

A Meshless Finite Difference Scheme for Compressible Potential Flows

Alejandro Ramos*
and Robert A. McDonald†

California Polytechnic State University, San Luis Obispo, CA, 93407

A meshless solution algorithm for the full potential equation has been developed by applying the principles of the Taylor Least Squares (TLS) method. This method allows for a PDE to be discretized on a local cloud of scattered nodes without the need of connectivity data. The process for discretizing the full potential equation within a meshless framework is outlined along with a novel Hermite TLS technique for enforcement of Neumann boundary conditions. Several two-dimensional test cases were solved that compare well with analytical and benchmark solutions. The first test case solved for the subcritical compressible flow over a circular cylinder at a freestream Mach number of 0.375. The last two cases solved for the non-lifting and lifting subcritical flows over a NACA 0012 airfoil with freestream conditions ($M_\infty = 0.72, \alpha = 0^\circ$) and ($M_\infty = 0.63, \alpha = 2^\circ$) respectively.

Nomenclature

<i>CFD</i>	Computational Fluid Dynamics
C_p	Pressure coefficient
<i>FPE</i>	Full Potential Equation
h	Characteristic length
M	Mach number
n	Iteration
ns	Total number of cloud support points
\mathbf{n}	Normal vector
<i>PDE</i>	Partial Differential Equation
r	Euclidian distance
S	Surface
<i>TLS</i>	Taylor Series Least Squares
\mathbf{V}	Total velocity vector
(U_∞, V_∞)	Freestream velocity components
V_n	Normal velocity component
w	Derivative coefficient
x, y	Cartesian directions
α	Angle of attack
Γ	Circulation
γ	Ratio of specific heats
ϵ	Error
θ	Angle measured from wake cut plane
μ	Doublet Strength
Φ	Total velocity potential
ϕ	Perturbation velocity potential
ρ	Density
Ψ	Vector potential

*Graduate Research Assistant, Aerospace Engineering, One Grand Avenue, AIAA Student Member.

†Associate Professor, Aerospace Engineering, One Grand Avenue, AIAA Senior Member.

Subscript

a	Airfoil
i	Indexing value
l	Lower
max	Maximum
ob	Outer Boundary
u	Upper
w	Wake
0	Cloud center
∞	Freestream

Superscript

$1x, 1y$	First derivatives in the x and y directions
$2x, 2y$	Second derivatives in the x and y directions

I. Introduction

Meshless methods can be best described as techniques that do not employ grid data to discretize a partial differential equation. There are a multitude of different algorithms within this research area that are formulated to solve a vast array of different engineering problems. A subset of solution techniques within this broad research area are those that apply the concept of discretizing a PDE on a local cloud of points. The cloud shown in Figure 1 can either be selected by having it encompass a constant local area or contain a pre-determined amount of points. In either case, this concept is not a relatively new one,¹ yet its implementation to solve aerodynamic problems has only been an active research topic in the last two decades.²

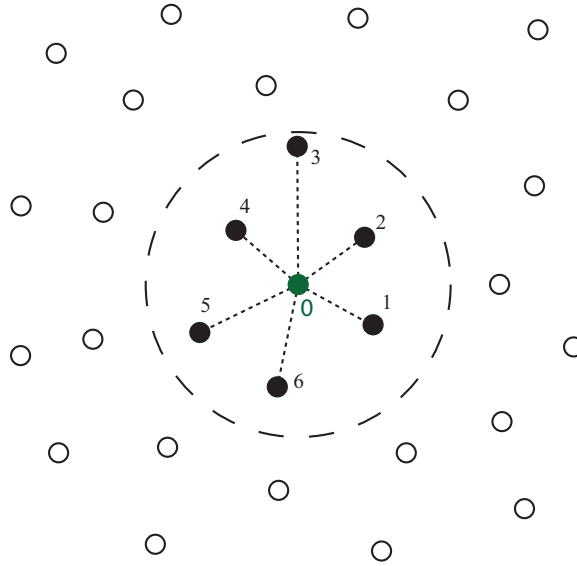


Figure 1. Point Cloud

One of the earliest methods that proposed the discretization of a PDE on a local cloud of points applied the least squares fit of a Taylor series.³⁻⁵ The earliest meshless method to solve aerodynamic problems by applying the local cloud concept was the work of Batina^{2,6} which solved the Euler/Navier Stokes equations. Batina discretized the PDE by solving a least squares polynomial fit on the local cloud of points and enforced Neumann boundary conditions by generating ghost nodes. The polynomial least square method was further developed by Oñate et. al. to solve other fluid mechanic and heat transfer problems of engineering interest.⁷⁻¹¹ Sridar and Balakrishnan revisited the Taylor series concept by developing an upwind meshless Euler solver

for two dimensional inviscid transonic flows.^{12,13}

These meshless discretization techniques have been used to model aerodynamic problems by solving some form of either the Euler or Navier-Stokes equations. This paper applies these techniques to solve the full potential equation which has the benefit of being a one equation one unknown model that can describe inviscid compressible flows. Specifically, the method developed in this paper is formulated to solve for a steady two-dimensional subcritical compressible flowfield.

Before a more detailed discussion on how to implement the meshless technique can take place, the assumptions necessary for a flow to be governed by the full potential equation will be addressed. Once the mathematical background has been established, the implementation of the TLS method to discretize the full potential equation will be discussed in detail. The enforcement of the Neumann boundary condition at walls will be specifically addressed in order to demonstrate how a Hermite interpolation technique can implicitly enforce the Neumann boundary condition. Finally some test cases will be analyzed in order to verify the accuracy of the meshless scheme.

II. The Full Potential Equation

The full potential equation (FPE) is a scalar partial differential equation which represents steady, irrotational flow. Although these limitations preclude using the FPE to model flows with physical phenomena including shocks, work addition, and friction, its single equation nature makes it an attractive approach for modeling inviscid subcritical compressible flows. The FPE is fundamentally an expression of the conservation of mass expressed here as the continuity equation for a steady flow.

$$\nabla \cdot (\rho \mathbf{V}) = 0 \quad (1)$$

Equation (1) represents the continuity equation in conservative form. Applying the chain rule to the left hand side results in the non-conservative form of the continuity equation.

$$\mathbf{V} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{V} = 0 \quad (2)$$

Although the non-conservative form can lead to numerical difficulties when modeling discontinuous problems, either form may be used to accurately solve continuous problems.¹⁴ This research is limited to subcritical and therefore shock-free flow such that the non-conservative form of the continuity equation may be used.

Helmholtz's decomposition theorem, sometimes called the fundamental theorem of vector calculus, states that any sufficiently smooth vector field whose curl and divergence vanish at infinity may be represented as the sum of two vector fields $\mathbf{V} = \nabla \Phi + \nabla \times \boldsymbol{\Psi}$.^{15,16}

The first component vector field, $\nabla \Phi$, is inherently irrotational (its curl vanishes) and the second component, $\nabla \times \boldsymbol{\Psi}$, is inherently solenoidal (its divergence vanishes). When decomposed in this way, Φ is called a scalar potential and $\boldsymbol{\Psi}$ is called a vector potential; the vector potential $\boldsymbol{\Psi}$ is also solenoidal.

When the vector field is known to be irrotational, the second term may be omitted, leaving an expression capable of representing any irrotational vector field $\mathbf{V} = \nabla \Phi$. Representing the velocity field in this way is the transformation which allows the continuity equation to be written as a scalar equation (in conservative form).

$$\nabla \cdot (\rho \nabla \Phi) = 0 \quad (3)$$

Or, equivalently in non-conservative form.

$$\rho \nabla^2 \Phi + \nabla \rho \cdot \nabla \Phi = 0 \quad (4)$$

To arrive at this equation, the velocity has been assumed to be continuous, smooth, and irrotational. These assumptions are consistent with an inviscid flow without shocks. Crocco's theorem¹⁷ dictates that such a flow will also be isentropic. The isentropic flow relations provide a scalar algebraic equation for density in terms of velocity (or velocity potential).¹⁸

$$\frac{\rho}{\rho_\infty} = \left[1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - |\nabla \Phi|^2) \right]^{\frac{1}{\gamma - 1}} \quad (5)$$

Equations 4 and 5 describe a compressible flow in terms of a single scalar differential equation. In subsequent sections the differential form of this equation will be solved for a two dimensional flowfield.

$$\rho \left(\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} \right) + \frac{\partial \rho}{\partial x} \frac{\partial \Phi}{\partial x} + \frac{\partial \rho}{\partial y} \frac{\partial \Phi}{\partial y} = 0 \quad (6)$$

Now that the governing PDE for full potential flows has been formulated, it is necessary to enforce the correct boundary conditions in order to establish a well posed problem. The velocity potential (Φ) is commonly broken up into a freestream (Φ_∞) and a perturbation (ϕ) component where the freestream potential describes the flow at a boundary infinitely far away from the perturbing surface and is assumed to be uniform and at a constant angle of attack

$$\Phi = \Phi_\infty + \phi \quad (7)$$

The problem domain in an external aerodynamics problem consists of a farfield boundary (S_∞), a wall boundary (S_a), and a wake cut plane (S_w) as shown in Figure 2. The frame of reference used here will assume that the airfoil is stationary and aligned with the x-axis and the freestream flow approaches the airfoil at an angle attack α with respect to the x-axis.

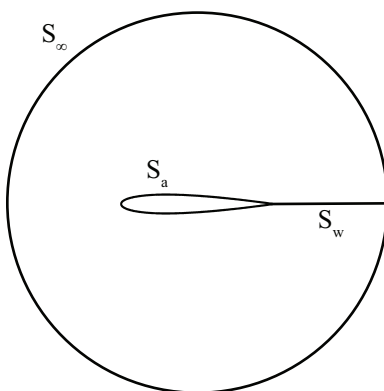


Figure 2. Problem Domain

The airfoil surface is a solid wall with the condition that the surface is also a streamline. As a consequence, this places a Neumann boundary condition on the velocity potential that states that no flow is normal to the surface.

$$\nabla \Phi \cdot \mathbf{n} = 0 \quad (8)$$

$$\frac{\partial \Phi}{\partial \mathbf{n}} = 0 \quad (9)$$

The circulation along with well posed boundary conditions need to be specified in order for a unique solution to be found for lifting flows; therefore, a wake surface is introduced in order to correctly model the circulation.¹⁹ The Kutta condition states that the value of the circulation is chosen so that the flow leaves the trailing edge of the airfoil smoothly. This means that for an airfoil with a finite trailing edge angle, such as the NACA 0012 analyzed in this paper, the trailing edge is also a stagnation point. The wake can be modeled by a constant strength doublet distribution emanating from the trailing edge or equivalently a point vortex located at the trailing edge. The strength of the doublet distribution or equivalently the strength of the point vortex is calculated by applying the Kutta condition. It is commonly enforced by calculating the jump in potential between a point just above the trailing edge and a point just below the trailing edge of the airfoil.²⁰

$$\mu_w = \Gamma = \Phi_u - \Phi_l \quad (10)$$

The velocity potential at the farfield boundary (Φ_{ob}) is composed of a superposition of a uniform velocity potential and the velocity potential induced by the point vortex that is enforcing the Kutta condition.²¹

$$\Phi_{ob} = U_{\infty}x + V_{\infty}y + \frac{\Gamma}{2\pi}\theta \quad (11)$$

where Γ is the circulation strength and θ is an angle measured anti-clockwise from the wake cut plane.

If the farfield boundary is sufficiently far away from the airfoil, the velocity can be considered the freestream value because the perturbation velocities decay as $r \rightarrow \infty$ where r is the Euclidean distance from a point on the surface of the airfoil to a point on the farfield boundary.

It has been demonstrated that a steady subcritical compressible flow can be modeled by a single scalar equation known as the full potential equation. By modeling an external aerodynamic problem with a wall, farfield boundary, and wake cut plane, a well posed problem is constructed that is capable of describing the scalar potential and density field along with the vector velocity field. The next section will describe the meshless method that has been formulated to solve this problem.

III. Solution Algorithm

The solution algorithm operates in an analogous manner to a standard finite difference solver. Figure 3 outlines the six main steps that the solver performs. This section will discuss how each of these steps are evaluated to solve the full potential equation.

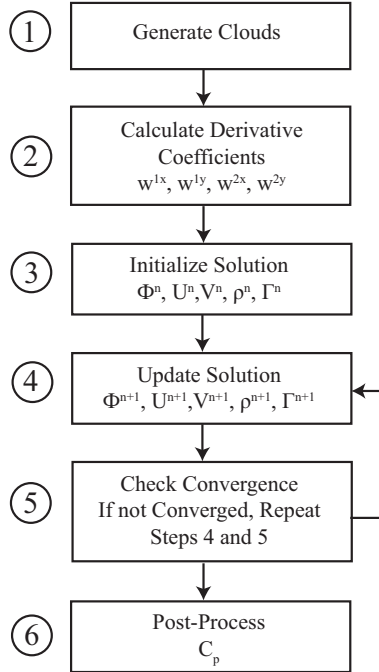


Figure 3. Iterative Procedure

III.A. Cloud Generation

The clouds for all the test cases were constructed from nodes generated by an unstructured mesh generator. While this point generation technique seems to eliminate the claim that this numerical technique is meshless, the main focus in this paper is the solution algorithm which does not depend on how the nodes are initially generated. For this algorithm to be a useful design tool, a point generation technique that does not require meshing of the problem geometry would be essential. Efforts have been made in this area by Löhner and Oñate^{22, 23} who developed an advancing front point generation technique which shows some promise for the eventual development of a fully integrated meshless point generator and solution algorithm.

As stated earlier, different criteria can be used to determine which points are selected to be a part of the local cloud. The current implementation applies a nearest neighbor search to determine the cloud support

points. Each cloud will contain a pre-determined amount of nearest neighbors instead of selecting clouds of constant area. Figure 4 depicts the typical point distributions for a circular cylinder and a NACA 0012 airfoil. There is a higher density of points near areas where there are expected to be high gradients such as the leading and trailing edge of the airfoil. The nearest neighbor search criterion will ensure that the point clouds in regions of higher point density will encompass a smaller area than clouds near the farfield boundary. For the simple two dimensional test cases, this method is quick and simple to implement. For problems with a large amount of surface and domain points such as complex three-dimensional configurations, an oct-tree data structure could be implemented to speed up the nearest neighbor search.

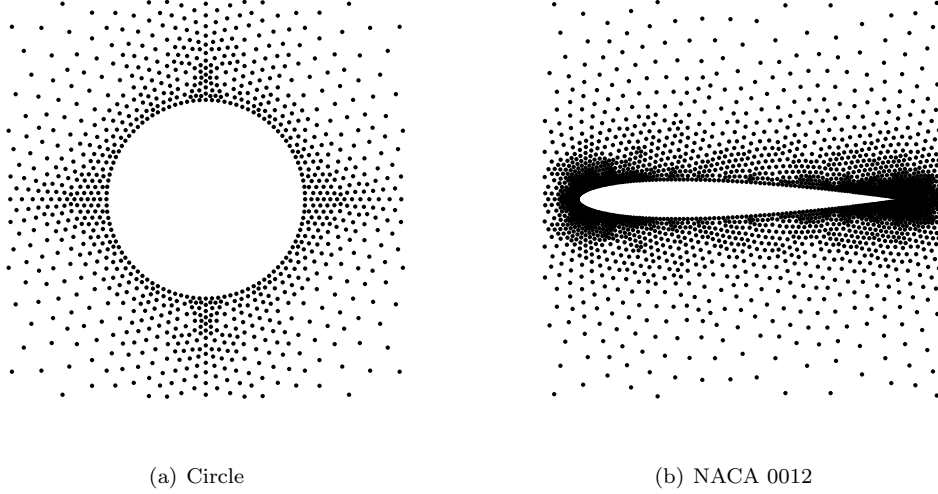


Figure 4. Close-Up View of Point Distributions

III.B. Derivative Coefficient Calculation

Any order derivative of Φ can be discretized into an algebraic sum of its nodal values in the local cloud.²⁴

$$\frac{\partial^m \Phi_0}{\partial d^m} \cong \sum_{i=1}^{ns} w_i^{md} \Delta \Phi_i \quad (12)$$

$$\Delta \Phi_i = \Phi_i - \Phi_0 \quad (13)$$

where m is the order of the derivative, d is the Cartesian direction, and ns is the number of support points in the cloud. For example, the estimate for the second y derivative at node 0 in the cloud shown in Figure 1 ($m = 2$, $d = y$, $ns = 6$) would be

$$\frac{\partial^2 \Phi_0}{\partial y^2} \cong \sum_{i=1}^6 w_i^{2y} \Delta \Phi_i \quad (14)$$

The truncated Taylor series between the cloud center $\Phi_0 = \Phi(x_0, y_0)$ and the cloud support points $\Phi_i = \Phi(x_i, y_i)$ can be expressed in compact form as:

$$\Phi_i = \Phi_0 + \sum_{q=1}^r \sum_{m=0}^q \left(\frac{\partial^q \Phi}{\partial x^{q-m} \partial y^m} \frac{\Delta x^{q-m} \Delta y^m}{(q-m)! m!} \right) \quad (15)$$

where r is the highest order term retained in the Taylor series. The r derivatives of Φ can be determined by applying a least squares minimization of Equation (15) for the ns support nodes in the cloud. It can be shown that the solution to the least squares problem results in an expression that relates the derivatives of Φ only in terms of the node spacing in the cloud.²⁵

$$\{d\} = \mathbf{C}\{\Phi\} \quad (16)$$

where $\{d\}$ is an $r \times 1$ vector containing the derivatives of Φ_0 at the cloud center, \mathbf{C} is an $r \times n_f$ matrix that results from the least squares minimization problem, and Φ is an $n_f \times 1$ vector containing $\Phi_i - \Phi_0$ at each of the n_f cloud support points. The sequencing of the derivatives in Equation (15) was chosen so that the first five entries in the $\{d\}$ vector are

$$\left[\frac{\partial \Phi_0}{\partial x} \quad \frac{\partial \Phi_0}{\partial y} \quad \frac{\partial^2 \Phi_0}{\partial x^2} \quad \frac{\partial^2 \Phi_0}{\partial x \partial y} \quad \frac{\partial^2 \Phi_0}{\partial y^2} \right]^T \quad (17)$$

With this sequence chosen, the derivative coefficients in Equation (12) can now be expressed as rows of the \mathbf{C} matrix. As an example the derivative coefficients for the first y-derivative of Φ are

$$w_i^{1y} = C_{2,i} \quad (18)$$

III.C. Solution Initialization

For all of the test cases the solution will be initialized with the freestream conditions at every point on the boundary and in the domain. Therefore the potential will be Φ_∞ , the velocity components are $U = U_\infty$ and $V = U_\infty$, and the density ratio is $\rho = 1$ at every point.

III.D. Solution Update

The continuity equation and boundary conditions need to be discretized in order for the PDE to be expressed in an algebraic form which can then be solved for the solution update at the cloud center. A Dirichlet boundary condition will be enforced at the farfield boundary which will be modeled as a circle centered at the middle of the airfoil with a radius that is sufficiently far away. The total velocity potential at each point on the outer boundary will be expressed as:²¹

$$\Phi_{ob} = \Phi_\infty + \frac{\Gamma}{2\pi} \theta \quad (19)$$

where Γ is the circulation strength and θ is an angle measured from a line extending horizontally from the trailing edge of the airfoil. After all the points have undergone one iteration, the circulation value is updated by enforcing the Kutta condition which is equal to the difference in velocity potential values between the closest points on the upper and lower side of the airfoil trailing edge.

A Neumann boundary condition is enforced at the airfoil surface. Since the total velocity potential is being determined instead of the perturbation potential, this boundary condition can be simply expressed as

$$\frac{\partial \Phi}{\partial \mathbf{n}} = 0 \quad (20)$$

Instead of focusing on the specifics surrounding boundary condition enforcement, it will be assumed for the time being that there is an implicit way to ensure that both the farfield and wall boundary conditions are being correctly enforced and all of the flow variables at points located on these boundaries can be determined by solving the discretized form of the continuity equation. Therefore the remainder of this section will specifically focus on how the continuity equation is discretized.

Since all the test cases involve subcritical compressible flows, the full potential equation in non-conservative form will be solved.

$$\rho \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial \rho}{\partial x} \frac{\partial \Phi}{\partial x} + \frac{\partial \rho}{\partial y} \frac{\partial \Phi}{\partial y} = 0 \quad (21)$$

This equation can be discretized using the derivative coefficients determined from subsection III.B.

$$\rho_0^n \left[\sum_{i=1}^{ns} w_i^{2x} (\Phi_i^n - \Phi_0^n) + \sum_{i=1}^{ns} w_i^{2y} (\Phi_i^n - \Phi_0^n) \right] + \sum_{i=1}^{ns} w_i^{1x} (\rho_i^n - \rho_0^n) \sum_{i=1}^{ns} w_i^{1x} (\Phi_i^n - \Phi_0^n) + \sum_{i=1}^{ns} w_i^{1y} (\rho_i^n - \rho_0^n) \sum_{i=1}^{ns} w_i^{1y} (\Phi_i^n - \Phi_0^n) = 0 \quad (22)$$

The goal is to re-arrange the terms in this algebraic equation to determine the updated $(n + 1)$ value of the velocity potential. In order to put Equation (22) in a more compact form let:

$$\rho_x = \sum_{i=1}^{ns} w_i^{1x} (\rho_i^n - \rho_0^n) \quad (23)$$

$$\rho_y = \sum_{i=1}^{ns} w_i^{1y} (\rho_i^n - \rho_0^n) \quad (24)$$

$$\rho_0^n \sum_{i=1}^{ns} (w_i^{2x} + w_i^{2y}) (\Phi_i^n - \Phi_0^n) + \rho_x \sum_{i=1}^{ns} w_i^{1x} (\Phi_i^n - \Phi_0^n) + \rho_y \sum_{i=1}^{ns} w_i^{1y} (\Phi_i^n - \Phi_0^n) = 0 \quad (25)$$

Equation (25) can then be split up into terms associated with the cloud center and terms associated with values at the cloud support points.

$$\begin{aligned} & \sum_{i=1}^{ns} \rho_0^n (w_i^{2x} + w_i^{2y}) + \rho_x \sum_{i=1}^{ns} w_i^{1x} + \rho_y \sum_{i=1}^{ns} w_i^{1y} \Phi_0^n = \\ & \sum_{i=1}^{ns} \rho_0^n (w_i^{2x} + w_i^{2y}) + \rho_x \sum_{i=1}^{ns} w_i^{1x} + \rho_y \sum_{i=1}^{ns} w_i^{1y} \Phi_i^n \end{aligned} \quad (26)$$

The updated value (Φ_0^{n+1}) can now be determined by solving Equation (26) for Φ_0 .

$$\Phi_0^{n+1} = \frac{\sum_{i=1}^{ns} \rho_0^n (w_i^{2x} + w_i^{2y}) + \rho_x \sum_{i=1}^{ns} w_i^{1x} + \rho_y \sum_{i=1}^{ns} w_i^{1y} \Phi_i^n}{[\rho_0^n (w_i^{2x} + w_i^{2y}) + \rho_x \sum_{i=1}^{ns} w_i^{1x} + \rho_y \sum_{i=1}^{ns} w_i^{1y}]} \quad (27)$$

Now that the value of the velocity potential has been updated, the velocity components and density can also be updated. The x and y components of the velocity are determined by numerically calculating the gradient of the velocity potential.

$$U_0^{n+1} = \sum_{i=1}^{ns} w_i^{1x} (\Phi_i^n - \Phi_0^{n+1}) \quad (28)$$

$$V_0^{n+1} = \sum_{i=1}^{ns} w_i^{1y} (\Phi_i^n - \Phi_0^{n+1}) \quad (29)$$

Once the velocity values are determined, the density ratio at the cloud center can be updated.

$$\rho_0^{n+1} = 1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - |\nabla \Phi_0^{n+1}|^2)^{\frac{1}{\gamma-1}} \quad (30)$$

III.E. Convergence Estimation

The convergence criteria for this algorithm will be defined as the maximum change of the velocity potential between the current and the previous iteration step for all the points on the boundaries and in the domain.

$$\epsilon = \max(|\{\Phi\}^{n+1} - \{\Phi\}^n|) \quad (31)$$

IV. Neumann Boundary Condition Enforcement

In order to satisfy the Neumann boundary condition on the surface, an extension of the Taylor series least squares problem addressed in Subsection III.B was implemented. This technique, given the name Constrained Hermite TLS (CHTLS), allows for the no normal flow boundary condition to be implicitly satisfied by adding information about the directional derivative of the velocity potential at boundary points.

When a cloud contains nodes that belong to the surface, additional terms are added to the least squares minimization problem. The specific details on the formulation of the CHTLS method can be found in the companion paper written by the authors.²⁵ Constraining the TLS problem to ensure that the Neumann boundary condition is enforced results in a modified form of Equation (16).

$$\{d\} = \mathbf{F}\{\Phi\} + \mathbf{G}\{h\} + \{H\}h_0 \quad (32)$$

The vector $\{h\}$ contains the directional derivative observations at the points where they are enforced in a least squares sense and h_0 is the directional derivative constraint at the point of interest. For potential flows the directional derivative is $\frac{\partial\Phi}{\partial n}$, which for a solid wall is zero, allowing \mathbf{G} and $\{H\}$ to be dropped. As with the normal TLS method the derivative coefficients are entries within the \mathbf{F} matrix. As an example, the derivative coefficients for the first y-derivative of Φ are

$$w_i^{1y} = F_{2,i} \quad (33)$$

V. Results

The TLS method was applied to solve a variety of non-lifting and lifting subcritical compressible cases. The first two cases are non-lifting flows over a circular cylinder and a NACA 0012 airfoil. In order to examine how well the method can solve lifting flows, the flow over a NACA 0012 at a two degree angle of attack was also modeled.

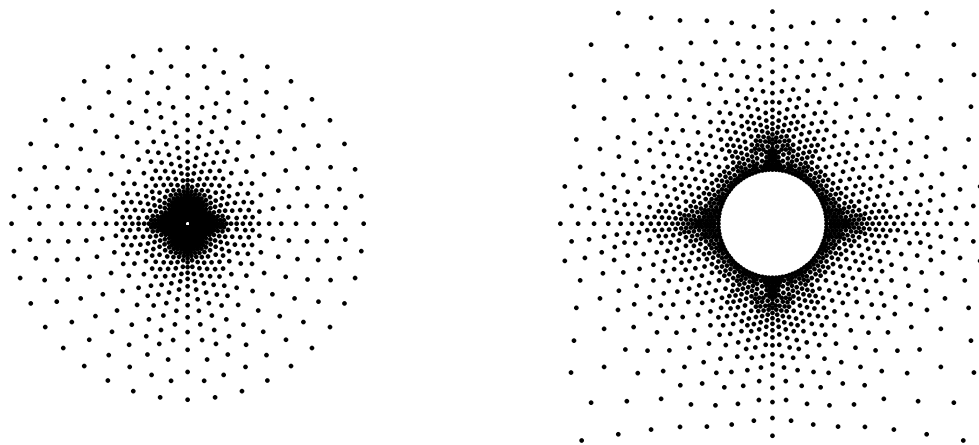
Section III.B showed that a truncated Taylor series evaluated at the cloud support points is used to determine the derivative coefficients. This Taylor series can keep an arbitrary number of terms with a higher number of terms corresponding to a higher order estimation. For example Equation (15) with $r = 2$ corresponds to the Taylor series that retains the second order terms.

$$\begin{aligned} \Delta\Phi_i &= \Delta x_i \frac{\partial\Phi_0}{\partial x} + \Delta y_i \frac{\partial\Phi_0}{\partial y} + \\ &\frac{\Delta x_i^2}{2} \frac{\partial^2\Phi_0}{\partial x^2} + \Delta x_i \Delta y_i \frac{\partial^2\Phi_0}{\partial x \partial y} + \frac{\Delta y_i^2}{2} \frac{\partial^2\Phi_0}{\partial y^2} + O(\Delta x_i^3, \Delta y_i^3) \end{aligned} \quad (34)$$

Sridar and Balakrishnan¹² proved theoretically that the order of the leading truncation error term for the estimate of the p_{th} derivative of a function is h^{r+1-p} where h is the characteristic length in the Taylor series expansion and r is the highest order of the terms kept in the Taylor expansion. The numerical experiments conducted in the companion paper²⁵ have shown that the CHTLS method retains the same order of accuracy properties when evaluating derivatives on clouds that contain boundary nodes. Therefore a set of derivative coefficients using $r = 2$ were used to estimate first derivatives in Equation (27) while another set of derivative coefficients using $r = 3$ were used to estimate second derivatives. These selections of r would ensure that all of the derivatives required would be second order accurate.

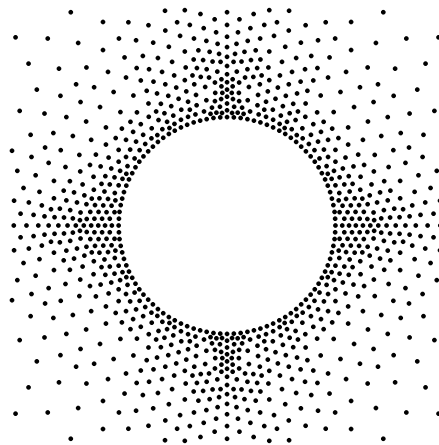
V.A. Circular Cylinder

The first case that will be examined is the sub-critical ($M_\infty = 0.375$) flow over a circular cylinder. The suction peak velocity for several node distributions was compared to the analytical solution ($V_{max} = 2.260$) calculated by van Dyke.²⁶ The node distribution, which has 100 nodes uniformly distributed on the cylinder surface, was constructed using ICEM CFD and is shown in Figure 5. The farfield boundary was chosen to be a circle with radius 20 measured from the circular cylinder center. The domain nodes were grown off the surface distribution to the farfield boundary using a 1.15 growth ratio. These metrics are summarized in Table 1.



(a) Complete View

(b) Mid View



(c) Close-Up View

Figure 5. Cylinder Node Distribution

Table 1. Baseline Distribution Metrics

Number of Boundary Nodes	100
Farfield Boundary Radius	20
Cell Growth Ratio	1.15
Total Number of Nodes	2084
Number of Support Points	18

The analytical solution for the peak velocity on a circular cylinder in compressible potential flow was developed by van Dyke using a Janzen-Rayleigh series expansion. This expansion involves a set of 29 terms and integer powers of M_∞^2 .²⁶

$$V_{max} = U_\infty \sum_{n=1}^{29} q_n (M_\infty^2)^{n-1} \quad (35)$$

Since this analytical solution only allows for a comparison at the velocity peak, a solution from the Ph. D. dissertation written by German²⁷ was also used since it provides data for the entire distribution of C_p on the cylinder surface. German developed a Riemannian geometric mapping technique to determine the compressible flowfield by solving a panel method on an equivalent incompressible shape.^{28,29} The method was extensively verified and showed that the peak velocity closely matched the analytical result determined by van Dyke. The results from the converged solution are presented in the figures below. The predicted pressure coefficient distribution, as shown in Figure 6, matches very closely with the results obtained by German²⁷ and van Dyke.²⁶

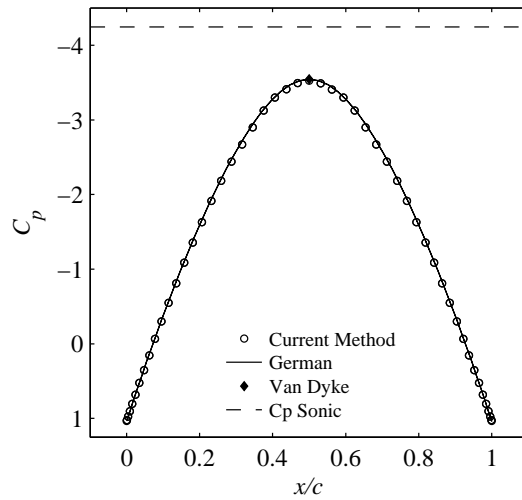


Figure 6. Upper Surface Pressure Coefficient for the Cylinder

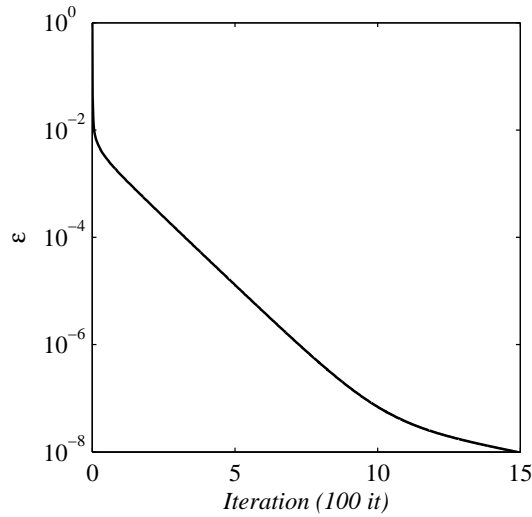


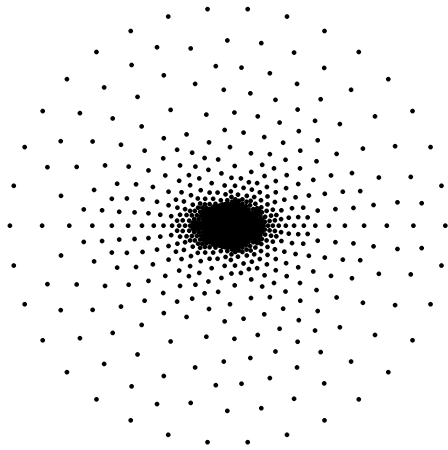
Figure 7. Cylinder Convergence History

V.B. Nonlifting NACA 0012 Airfoil

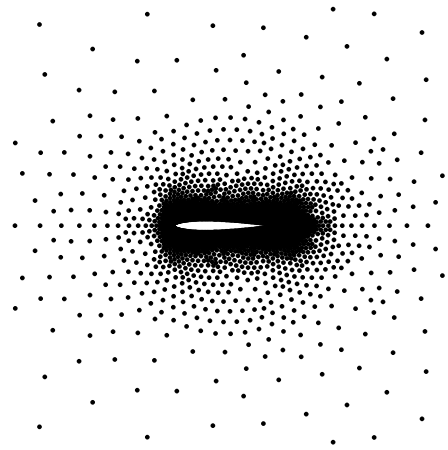
The second case that was examined was the sub-critical flow over a NACA 0012 airfoil with a freestream Mach number of 0.72 at a zero angle of attack. The grid has 320 nodes on the airfoil surface, distributed using the biexponential bunching law in ICEM CFD. The biexponential scheme produces exponentially distributed nodes across the airfoil surface given the spacing and growth ratio at the leading and trailing edge of the airfoil. The growth ratio was held at 1.01 to keep the node spacing nearly constant for the leading and trailing edges. The off body nodes were grown with a 1.15 growth ratio for 0.5 chord lengths, then a 1.2 ratio extended to the farfield boundary. The farfield boundary is a circle with a radius of 10 chord lengths measured from the airfoil center. These grid metrics produced a distribution with a total of 4228 nodes. The solution converged to an iteration tolerance of 10^{-8} in approximately 1500 iterations, similar to the cylinder problem. The predicted pressure coefficient distribution, as shown in Figure 9, matches very closely with the results obtained by Lock³⁰ and German.²⁷ The pressure coefficient distribution given by Lock can be considered a benchmark solution since the author states that the results are within one percent of the exact full potential solution.

Table 2. NACA 0012 Baseline Distribution Metrics

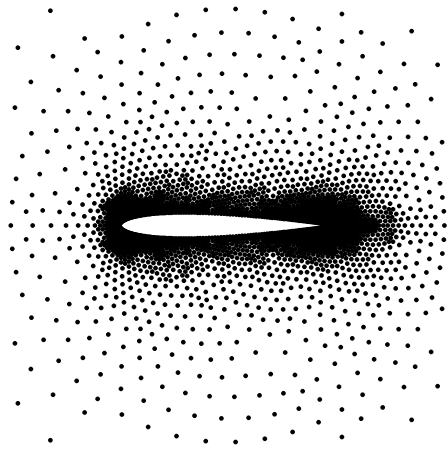
Number of Boundary Nodes	320
Farfield Boundary Radius	10
Cell Growth Ratio	1.15 - 1.2
Total Number of Nodes	4228
Number of Support Points	12



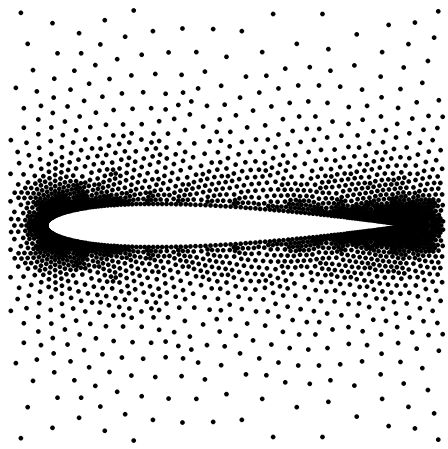
(a) Complete View



(b) Far View



(c) Mid View



(d) Close-Up View

Figure 8. NACA 0012 Node Distribution

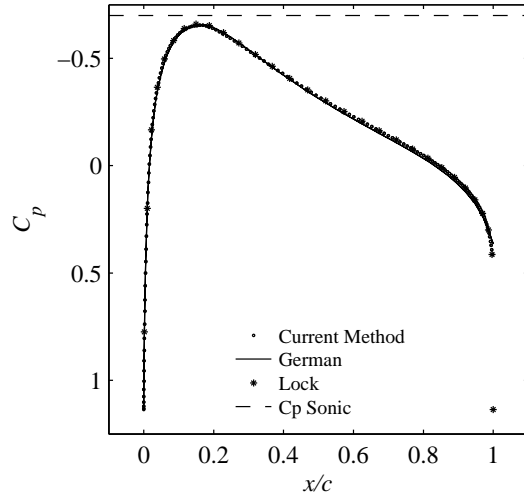


Figure 9. Pressure Coefficient Distribution for Non-Lifting Case ($M_\infty = 0.72$)

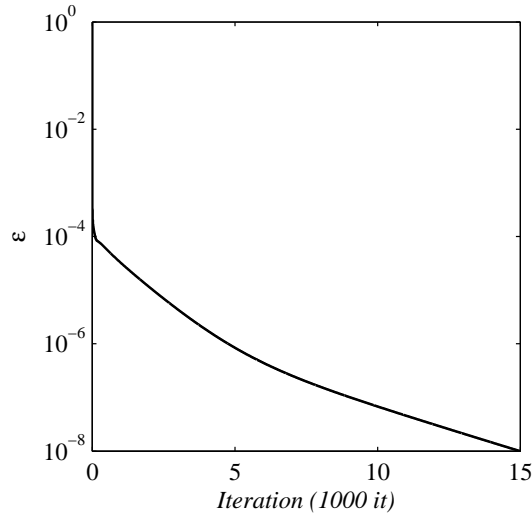


Figure 10. Non-Lifting Convergence History

V.C. Lifting NACA 0012 Airfoil

The last case that was examined was the flow over the NACA 0012 airfoil with a freestream Mach number of 0.63 at an angle of attack of 2° . These freestream conditions were chosen because the flowfield is still subcritical and the benchmark results are available in the AGARD report written by Lock.³⁰ The results are shown for the solution of the full potential equation using the same node distribution as the non-lifting case. The pressure coefficient distribution matches quite well with the results provided by Lock with the largest disparity between the two solutions located near the suction peak as shown in Figure 11. The number of iterations required for the solution to converge was an order of magnitude larger than the two previous problems. This longer convergence time is attributed to the changing value in circulation as the Kutta condition is updated. Lock reports that the local Mach number reaches a maximum value of 0.983 which implies that the meshless algorithm can accurately model cases with nearly sonic conditions.

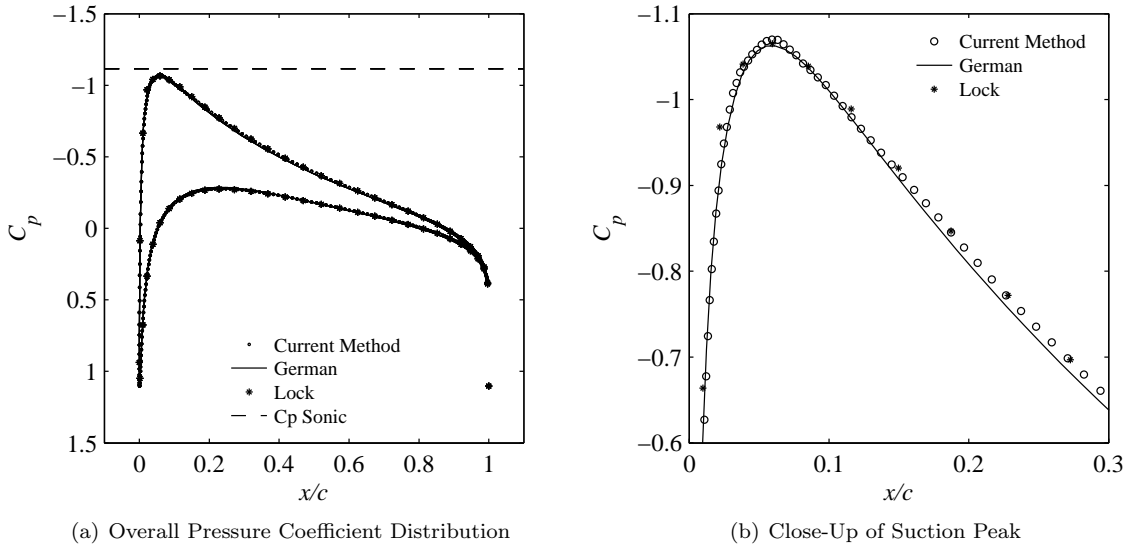


Figure 11. NACA 0012 Lifting Case Solution

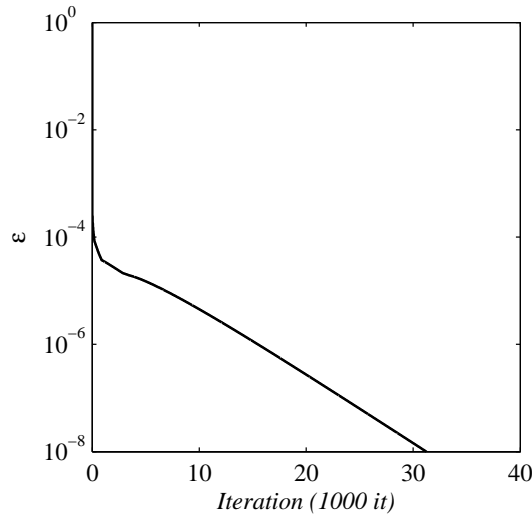


Figure 12. Lifting Case Convergence History

VI. Conclusion

The meshless method developed in this paper yielded accurate results for the subcritical compressible flowfield about two-dimensional geometries. These solutions were obtained by discretizing and solving the full potential equation which has the same fidelity as an Euler code for subcritical flows but with the added benefit of only requiring the solution of one variable. The CHTLS method yielded accurate results by implicitly satisfying the Neumann boundary condition for clouds containing nodes belonging to a wall boundary. The methods outlined in this paper have established the groundwork necessary for developing a meshless full potential aerodynamic analysis tool but there is still a variety of different topics that can still be explored such as extending the method to three dimensions and solving for transonic flowfields.

Acknowledgments

The authors would like to thank Craig Nickol, the primary technical monitor, for NASA Research Announcement Grant NNX07A014A which supported this work.

References

- ¹Forsythe, G. E. and Wasow, W. R., *Finite Difference Methods for Partial Differential Equations*, John Wiley and Sons, 1960.
- ²Batina, J. T., "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Two-Dimensional Applications," TM-107631, NASA, 1992.
- ³MacNeal, R. H., "An Asymmetrical Finite Difference Network," *Quarterly of Applied Mathematics*, Vol. 11, 1953, pp. 295–310.
- ⁴Liszka, T. and Orkisz, J., "The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics," *Computers and Structures*, Vol. 11, 1980, pp. 83–95.
- ⁵Liszka, T., "An Interpolation Method for an Irregular Net of Nodes," *International Journal for Numerical Methods in Engineering*, Vol. 20, 1984, pp. 1599–1612.
- ⁶Batina, J. T., "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex-Aircraft Applications," Tech. rep., NASA TM-107727, Langley Research Center, Hampton, VA, 1993.
- ⁷Oñate, E., Idelsohn, S., Zienkiewicz, O. C., and Taylor, R. L., "A Finite Point Method in Computational Mechanics: Applications to Convective Transport and Fluid Flow," *International Journal for Numerical Methods in Engineering*, Vol. 39, 1996, pp. 3839–3866.
- ⁸Oñate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L., and Sacco, C., "A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, 1996, pp. 315–346.
- ⁹Oñate, E. and Idelsohn, S., "A Mesh-Free Finite Point Method for Advective-Diffusive Transport and Fluid Flow Problems," *Computational Mechanics*, Vol. 21, 1998, pp. 283–292.
- ¹⁰Oñate, E., Sacco, C., and Idelsohn, S., "Meshless Analysis of Incompressible Flows Using the Finite Point Method," Technical report, International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain, 2000.
- ¹¹Lohner, R., Sacco, C., Oñate, E., and Idelsohn, S., "A Finite Point Method for Compressible Flow," *International Journal for Numerical Methods in Engineering*, Vol. 53, 2002, pp. 1765–1779.
- ¹²Sridar, D. and Balakrishnan, N., "An Upwind Finite Difference Scheme for Meshless Solvers," *Journal of Computational Physics*, Vol. 189, 2003, pp. 1–29.
- ¹³Sridar, D. and Balakrishnan, N., "Convergence Acceleration of an Upwind Least Squares Finite Difference Based Meshless Solver," *AIAA Journal*, Vol. 44, 2006, pp. 2189–2196.
- ¹⁴Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Taylor and Francis, 2nd ed., 1997.
- ¹⁵Aris, R., *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*, Dover Publications, Englewood Cliffs, New Jersey, 1962.
- ¹⁶Borisenko, A. I., Tarapov, I. E., and Silverman, R. A., *Vector Tensor Analysis with Applications*, Dover Publications, 1979.
- ¹⁷John, J. E. and Keith, T. G., *Gas Dynamics*, Pearson Prentice Hall, 3rd ed., 2006.
- ¹⁸Morino, L. and Iemma, U., "Boundary Integral Equations and Conservative Dissipation Schemes for Full Potential Transonic Flows," *Computational Mechanics*, Vol. 13, 1993, pp. 90–99.
- ¹⁹Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw Hill, New York, 4th ed., 2007.
- ²⁰Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, Cambridge University Press, Cambridge, MA, 2nd ed., 2001.
- ²¹Holst, T. L., "Transonic Flow Computations Using Nonlinear Potential Methods," *Progress in Aerospace Sciences*, Vol. 36, 2000, pp. 1–61.
- ²²Lohner, R. and Oñate, E., "An Advancing Front Point Generation Technique," *Communications in Numerical Methods in Engineering*, Vol. 14, 1998, pp. 1097–1108.
- ²³Lohner, R. and Oñate, E., "A General Advancing Front Technique for Filling Space with Arbitrary Objects," *International Journal for Numerical Methods in Engineering*, Vol. 61, 2004, pp. 1977–1991.
- ²⁴Katz, A. and Jameson, A., "Edge-Based Meshless Methods for Compressible Flow Simulations," *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.
- ²⁵McDonald, R. A. and Ramos, A., "Hermite TLS for Unstructured and Mesh-free Derivative Estimation Near and On Boundaries," *Submitted to the 2011 AIAA Aerospace Sciences Meeting*, 2011.
- ²⁶Van Dyke, M., "Long Series in Mechanics: Janzen-Rayleigh Expansion for a Circle," *Meccanica*, Vol. 33, 1998, pp. 517–522.
- ²⁷German, B. J., *A Riemannian Geometric Mapping Technique for Identifying Incompressible Equivalents to Subsonic Potential Flows*, Ph.D. thesis, Georgia Institute of Technology, 2007.
- ²⁸German, B. J., "Laplacian Equivalents to Subsonic Potential Flows," *AIAA Journal*, Vol. 47, 2009, pp. 129–141.
- ²⁹German, B. J. and Sivakumar, H., "Laplacian Equivalents to Subsonic Cascade Flows," *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, 2009.
- ³⁰Lock, R. C., "Test Cases for Numerical Methods in Two-Dimensional Transonic Flows," Tech. Rep. 575, AGARD, Neuilly-sur-Seine, France, 1970.