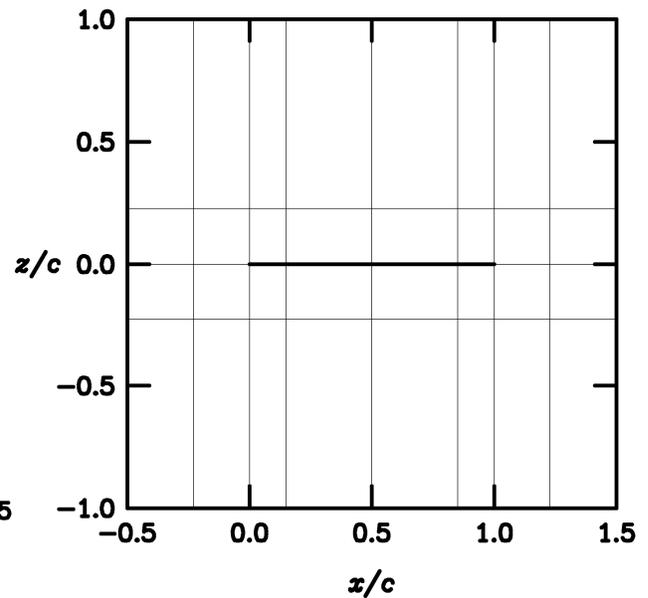
(a) Mesh 4 (97 x 65).



(b) Mesh 3 (49 x 33).

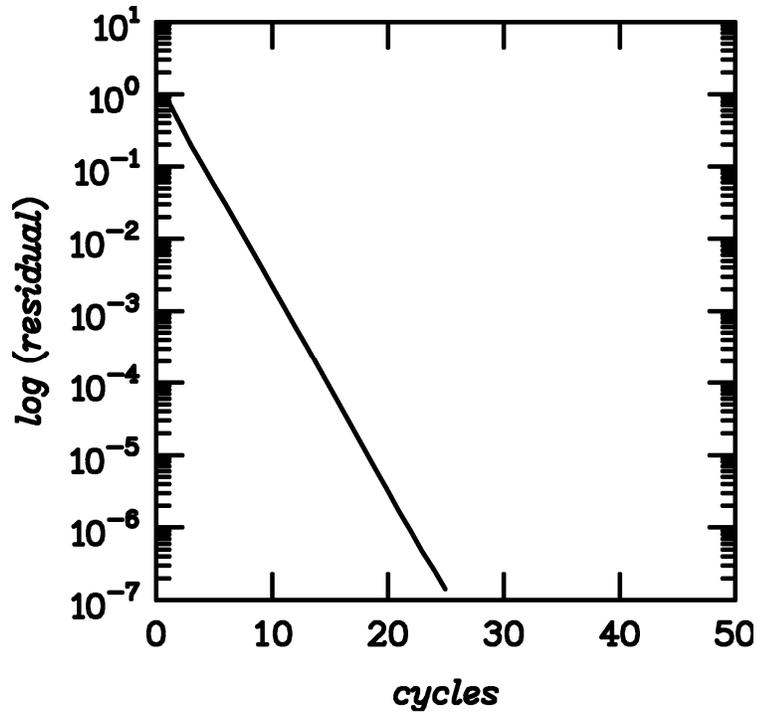


(c) Mesh 2 (25 x 17).

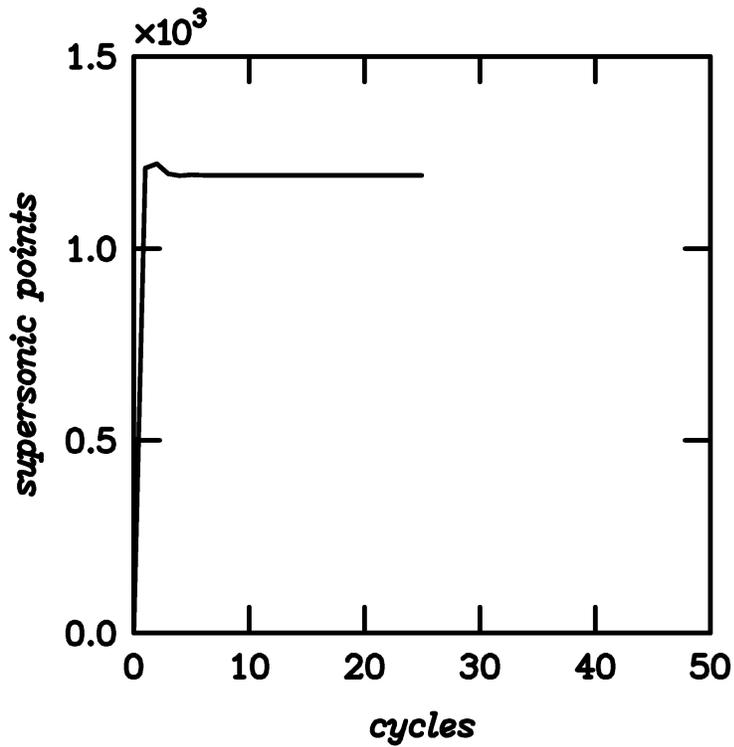


(d) Mesh 1 (13 x 9).

Figure 28 – Near field view of finite volume root sectional meshes for the F-5 fighter wing for ASP3D multigrid calculations.

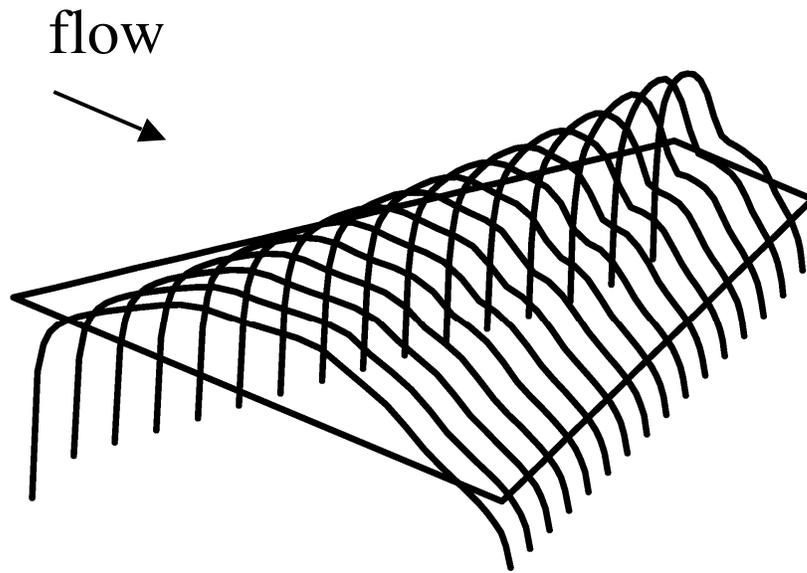


(a) Convergence history.

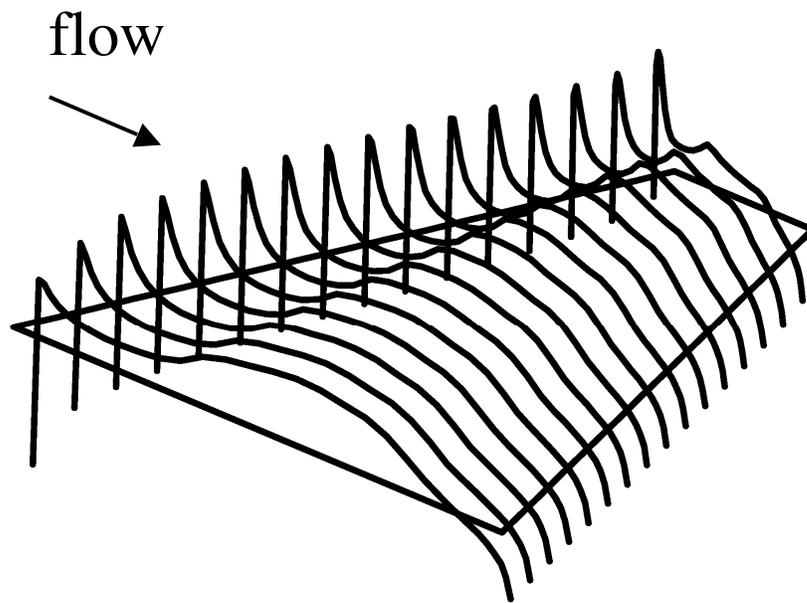


(b) Number of supersonic points.

Figure 29 – ASP3D multigrid calculations for the F-5 fighter wing at $M_\infty = 0.897$ and $\alpha = -0.004^\circ$.

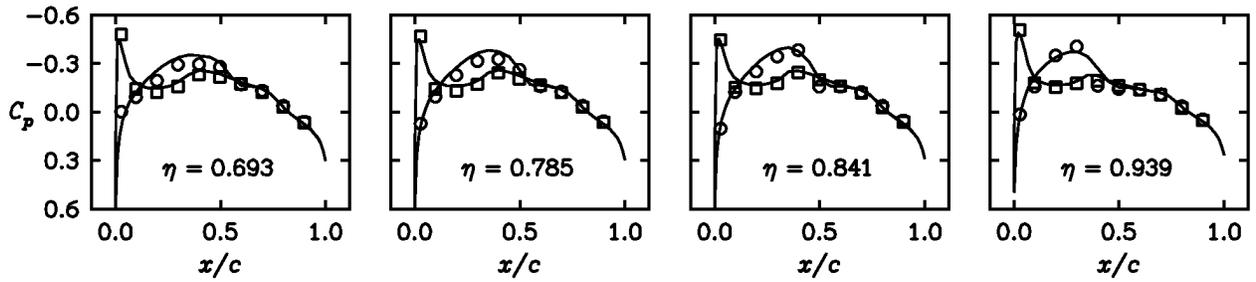
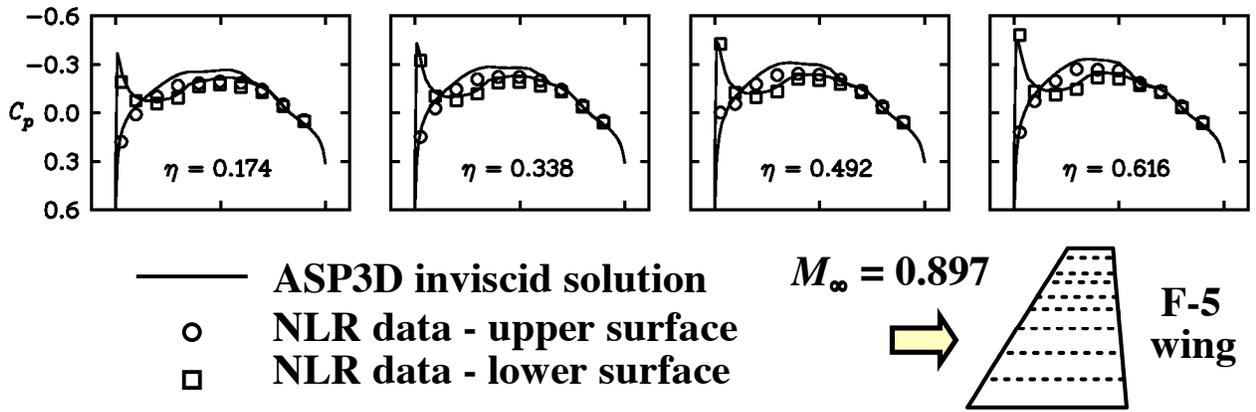


(a) Upper surface.

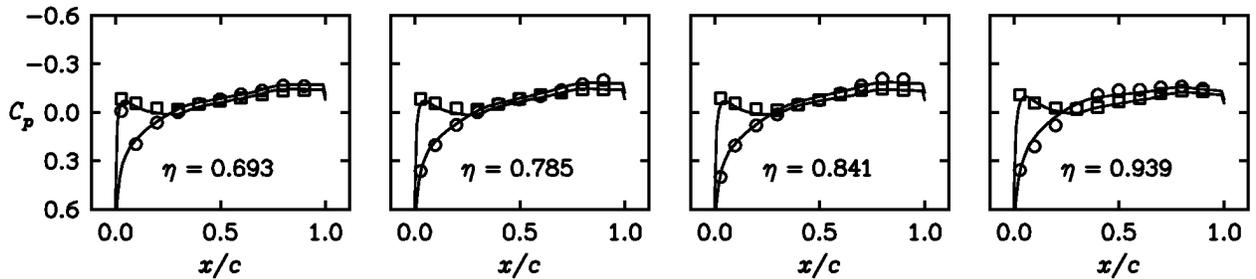
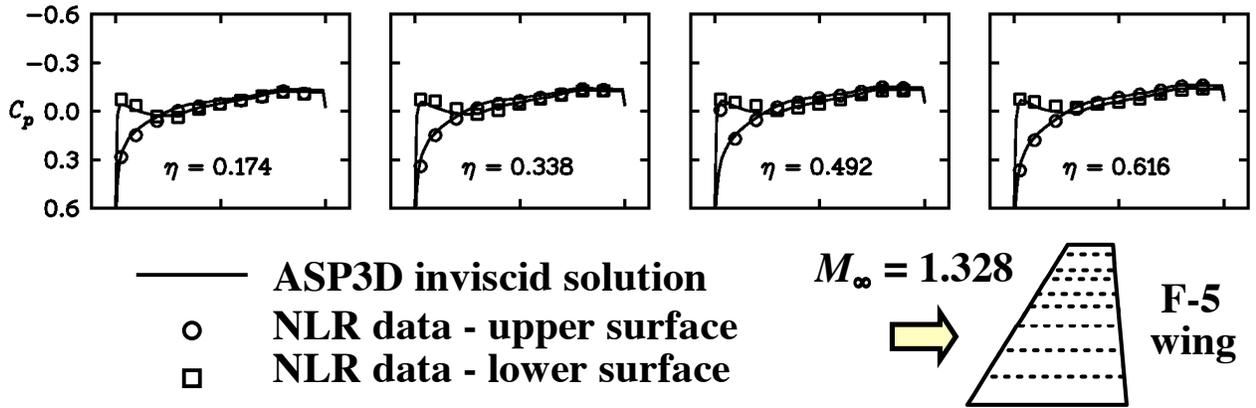


(b) Lower surface.

Figure 30 – ASP3D pressure coefficient distributions on the F-5 fighter wing at $M_\infty = 0.897$ and $\alpha = -0.004^\circ$.

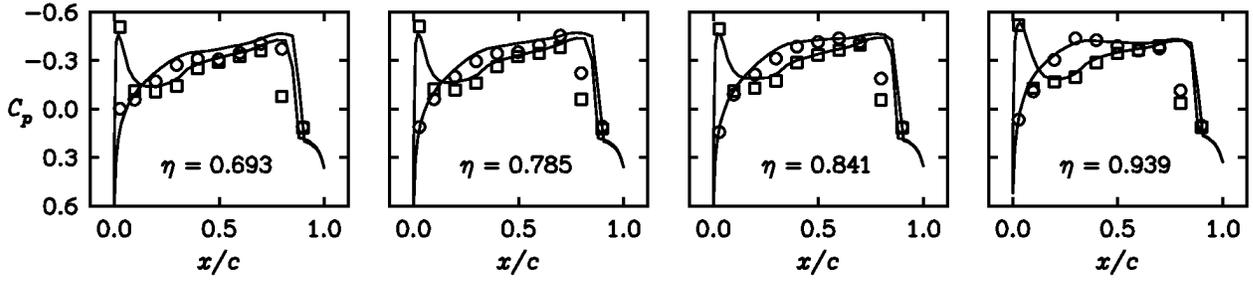
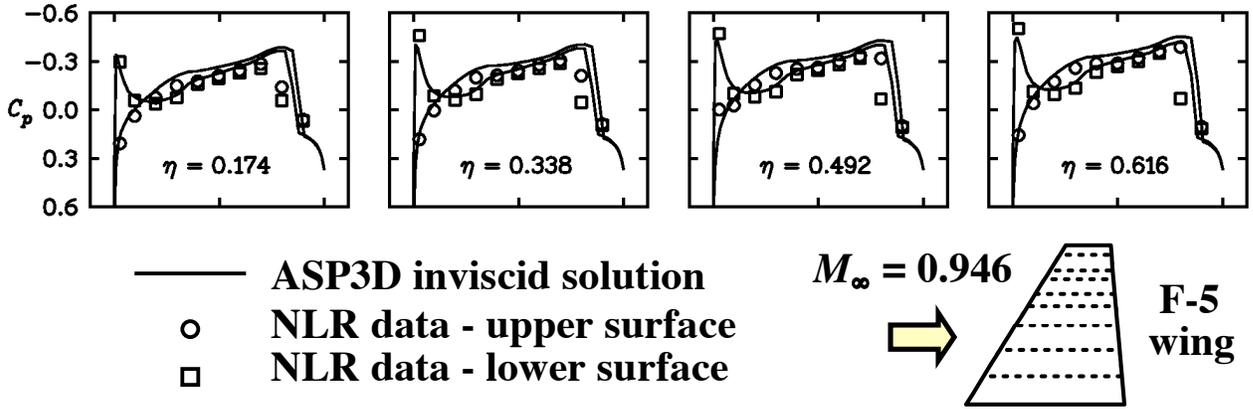


(a) $M_\infty = 0.897$ (subsonic freestream) and $\alpha = -0.004^\circ$.

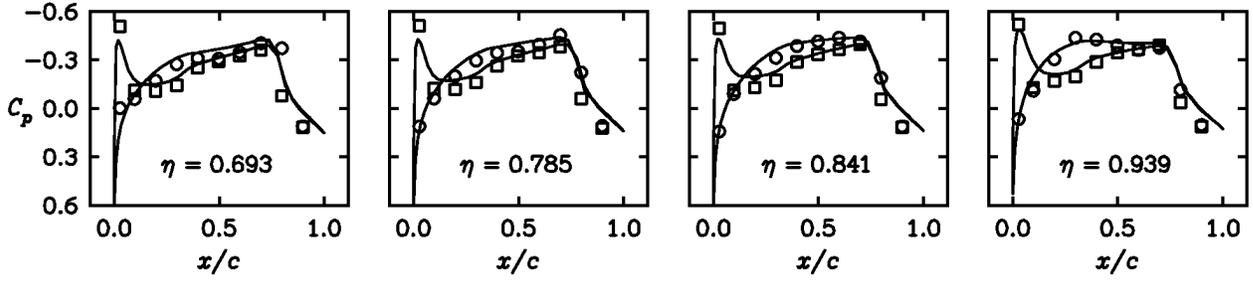
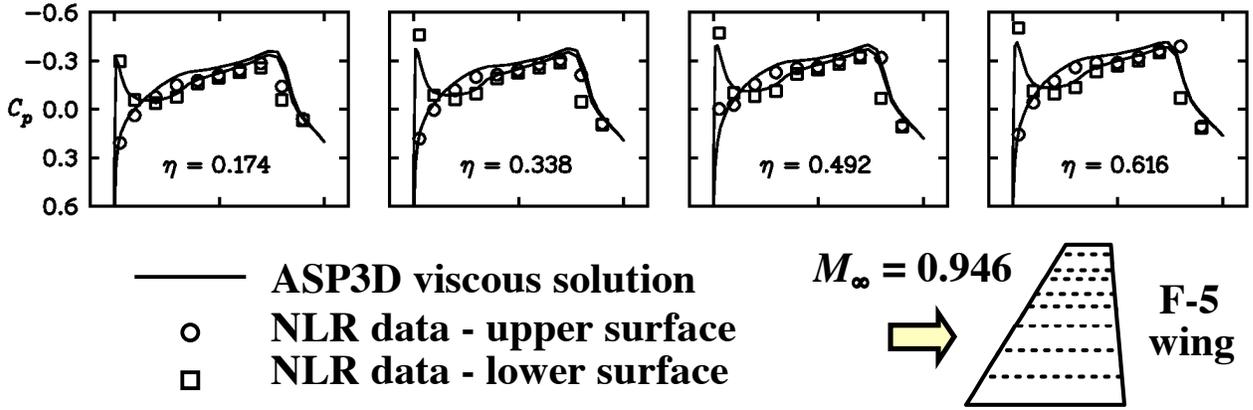


(b) $M_\infty = 1.328$ (supersonic freestream) and $\alpha = -0.005^\circ$.

Figure 31 – ASP3D inviscid pressure coefficient comparisons with NLR experimental data⁴⁵ for the F-5 fighter wing.

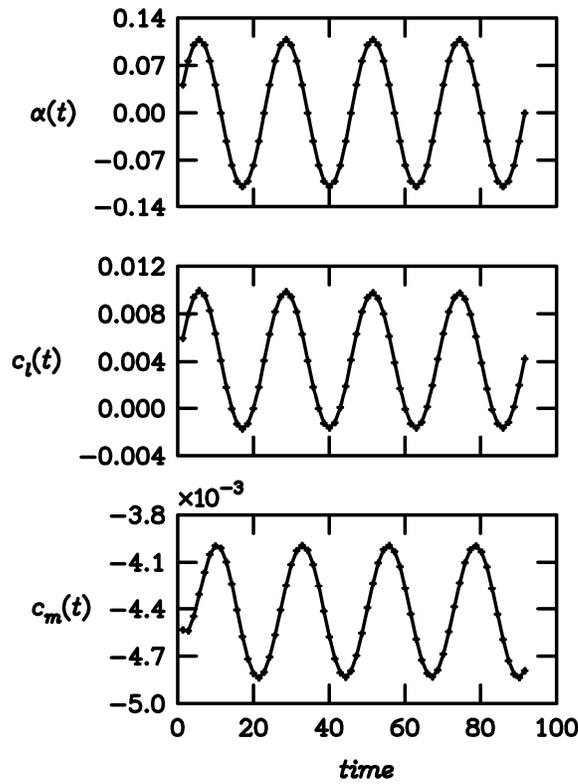


(a) Inviscid calculation.

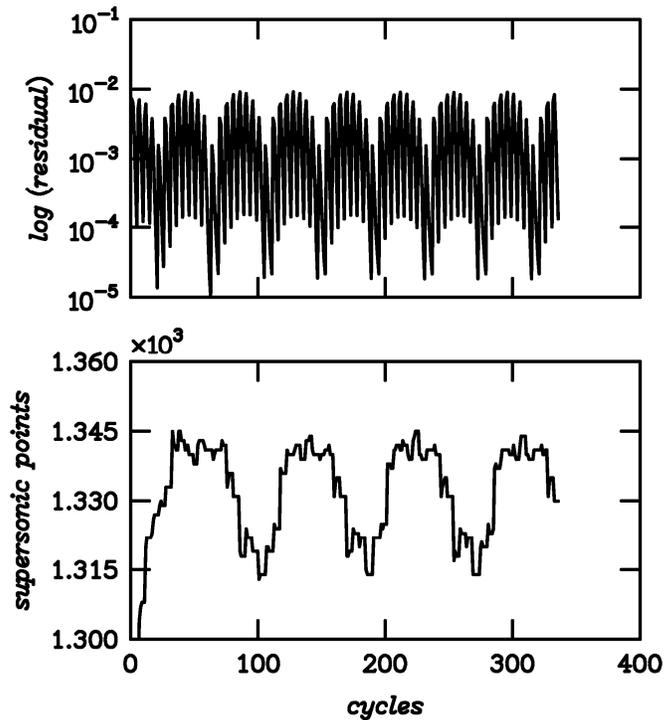


(b) Viscous calculation.

Figure 32 – ASP3D pressure coefficient comparisons with NLR experimental data⁴⁵ for the F-5 fighter wing at $M_\infty = 0.946$, $\alpha = -0.004^\circ$, and $Re = 5.89 \times 10^6$.

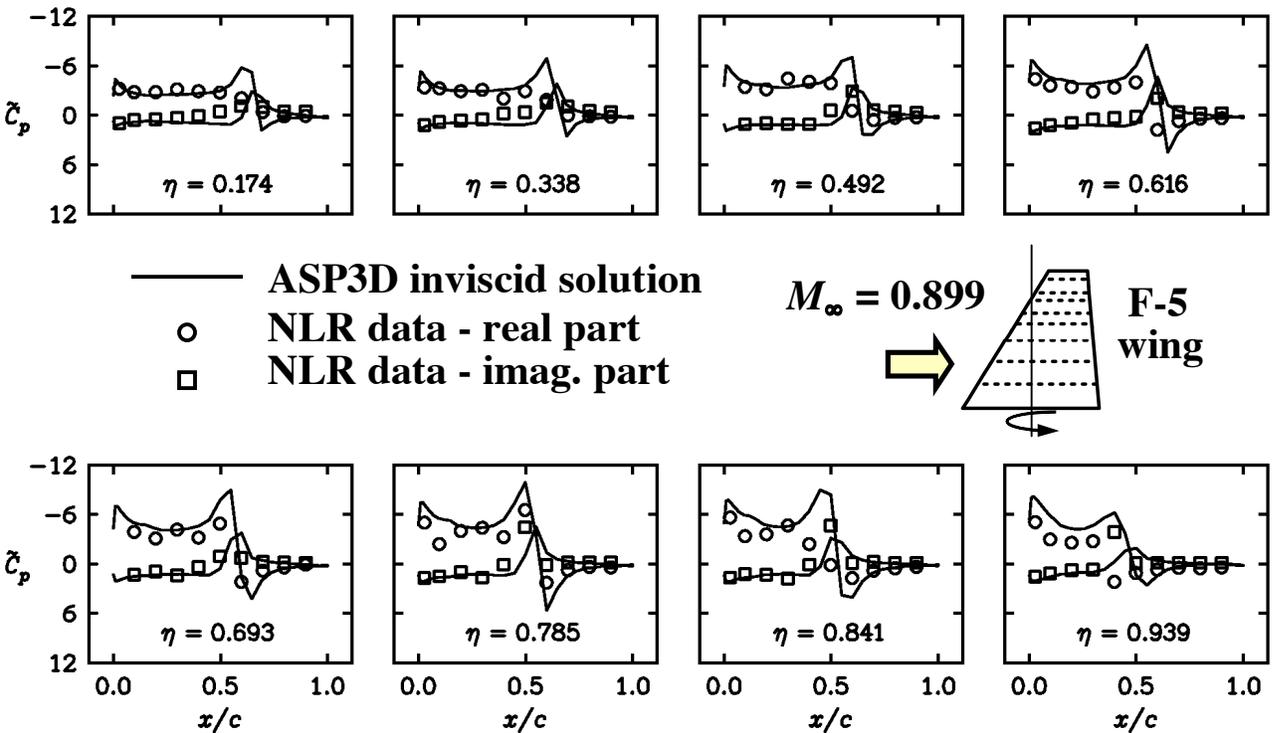


(a) Lift and moment responses.

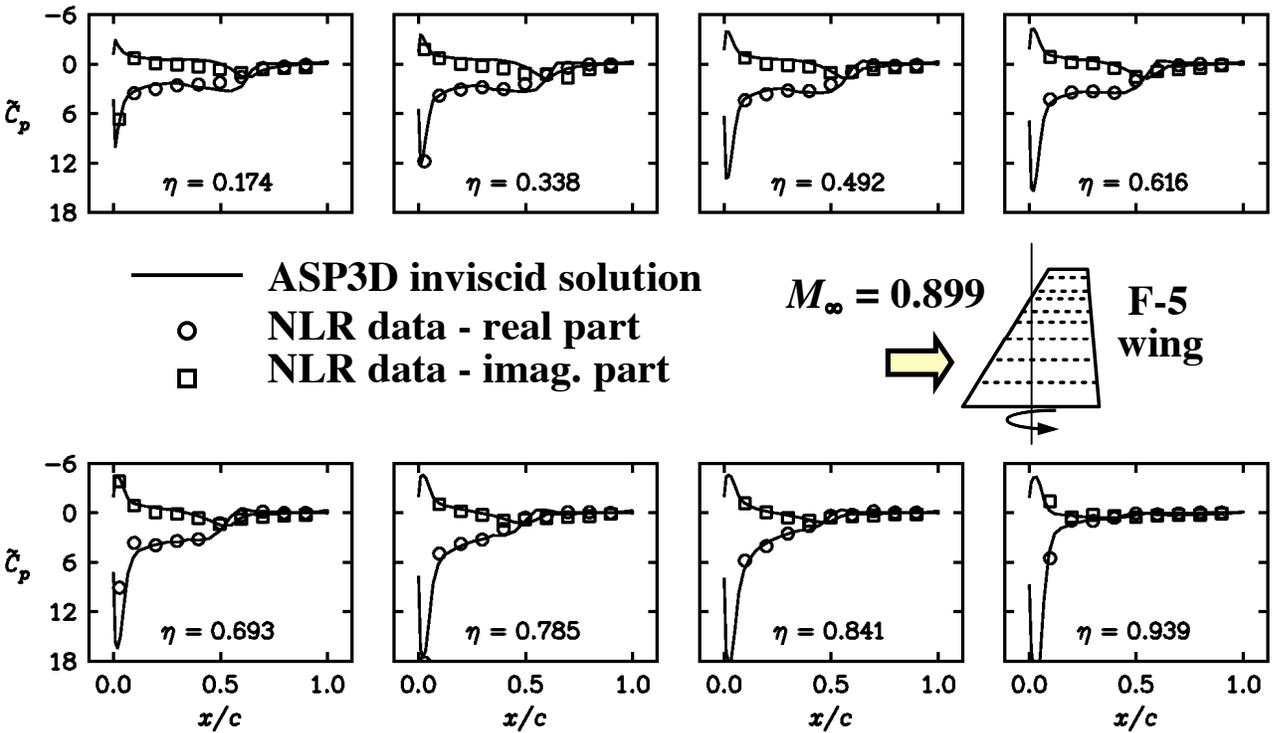


(b) Residual and number of supersonic points.

Figure 33 - ASP3D inviscid unsteady calculations for the F-5 fighter wing at $M_\infty = 0.899$, $\alpha_0 = -0.005^\circ$, $\alpha_1 = 0.109^\circ$, and $k = 0.237$, computed using the iterative multigrid scheme with only 16 steps per cycle of pitching motion.



(a) Upper surface.



(b) Lower surface.

Figure 34 - ASP3D inviscid unsteady pressure coefficient comparisons with NLR experimental data⁴⁵ for the F-5 fighter wing pitching at $M_\infty = 0.899$, $\alpha_0 = -0.005^\circ$, $\alpha_1 = 0.109^\circ$, and $k = 0.237$.

Boundary conditions:

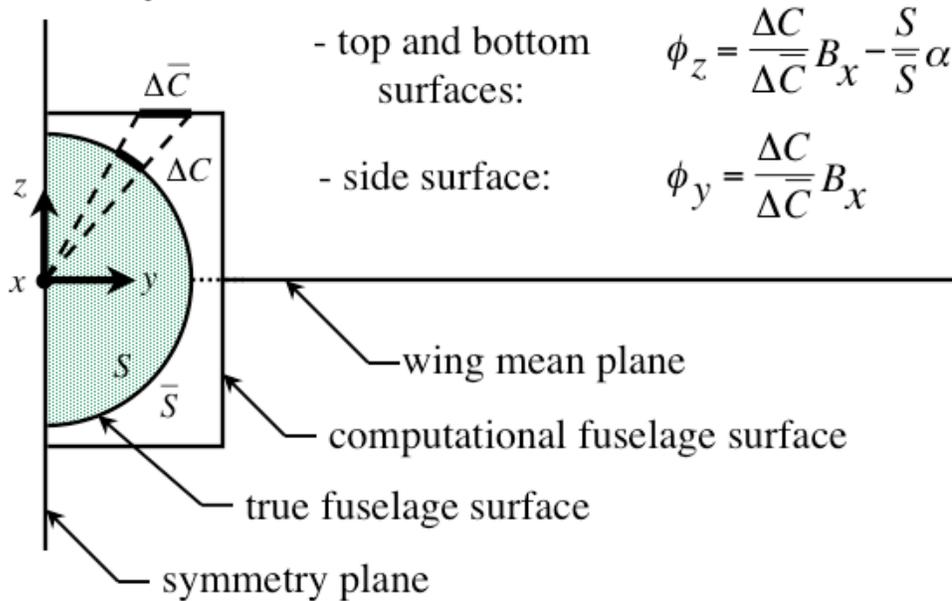


Figure 35 – Overview of ASP3D fuselage modeling including boundary conditions imposed along the top/bottom and side computational fuselage surfaces.

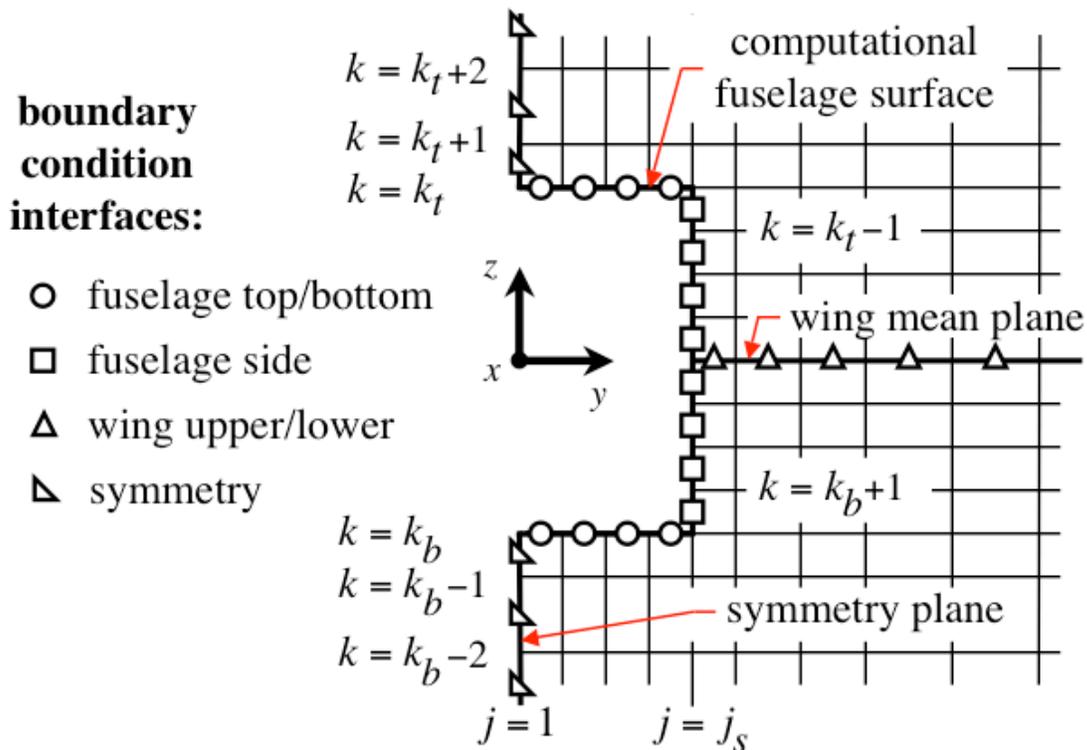


Figure 36 – ASP3D treatment of the y-z sectional finite volume mesh including cell interfaces where various boundary conditions are imposed.

Wing definition:

- sweep = 46.76 degrees
- aspect ratio = 4.0
- taper ratio = 0.6
- NACA 65A006 airfoil

Fuselage definition:

- axisymmetric body
- fineness ratio = 12
- sting mounted

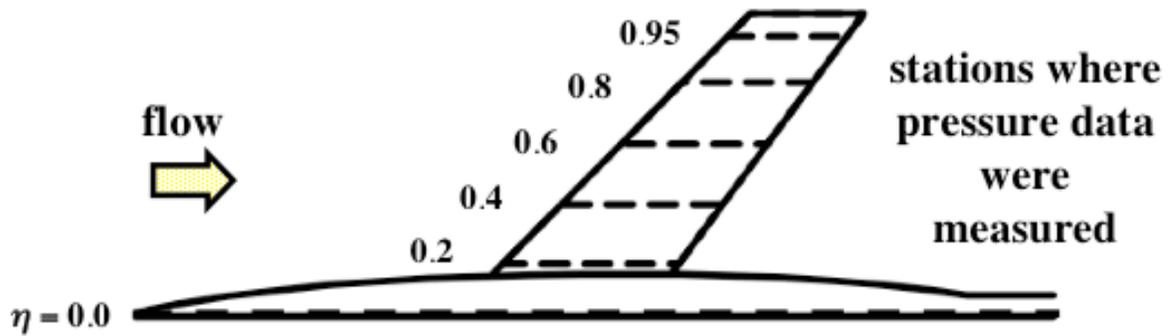


Figure 37 – Geometrical definition of the NACA RM L51F07 wing-fuselage configuration.⁴⁶

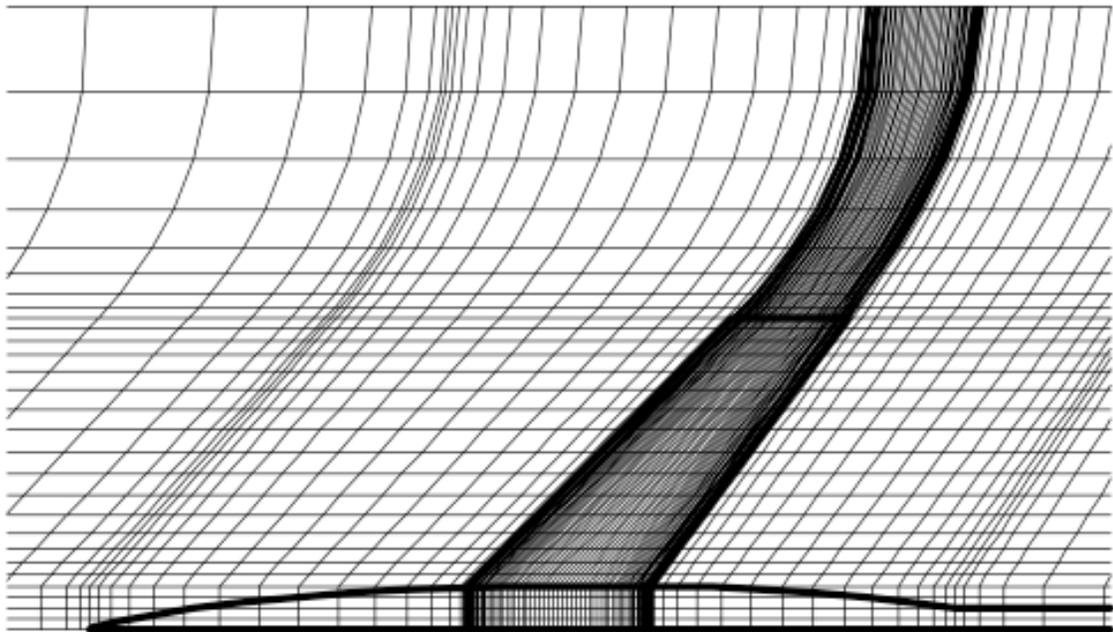


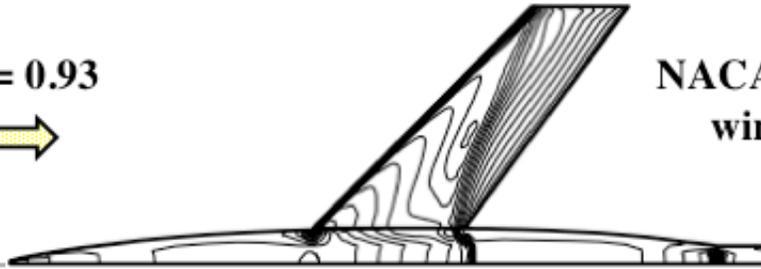
Figure 38 – Near field view of 129 x 29 planform mesh of the 129 x 29 x 75 total finite volume mesh for the NACA RM L51F07 wing-fuselage configuration.

upper surface:

$M_\infty = 0.93$

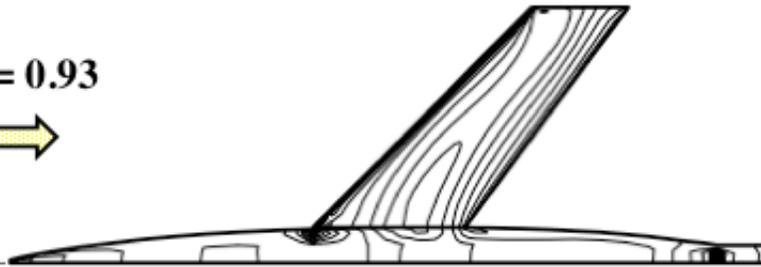


NACA RM L51F07
wing-fuselage
at $\alpha = 2^\circ$



lower surface:

$M_\infty = 0.93$



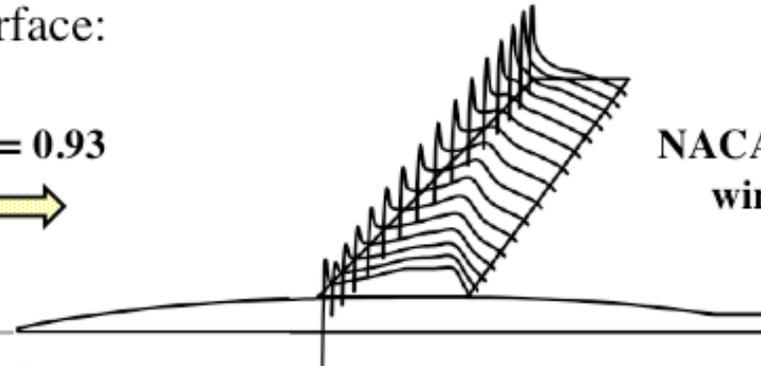
(a) Pressure coefficient contour lines on the wing and fuselage.

upper surface:

$M_\infty = 0.93$

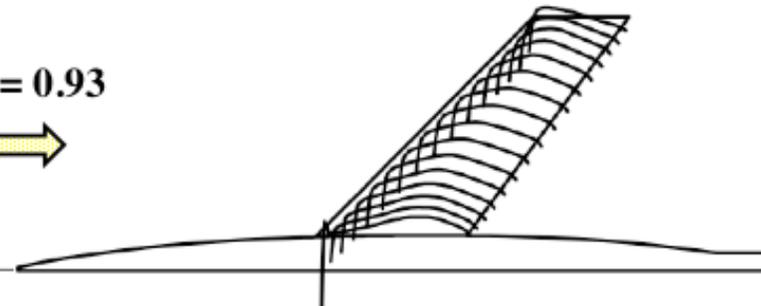


NACA RM L51F07
wing-fuselage
at $\alpha = 2^\circ$



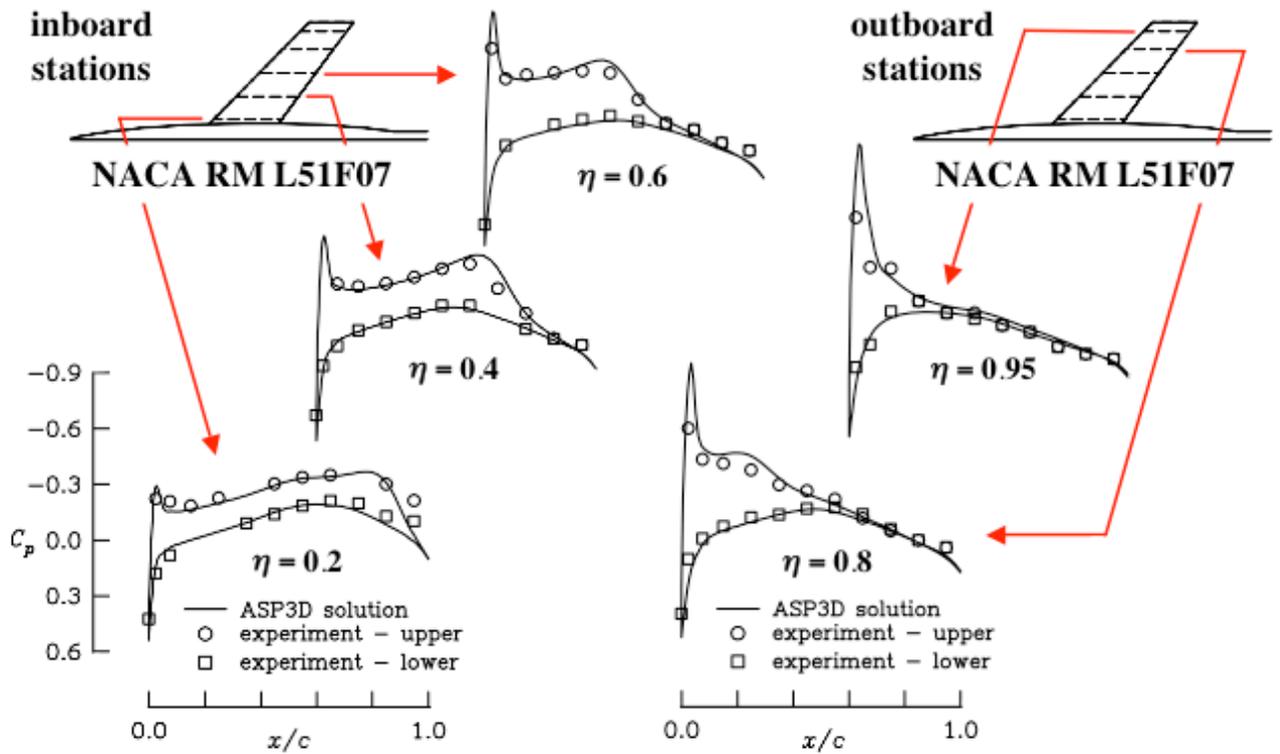
lower surface:

$M_\infty = 0.93$

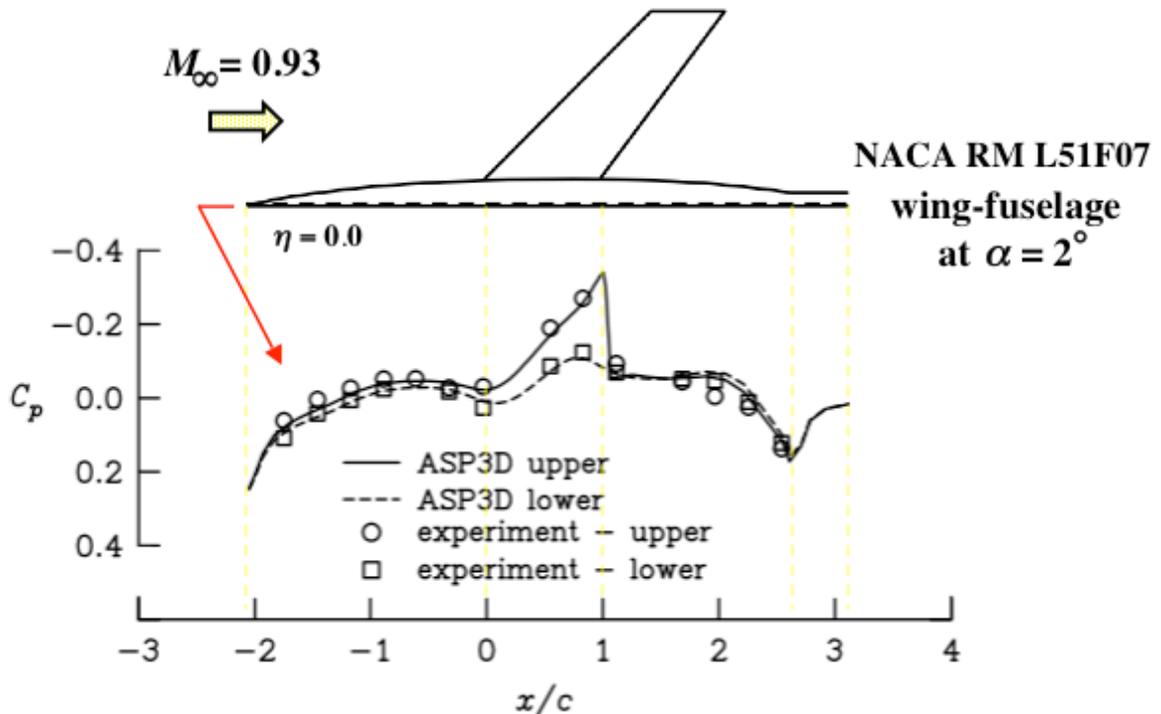


(b) Pressure coefficient distributions on the wing.

Figure 39 - ASP3D viscous calculations for the NACA RM L51F07 wing-fuselage configuration at $M_\infty = 0.93$, $\alpha = 2.0^\circ$, and $Re = 10^7$.



(a) Comparisons for the wing.



(b) Comparisons for the fuselage.

Figure 40 – ASP3D pressure coefficient comparisons with experimental data for the NACA RM L51F07 wing-fuselage configuration at $M_\infty = 0.93$, $\alpha = 2.0^\circ$, and $Re = 10^7$.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
01- 12 - 2005		Technical Memorandum			
4. TITLE AND SUBTITLE Introduction of the ASP3D Computer Program for Unsteady Aerodynamic and Aeroelastic Analyses			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Batina, John T.			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER 23-090-50-70		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199			8. PERFORMING ORGANIZATION REPORT NUMBER L-19159		
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-2005-213909		
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 02 Availability: NASA CASI (301) 621-0390					
13. SUPPLEMENTARY NOTES An electronic version can be found at http://ntrs.nasa.gov					
14. ABSTRACT A new computer program has been developed called ASP3D (Advanced Small Perturbation – 3D), which solves the small perturbation potential flow equation in an advanced form including mass-consistent surface and trailing wake boundary conditions, and entropy, vorticity, and viscous effects. The purpose of the program is for unsteady aerodynamic and aeroelastic analyses, especially in the nonlinear transonic flight regime. The program exploits the simplicity of stationary Cartesian meshes with the movement or deformation of the configuration under consideration incorporated into the solution algorithm through a planar surface boundary condition. The ASP3D code is a significant improvement to the state-of-the-art for transonic aeroelastic analyses over the CAP-TSD code (Computational Aeroelasticity Program – Transonic Small Disturbance), which was developed principally by the author in the mid-1980s. The author is in a unique position as the developer of both computer programs to compare, contrast, and ultimately make conclusions regarding the underlying formulations and utility of each code. The paper describes the salient features of the ASP3D code including the rationale for improvements in comparison with CAP-TSD.					
15. SUBJECT TERMS Aeroelasticity; Small perturbation flow; Boundary conditions; Unsteady aerodynamics; Computer programs; ASP3D; CAP-TSD					
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)
U	U	U	UU	74	19b. TELEPHONE NUMBER (Include area code) (301) 621-0390