38th Fluid Dynamics Conference and Exhibit<BR>
23 - 26 June 2008, Seattle, Washington

AIAA 2008-3989

Edge-based Meshless Methods for Compressible Viscous Flow with Applications to Overset Grids

Aaron Katz[*] and Antony Jameson[†]

Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305, USA

This paper concerns two key developments of a meshless method: (1) meshless discretization of the Navier-Stokes equations, and (2) embedding the the meshless method within an overset grid context. First, the spatial and temporal discretization procedures for the meshless method applied to the Navier-Stokes equations are given. Next, the meshless method is shown to be well suited for overset grid applications. In the overset approach, anisotropic viscous grids are applied near the body while isotropic cartesian grids cover the remainder of the domain to the far field boundaries. In place of conventional interpolation procedures, the meshless method accomplishes the task of inter-grid communication between near-body and off-body grid components. The meshless interface appears to provide an attractive alternative to interpolation, potentially improving the accuracy and efficiency of overset grid approaches. Two dimensional results are given for laminar viscous flows, showing excellent agreement with established codes.

I. Introduction

Difficulties in obtaining quality meshes for viscous flow simulations has recently generated interest in so-called meshless schemes. As the name implies, meshless schemes do not require the use of a traditional mesh for the underlying spatial discretization. Instead, these schemes use local clouds of points from which flow derivatives may be approximated for use in a set of governing partial differential equations. Often, the method of least squares is employed on the local point clouds to compute derivatives for use in collocation methods based on the strong form of the governing equations.

Underlying the development of meshless methods is the hypothesis that generating suitable point clouds for a meshless scheme relieves to some degree the burden of obtaining a traditional mesh for complex configurations. This hypothesis has yet to be proven in practice. However, the advancing point technique of Löhner and Oñate[1] demonstrated significant progress in point generation for meshless schemes. Their work addressed one of the two underlying issues of discretization for meshless shemes: obtaining a suitable point distribution in space. The second issue is the selection of local clouds of points within the distribution such that solution representation on local scattered data is compact and accurate. Choosing suitable clouds implies the identification of certain metrics to assess cloud quality. For least squares methods, the condition number of the least squares matrix arising from the local cloud has been the most widely used metric for determining the "goodness" of a local discretization.[2,3]

Point generation aside, recent progress in the area of meshless methods for CFD computations has shown great promise in terms of accuracy and efficiency.[2–5] Katz and Jameson[2] have shown that meshless methods applied to inviscid problems in two dimensions produce accurate results compared to the FLO 76 finite volume code of Jameson. The accuracy was shown for transonic flow problems, in which the meshless method produced the correct shock jumps and locations despite the lack of a formal proof of conservation. The method also exhibited good convergence with the development of "multicloud" within the meshless framework. In addition, Luo, Baum, and Löhner[4] have used a meshless method to compute boundary values within a Cartesian mesh framework. Ghosh and Deshpande[6] have applied a meshless method to inviscid flow problems using a kinetic/Boltzman discretization. Sridar and Balakrishnan[3] have developed an upwind meshless solver which they demonstrated on subsonic and transonic test cases. An important aspect of

[*] PhD Candidate, Department of Aeronatics and Astronautics, Stanford, CA, AIAA Member.

[†] Thomas V. Jones Professor of Engineering, Department of Aeronautics and Astronautics, Stanford, CA, AIAA Member.

their work was a numerical demonstration showing highly accurate mass conservation even for random point distributions.

This paper addresses both physical accuracy and domain discretization in the context of a meshless method. First, the inviscid meshless scheme of Katz and Jameson[2] is extended to viscous flows. An accurate and stable algorithm for the discretization of the viscous flux terms of the Navier-Stokes equations is highlighted. Next, the meshless scheme is placed within the framework of an overset grid approach as a means of communicating between multiple grids of varying type. The application of the viscous meshless scheme to the overset method shows immense potential to solve real world problems of great complexity. The main advantage of the overset approach is the ability to leverage different grid types to capture widely varying scales and physics present in complex flows. The overhead associated with the use of multiple grid types is largely reflected in the complex procedures used to communicate between grids. The meshless method is shown to be extremely well suited for the task of grid communication, enhancing the efficiency and accuracy of overset shemes. In fact, the meshless scheme may be thought of as yet another "grid" type present in a Chimera scheme,[7] seamlessly connected to other grids. The paper concludes with a several two-dimensional test case results for steady laminar flows, which show excellent agreement with established methods, such as Jameson's FLO103 code, and the unstructured scheme of Mavriplis.[8]

II. Meshless Formulation for the Navier-Stokes Equations

For many classes of fluid flow problems, viscous effects account for a significant portion of flow features. Examples of highly viscous problems are high angle of attack aerodynamics and low Reynolds number flows. The Navier-Stokes equations are often employed to accurately model such flows. The non-dimensional differential form of the Navier-Stokes equations in two dimensions may be expressed as

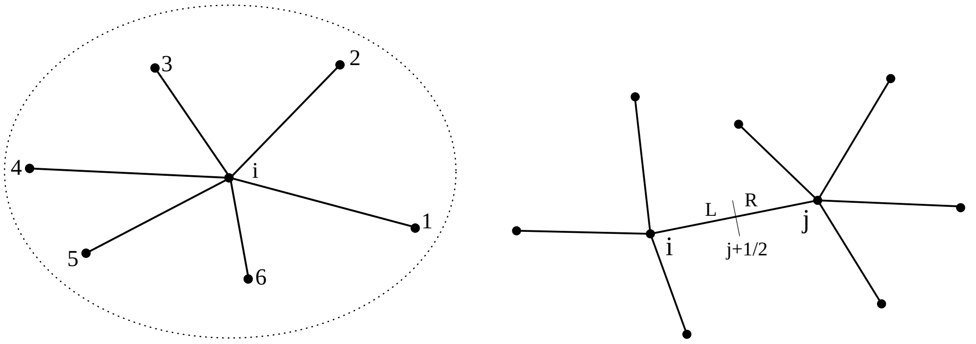
$$\frac{\partial w}{\partial t} + \frac{\partial f_e}{\partial x} + \frac{\partial g_e}{\partial y} = \frac{M_\infty}{Re_\infty}\left(\frac{\partial f_v}{\partial x} + \frac{\partial g_v}{\partial y}\right), \tag{1}$$

where w is the vector of conserved variables, and f_e and g_e are the inviscid flux vectors defined as

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad f_e = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uH \end{pmatrix}, \quad g_e = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + P \\ \rho vH \end{pmatrix}.$$

In the above notation, ρ, u, v, P, E, and $H = E + \frac{P}{\rho}$, are the non-dimensional density, velocity components, pressure, total energy, and total enthalpy. Viscous effects are introduced by the viscous flux terms, f_v and g_v, which are defined as

$$f_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{pmatrix}, \quad g_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{pmatrix}.$$



(a) A local point cloud. (b) A typical edge connecting two point clouds.

Figure 1. Local meshless discretization

The shear stress and heat flux terms are expanded as

$$\begin{aligned}\tau_{xx} &= 2\mu u_x - \frac{2}{3}\mu(u_x + v_y)\\ \tau_{yy} &= 2\mu v_y - \frac{2}{3}\mu(u_x + v_y)\\ \tau_{xy} &= \tau_{xy} = \mu(u_y + v_x)\\ q_x &= -\kappa\frac{\partial T}{\partial x}\\ q_y &= -\kappa\frac{\partial T}{\partial y},\end{aligned}$$

where μ is the viscosity, κ is the coefficient of heat conduction, and T is the temperature. The Navier-Stokes equations are completed by the equation of state and Sutherland's law:

$$E = \frac{P}{(\gamma-1)\rho} + \frac{1}{2}(u^2+v^2), \quad \frac{\mu}{\mu_{ref}} = \left(\frac{T}{T_{ref}}\right)^{\frac{3}{2}} \frac{T_{ref}+T_o}{T+T_o}.$$

The computer simulation of the Navier-Stokes equations requires a consistent and stable discretization. Here the paradigm of partial discretization in space, followed by the numerical solution of the resulting ordinary differential equation in time is adopted. The details of the spatial and temporal aspects of the discretization will now be discussed.

II.A. Spatial Discretization

Instead of volumes or point stencils typically used to discretize partial differential equations, the meshless method makes use of clouds of neighboring points, as shown in Fig.(1(a)). As a preprocessing step, for each point in the domain, a local set of nearest neighbors must be defined. As was demonstrated by Katz and Jameson,[2] the meshless clouds defined for each point may be implicity represented as a list of "edges," or connections between neighboring points. This set of edges completely defines the connectivity over the entire domain.

A number of scattered data approximation schemes may be used to represent the dependent variables on local clouds of points. Many of these schemes lead to the following form for derivatives of a property ϕ at node i with n surrounding nearest neighbors:

$$\frac{\partial \phi}{\partial x} \approx \sum_{j=1}^{n} a_{ij}\Delta\phi_{ij}, \quad \frac{\partial \phi}{\partial y} \approx \sum_{j=1}^{n} b_{ij}\Delta\phi_{ij}, \tag{2}$$

where $\Delta\phi_{ij} = \phi_j - \phi_i$, and a_{ij} and b_{ij} are metric coefficients arising from the local approximation. In this work, a least squares method based on Taylor series is used. Detailes of the least squares procedure are contained in Appendix A. In light of Eq.(2), the Navier-Stokes equations of Eq.(1) may be semi-discretized as

$$\frac{\partial w_i}{\partial t} + \sum_{j=1}^{n} \left(a_{ij}\Delta f_{eij} + b_{ij}\Delta g_{eij}\right) = \frac{M_\infty}{Re_\infty} \sum_{j=1}^{n} \left(a_{ij}\Delta f_{vij} + b_{ij}\Delta g_{vij}\right),$$

or using a directed flux, $F = af + bg$,

$$\frac{\partial w_i}{\partial t} + \sum_{j=1}^{n} \Delta F_{eij} = \frac{M_\infty}{Re_\infty} \sum_{j=1}^{n} \Delta F_{vij}. \tag{3}$$

Unfortunately, the dissipative nature of the viscous terms present in Eq.(3) is not sufficient to damp instabilities due to odd/even decoupling and discontinuities in the solution arising from shock waves. In order to construct a stable scheme, the LED criterion of Jameson[9] is invoked, leading to additive artificial dissipation terms. Consider an edge connecting two nodes, as shown in Fig.(1(b)). If the differencing is extended to the midpoint of the edge $(j + \frac{1}{2})$, Eq.(3) becomes

$$\frac{\partial w_i}{\partial t} + 2\sum_{j=1}^{n} \Delta F_{eij+\frac{1}{2}} = \frac{2M_\infty}{Re_\infty} \sum_{j=1}^{n} \Delta F_{vij+\frac{1}{2}}. \tag{4}$$

Diffusive terms may be added by defining the midpoint Euler flux as

$$F_{ej+\frac{1}{2}} = \frac{1}{2}(F_{ei} + F_{ej}) - \frac{1}{2} d_{j+\frac{1}{2}} \tag{5}$$

The diffusive term may be constructed from any number of schemes including scalar and characteristic schemes. In this work Jameson's CUSP scheme[9] is used:

$$d_{j+\frac{1}{2}} = \alpha^* c(w_R - w_L) + \beta(F_{eR} - F_{eL}). \tag{6}$$

Left and right states are formed using SLIP[9] reconstruction.

In addition to the Euler flux, the viscous flux must also be computed at the edge midpoint as required by Eq.(4). If care is not taken, the discretized viscous terms may lead to odd/even decoupling and eventual solution instability. A stable viscous discretization is obtained by carefully observing the nature of the viscous fluxes. Unlike the Euler flux, the viscous flux terms contain gradients of the velocity and temperature, ∇u, ∇v, and ∇T. These gradients may be computed with the same least squares procedure as Eq.(2). If the midpoint viscous fluxes are computed using an arithmetic average of these gradients, odd/even decoupling will ensue as a result of the recycling of the gradients to form the flux derivatives. Instead, a modified average of the velocity and temperature gradients may be computed as:[10]

$$\nabla\phi_{j+\frac{1}{2}} = \overline{\nabla\phi_{j+\frac{1}{2}}} - \left(\overline{\nabla\phi_{j+\frac{1}{2}}} \cdot \vec{s}_{ij} - \frac{\phi_j - \phi_i}{|\vec{r}_j - \vec{r}_i|}\right)\vec{s}_{ij}, \tag{7}$$

where $\overline{\nabla\phi_{j+\frac{1}{2}}} = \frac{1}{2}(\nabla\phi_i + \nabla\phi_j)$ is the average of the gradients at i and j using the least squares method of Eq.(2), and $\vec{s}_{ij} = \frac{\vec{r}_j - \vec{r}_i}{|\vec{r}_j - \vec{r}_i|}$ is the unit vector in the direction of the edge. Here, ϕ represents velocity and temperature. For example, the shear stress term, τ_{xy}, is computed as

$$\tau_{xy,j+\frac{1}{2}} = \frac{1}{2}(\mu_i + \mu_j)(u_{y,j+\frac{1}{2}} + v_{x,j+\frac{1}{2}}),$$

where the derivatives of velocity are computed as shown in Eq.(7). Finally, the semi-discretized form of the Navier-Stokes equations may be expressed as

$$\frac{\partial w_i}{\partial t} + R_i = 0, \tag{8}$$

where

$$R_i = Q_i - D_i, \tag{9}$$

$$Q_i = \sum_{j=1}^{n} \Delta F_{eij}, \quad D_i = \sum_{j=1}^{n} d_{ij+\frac{1}{2}} + \frac{2M_\infty}{Re_\infty} \sum_{j=1}^{n} \Delta F_{vij+\frac{1}{2}}. \tag{10}$$

It is widely recognized, and rightfully so, that meshless methods are not formally conservative. This is due to the fact that the algorithm solves an uncoupled set of least squares problems over the domain, leading to least-squares coefficients which are not aligned across edges. However, for certain specialized cases, meshless methods are conservative, such as on a Cartesian mesh, where the derivatives reduce to central differences. Though conservation is not attained in general, meshless schemes computed on well-conditioned local clouds of points appear to mimic the behavior of conservative schemes, often obtaining the correct shock jumps at the correct locations as compared with finite volume methods.[2]

II.B. Integration to Steady State

Equation (8) represents a system of ordinary differential equations, which may be integrated in psuedo-time to reach steady state. In this work, Jameson's modified Runge Kutta schemes for steady state integration have been used.[9] The schemes integrate the convective (Q) and diffusive (D) portions of the residual (R) separately. An m-stage scheme may be expressed as

$$\begin{aligned}
w^{(n+1,0)} &= w^n \\
&\vdots \\
w^{(n+1,k)} &= w^n - \alpha_k \Delta t \left(Q^{(k-1)} + D^{(k-1)} \right) \\
&\vdots \\
w^{(n+1)} &= w^{(n+1,m)},
\end{aligned}$$

where

$$\begin{aligned}
Q^{(0)} &= Q(w^n), \quad D^{(0)} = D(w^n) \\
&\vdots \\
Q^{(k)} &= Q(w^{(n+1,k)}), \quad D^{(k)} = \beta_k D(w^{(n+1,k)}) + (1-\beta_k) D^{(k-1)}.
\end{aligned}$$

The following five-stage scheme has been used with much success:

$$\begin{aligned}
\alpha_1 &= \frac{1}{4} & \beta_1 &= 1 \\
\alpha_2 &= \frac{1}{6} & \beta_2 &= 0 \\
\alpha_3 &= \frac{3}{8} & \beta_3 &= 0.56 \\
\alpha_4 &= \frac{1}{2} & \beta_4 &= 0 \\
\alpha_5 &= 1 & \beta_5 &= 0.44
\end{aligned}$$

Local time stepping may be employed by estimating the spectral radius of the spatial operator at each meshless node. The spectral radius of the inviscid flux for the meshless scheme, λ_I, may be approximated as

$$\lambda_I \approx \sum_{j=1}^{n} \left(|a_{ij} u_i + b_{ij} v_i| + \sqrt{a_{ij}^2 + b_{ij}^2} c_i \right), \tag{11}$$

where u,v, and c are the local velocities and speed of sound. Adapting the derivation of Mavriplis for unstructured meshes[8] to the meshless scheme, the spectral radius of the viscous flux, λ_V may be approximated as

$$\lambda_V \approx \frac{\gamma^{\frac{3}{2}} M_\infty}{Re_\infty Pr} \sum_{j=1}^{n} \frac{\mu_{ij}}{\rho_{ij}} \left(a_{ij}^2 + b_{ij}^2 \right), \tag{12}$$

where Pr is the Prandtl number and μ_{ij} and ρ_{ij} are the average viscosity and density across the edge connecting nodes i and j. The local time step estimate for the meshless scheme is then

$$dt_i = \frac{CFL}{\lambda_I + \lambda_V} \tag{13}$$

Using the above time step estimates, $CFL = 3$ was obtained. Additionally, implicit residual averaging may be used to increase the support for each node, roughly doubling the allowable CFL number. The smoothing procedure is performed in a manner similar to other unstructured grid solvers,[11] replacing neigboring cells with points in the meshless cloud. In practice, $CFL = 8$ has been used in a variety of test cases with the residual averaging procedure turned on.

III. Application to Overset Grids

While it has been shown that meshless methods provide accurate results quite efficiently, there should be clear justification for preferring such a method over a conventional CFD approach. After all, formal conservation is sacrificed using meshless methods. Additionally, distributing points and forming meshless clouds in an optimal manner is non-trivial, potentially on the order of difficulty as generating a mesh. While not entirely relieving the problems associated with mesh generation, meshless methods may provide a significant advantage within some computational frameworks. This section highlights such a framework: the application of a meshless method to overset grids as a means of coupling grid components together.

In the overset approach, the solution domain is decomposed into near-body regions, which extend a short distance away from solid bodies, and off-body regions, which cover the remainder of the domain to the far field boundaries.[12] Near-body grids are typically unstructured, prismatic, or curvilinear grids capable of anisotropic refinement to capture boundary layers. Off-body grids are typically composed of patches of automatically generated Cartesian grids which present significant gains in algorithmic efficiency and low memory requirements. The Cartesian grids are often adaptive, capable of refining to geometry and/or solution features.[13] The paradigm of using multiple grids of varying type facilitates solutions for complex problems including simulations involving multiple bodies in relative motion.

Traditionally, domain connectivity for overset grids has been accomplished with linear interpolation. The connectivity procedure involves the identification of donor cells and recipient nodes along grid boundaries. Meakin[14] has shown that spatial and temporal accuracy may be maintained using an interpolation method if grids in the interpolation region are sufficiently refined, in spite of a loss of formal conservation. Because the entire process of donor identification and interpolation can cost 10 to 20 percent of a flow solver iteration,[15] efficiency in the domain connectivity algorithm is critical.

In this section, a novel approach to domain connectivity is proposed using the meshless method described in the previous section. The meshless approach presents potential advantages in accuracy and efficiency over interpolation. First, since the meshless method uses the fluid equations integrated in time just as other grid components, potential time lagging errors for unsteady problems with implicit integration[14] using large time steps could be alleviated to some degree. Accuracy may also be improved in the presence of solution discontinuities and high gradients, since the meshless method, unlike interpolation, obeys the proper characteristic propagation of information. In addition to accuracy, the task of defining a meshless cloud, as discussed in this section, appears to be less computationally demanding than donor cell identification, leading to gains in algorithm efficiency. Though many of these claims require further investigation in a variety of scenarios, preliminary results are encouraging.

III.A. Determining Meshless Clouds in the Interface Region

An efficient method for determining which points receive a meshless discretization and the selection of point clouds will now be described. In short, all points in all grid components of the overset system that do not have a complete stencil for their respective solution procedures are flagged as meshless. This means these points will receive a meshless spatial discretization, while all other points receive a discretization from their respective grid method. While overset grid simulations may contain multiple aerodynamic bodies each containing multiple near-body grids, as well as an arbitrary number of off body Cartesian blocks, the following procedure will be limited to a single body possessing a single near-body grid embedded in a single off body Cartesian block. Future work will be generalized to handle more complex test cases. The steps for determining the meshless points and clouds may be summarized as follows:

1. Determine near-body meshless points
2. Determine Cartesian meshless points
3. Determine near-body support points
4. Determine Cartesian support points

The above steps will now be described in detail and are illustrated in Fig.(2).

III.A.1. *Determine near-body meshless points*

First, near-body meshless points are identified. Near-body meshless points are points within the near-body which do not have a complete finite volume stencil arising from domain boundaries. For conventional second-order methods, the near-body meshless points are the two outermost layers of the near-body grid system. Near-body meshless points inherit the connectivity that is available from the near-body, as shown by the edges in Fig.(2(b)). Once the near-body meshless points are identified, any number of Cartesian hole-cutting algorithms may be employed to blank cartesian cells which lie within the near-body domain. In this work, a ray-tracing based hole-cutting algorithm has been used.[16]

III.A.2. *Determine Cartesian meshless points*

Second, Cartesian meshless points are identified, and their connectivity to the near-body meshless points is established. For each outermost near-body meshless point, the four Cartesian points which contain it are identified. The (I, J) coordinates of the lower left Cartesian point may be found from

$$I_1 = FLOOR\left(\frac{(x_n - x_L)}{dx}\right) \tag{14}$$

$$J_1 = FLOOR\left(\frac{(y_n - y_L)}{dy}\right). \tag{15}$$

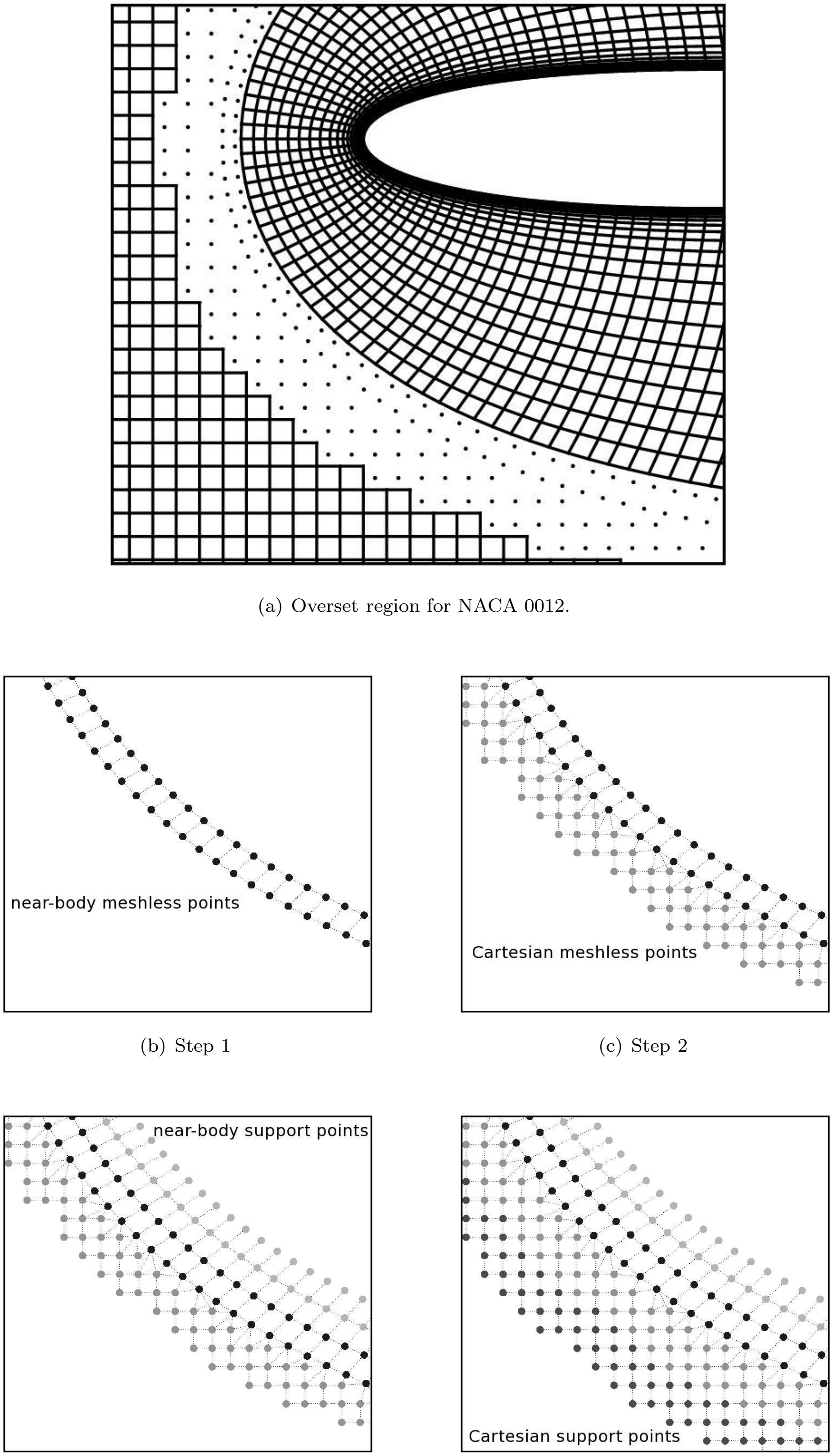
Of the four Cartesian points containing the near-body point, those points that are not blanked from the hole-cutting procedure are flagged as Cartesian meshless points, and their connectivity to the near-body point which found them is established.

While the result of this procedure results in well-conditioned meshless clouds for all near-body points, in general there will remain Cartesian points along the hole-cut boundary which do not connect to any near-body point. These Cartesian points will remain poorly conditioned unless they are connected to nearby near-body meshless points. The procedure to establish these connections is detailed in Fig.(3). First, the blanked points within the vicinity of the hole-cut surface are flagged with their nearest near-body meshless point. Next, connections are formed from the Cartesian points with incomplete stencils to the flag of the blanked Cartesian point. This procedure is extremely fast, since proximity searches may be done from the perspective of the near-body searching for Cartesian points in Cartesian index space, similar to Eq.(14)

III.A.3. *Determine near-body support points*

The previous two steps involved the determination of near-body and Cartesian meshless points. These points receive a meshless discretization in space. The next two steps involve the determination of support points, which do not receive a meshless discretization, but merely provide boundary condition layers for the meshless points. The support points are owned by other grid components and receive the spatial discretization typical of their respective grids.

Near-body support points are determined by the requirement that each near-body meshless node must have a well-defined least squares stencil. Additionally, each of these nearest neighbors must have a well-defined least squares stencil. These requirements would suggest picking points in the near-body grid corresponding to the nodes in Fig.(2(d)). For structured near-body grids, determination of near-body support points is trivial. Again, near-body support nodes inherit the connectivity of the near-body grid.



(a) Overset region for NACA 0012.

(b) Step 1

(c) Step 2

(d) Step 3

(e) Step 4

Figure 2. Procedure for determining the meshless interface

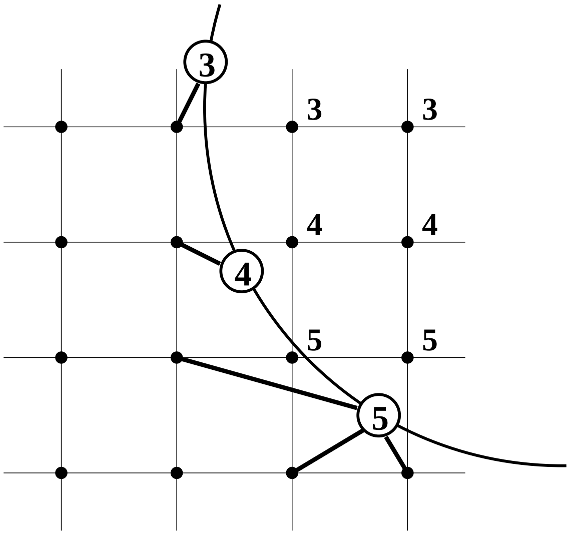


Figure 3. Connecting Cartesian points to near-body points.

III.A.4. Determine Cartesian support points

Similar to near-body support points, Cartesian support points do not receive a meshless discretization, but provide an adequate boundary region for the Cartesian meshless points. The selection of the Cartesian support points is trivial and is performed by marching away from the Cartesian meshless nodes along coordinate directions, until their meshless stencil requirements are met. The procedure results in adding the nodes shown in Fig.(2(e)). Once again the connectivity for these nodes is borrowed from the Cartesian mesh itself.

Once all meshless points, support points, and connecting edges have been identified, the condition number, κ, of the least squares matrix at each node is computed to assess local cloud quality. In practice, for $\kappa < 4$, stability and good convergence have been observed. The quality of local clouds as reflected in the condition number appears to be strongly correlated to the discrepancy in resolution of near-body and off-body grid components at the interface region. As the off-body spacing becomes more than a factor of about three larger or smaller than the near-body grid, the condition of local clouds is greatly compromised. Clearly, matching grid resolution in the interface is needed for stability and accuracy for the meshless approach just as for interpolation. This is consistent with the findings of Meakin[14] who found that accuracy for time accurate overset problems is most strongly influenced by mesh resolution of grid components in the overlap region.

From the above procedures it is clear that all algorithms required to obtain suitable meshless clouds for use in the overset scheme are quite simple and fast. These simple algorithms contrast with the complex and relatively expensive bin or octree search techniques to find near-body donor cells for off-body Cartesian grids.

III.B. Implementation using Python

It should be emphasized that for steady-state problems, the entire meshless connectivity routines may be performed in a preprocess step and accessed throughout the computation. The problem of grid communication is then reduced to the proper injection of Dirichlet boundary conditions for each grid component in the domain at each time step. Boundary conditions for each domain component are injected, the solution is advanced in pseudo-time, and the computation proceeds until steady-state is acheived. In this work, the iterative process is driven at the top level by a Python script. The Python script calls the necessary solution procedures (near-body, off-body, and meshless), which have been compiled as shared object files to give Python access to data at run time. This approach requires minimal modification to existing codes to be placed in the context of a globally hybrid scheme, and has recently been implemented successfully by Sitaraman et. al.[13]

A simplified version of the top level Python script used in this work is shown in Fig.(4). Note that each

```
# import necessary modules
import flo103
import cartsolver
import meshless

# Allocation and initialization steps
flo103.dimension()
flo103.initialize()
if bc==-1:
    cartsolver.dimensionc()
    cartsolver.initializec()
    meshless.dimensionm()
    meshless.initializem()

# Iterate until steady state is reached
time      = 0
converged = False
while not converged:
    time  = time+1
    converged = flo103.takestep()
    if bc==-1:
        convergedc = cartsolver.takestepc()
        convergedm = meshless.takestepm()
    elif bc>=0:
        flo103.farfield()
```

Figure 4. Top level Python driver for hybrid meshless scheme

solution procedure is imported as a shared object and called as a Python routine. The near-body object is a close variant of Jameson's FLO103 two dimensional viscous code with only slight modifications to fit within the Python framework. The off-body solver is also based on FLO103, but streamlined to handle Cartesian meshes. The meshless object is precisely the viscous scheme described in this work. Note that FLO103 may be run stand-alone or in the hybrid context in this code infrastructure by simply switching the flag `bc` to various values. In summary, the Python infrastructure is an extremely uninstrusive method of coupling existing codes to form a hybrid scheme. Only minimal modifications to stand-alone source codes need to be made, and only at the highest levels which are actually called by the Python script.

IV. Results and conclusions

The results in this section were obtained using the hybrid mesh shown in Fig.(5(a)) around the NACA 0012 airfoil. The near-body mesh was generated using the mesh generation package of FLO103, with certain modifications to smooth the wake spacing behind the boundary layer. The objective of smoothing the wake region was to provide even spacing from which to interface with the meshless scheme. All the results in this section are for relatively low Reynolds number, so high reslution of the wake is only critical a short distance behind the airfoil. For higher Reynolds numbers, the smoothing region should be extended several chord lengths downstream to properly capture the wake.

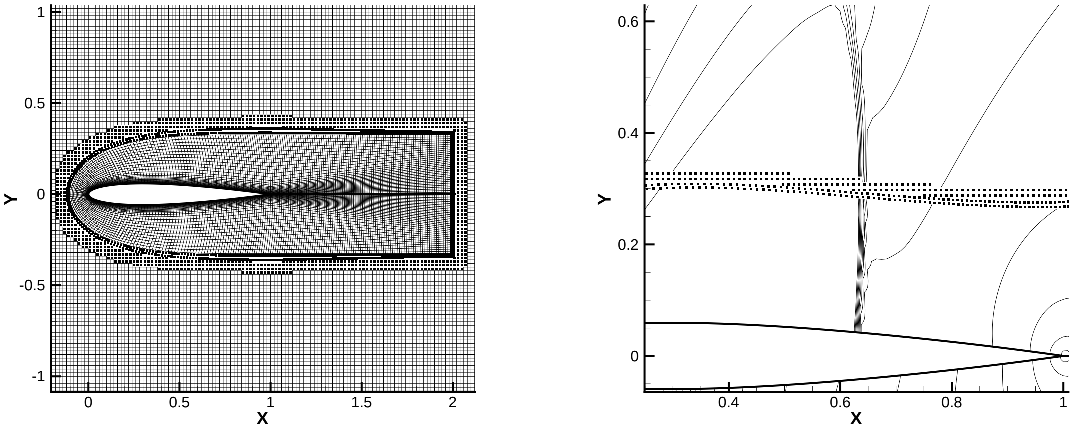
Shown in Fig.(5(b)) is a close-up view of the interface region showing a shock in transonic flow. Note the clean capturing of the discontinuity and the seamless transition across the meshless interface. This case highlights one strength of the meshless scheme within the hybrid approach, that of capturing regions of high solution gradients. Clearly, the meshless scheme is extremely well-suited to propagate fluid information across meshes since it makes use of the underlying physical model (Euler or Navier-Stokes).

The hybrid scheme also performs extremely well at predicting separation compared with the scheme of Mavriplis.[8] The predicted separation point at 82.7% for the NACA 0012 at $M = 0.5$, $\alpha = 0^o$ and $Re = 5000$ is in the middle of the range predicted by Mavriplis as shown in Fig.(6(a)) and displayed in Table 1. Contours of Mach number for this case are shown in Fig.(6(b)).

The remaining figures, Figs.(7-10), containing solution contours and surface pressure plots, highlight the accuracy of the hybrid meshless scheme for laminar viscous flows in two-dimensions. The surface pressure curves for the hybrid scheme are overplotted with FLO103 stand-alone results, showing excellent agreement. Accompanying each set of plots is a table of lift and drag coefficients due to pressure comparing the hybrid scheme with FLO103, and when available, the viscous scheme of Mavirplis.[8] The lift and drag coefficients compare quite well for all cases tested. The conclusion to be drawn from these results is that the meshless interface provides an accurate means of computing viscous flows for complex configurations within an overset approach. Solution features are captured cleanly and seamlessly, as indicated by the solution contours. Global accuracy is attained, as indicated by the lift and drag coefficients. The results appear very promising, and encourage further investigation for more complex problems, such as high Reynolds number flows and flows in three dimensions.

V. Acknowledgements

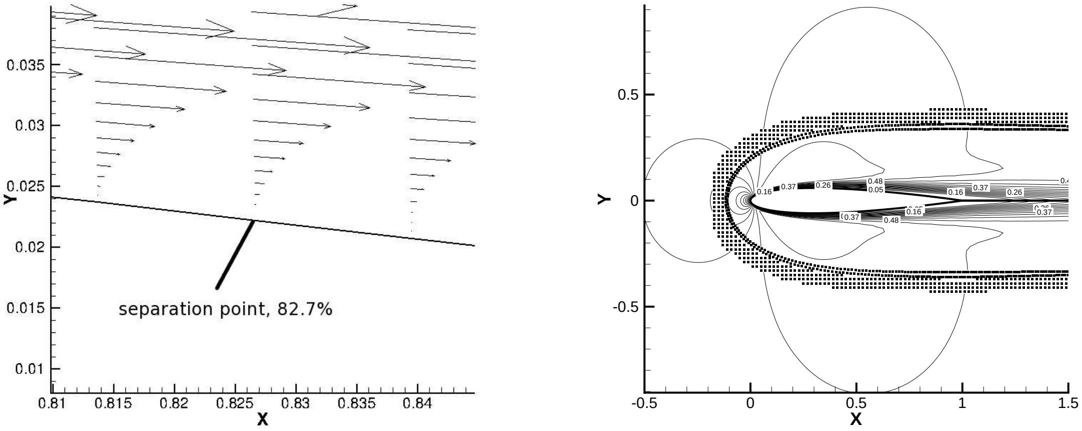
Development was performed primarily under a National Defense Science and Engineering Graduate (NDSEG) fellowship. The first author has benefited greatly from the NDSEG program administered through the High Performance Computing Modernization Office of the Department of Defense. Likewise, the second author has benefited greatly from the long term and continuing support of the AFOSR Computational Mathematics Program directed by Dr. Fariba Fahroo. We would also like to acknowledge the members of the HPC Institute for Advanced Rotorcraft Modeling and Simulation (HIARMS) team located at NASA Ames Research Center, whose continuing support has facilitated certain aspects of this work. In particular, Dr. Roger Strawn, Dr. Andrew Wissink, Dr. Venkateswaran Sankaran, and Dr. Jayanarayanan Sitaraman have been instrumental in the development of this work.



(a) Hybrid mesh with meshless interface

(b) Shock capturing ability for inviscid transonic flow

Figure 5. Flow over NACA 0012, $M = 0.8$, $\alpha = 1.25^o$, **showing mesh**



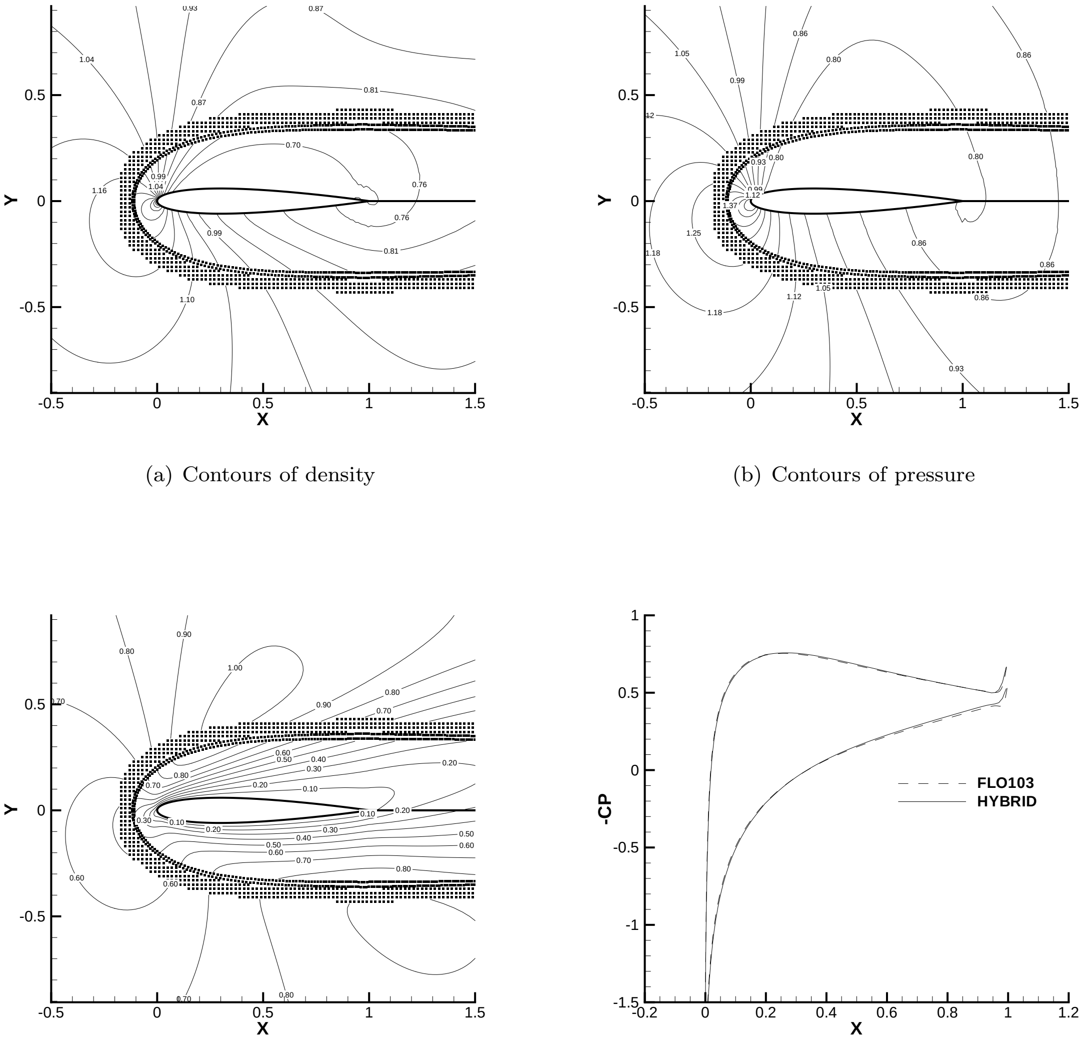
(a) Separation point

(b) Contours of Mach number

Figure 6. Flow over NACA 0012, $M = 0.5$, $\alpha = 0^o$, $Re = 5000$

Table 1. Comparison of separation point location

	separation location
Mavriplis[8]	80.9%-83.4%
Hybrid	82.7%



(a) Contours of density

(b) Contours of pressure

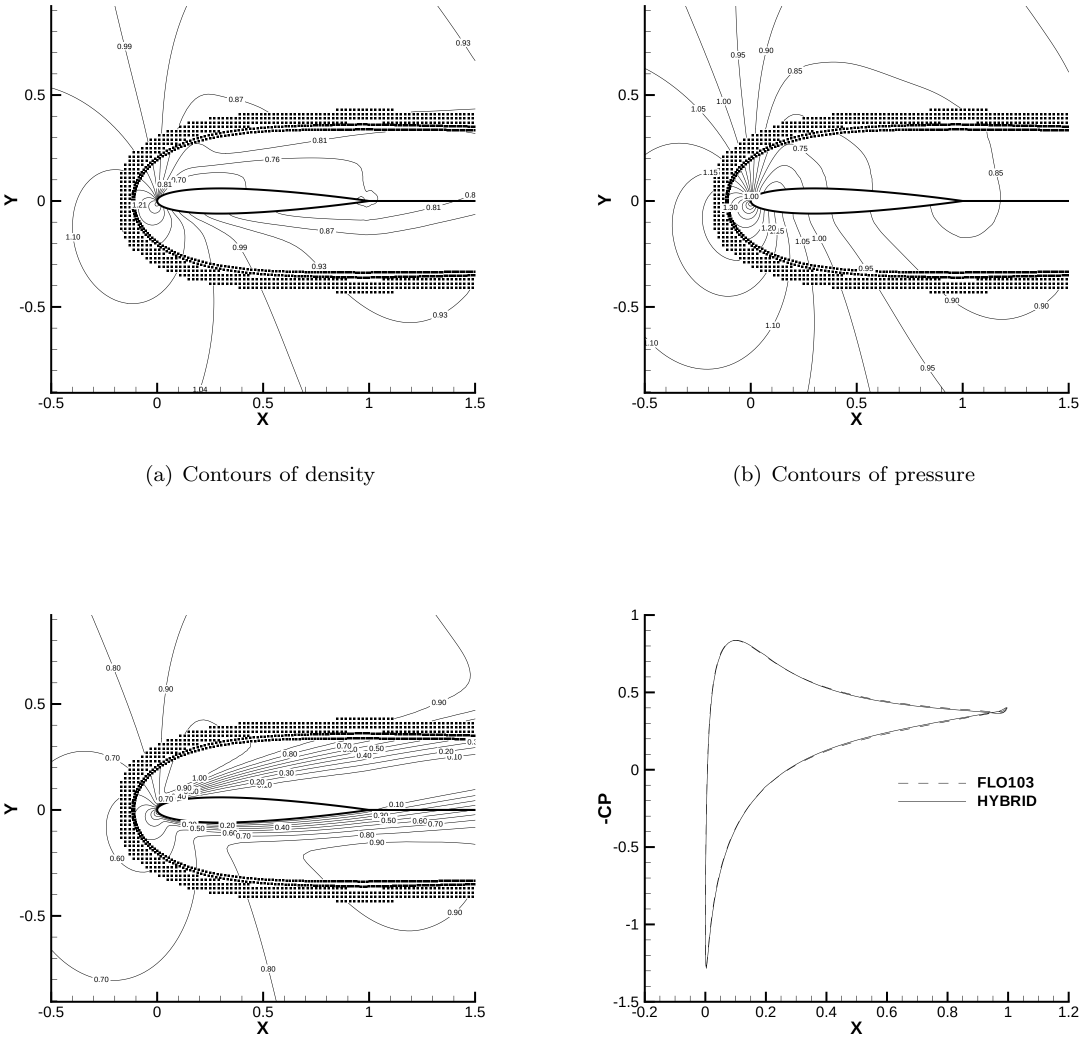
(c) Contours of Mach number

(d) Surface pressure coefficient

Figure 7. Flow over NACA 0012, $M = 0.8$, $\alpha = 10^o$, $Re = 73$

Table 2. Lift and drag coefficients due to pressure

	c_l	c_d
FLO 103	0.5480	0.2083
Mavriplis[8]	0.5886	0.2191
Hybrid	0.5446	0.2114



(a) Contours of density

(b) Contours of pressure

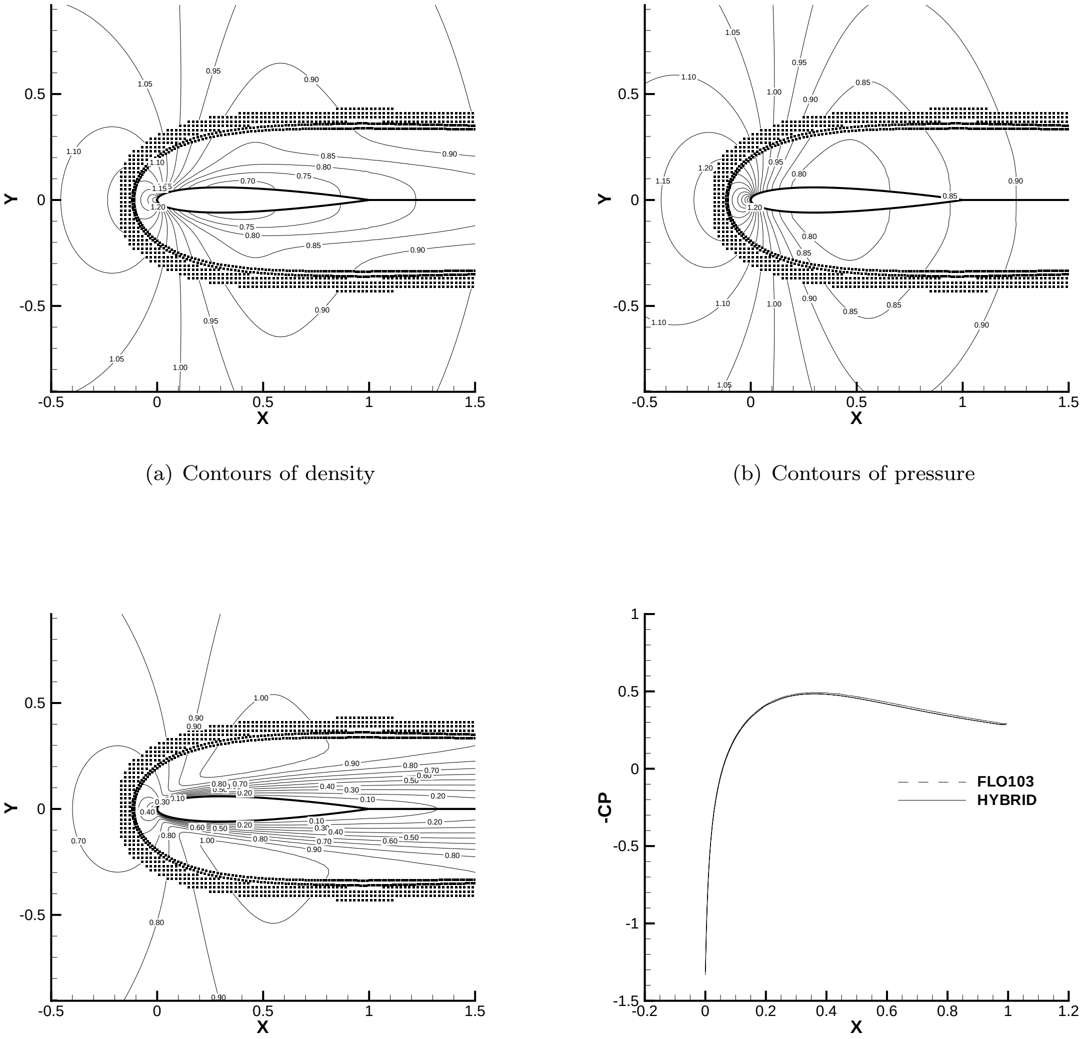
(c) Contours of Mach number

(d) Surface pressure coefficient

Figure 8. Flow over NACA 0012, $M = 0.8$, $\alpha = 10^o$, $Re = 500$

Table 3. Lift and drag coefficients due to pressure

	c_l	c_d
FLO 103	0.4311	0.1421
Mavriplis[8]	0.4469	0.1474
Hybrid	0.4234	0.1420



(a) Contours of density

(b) Contours of pressure

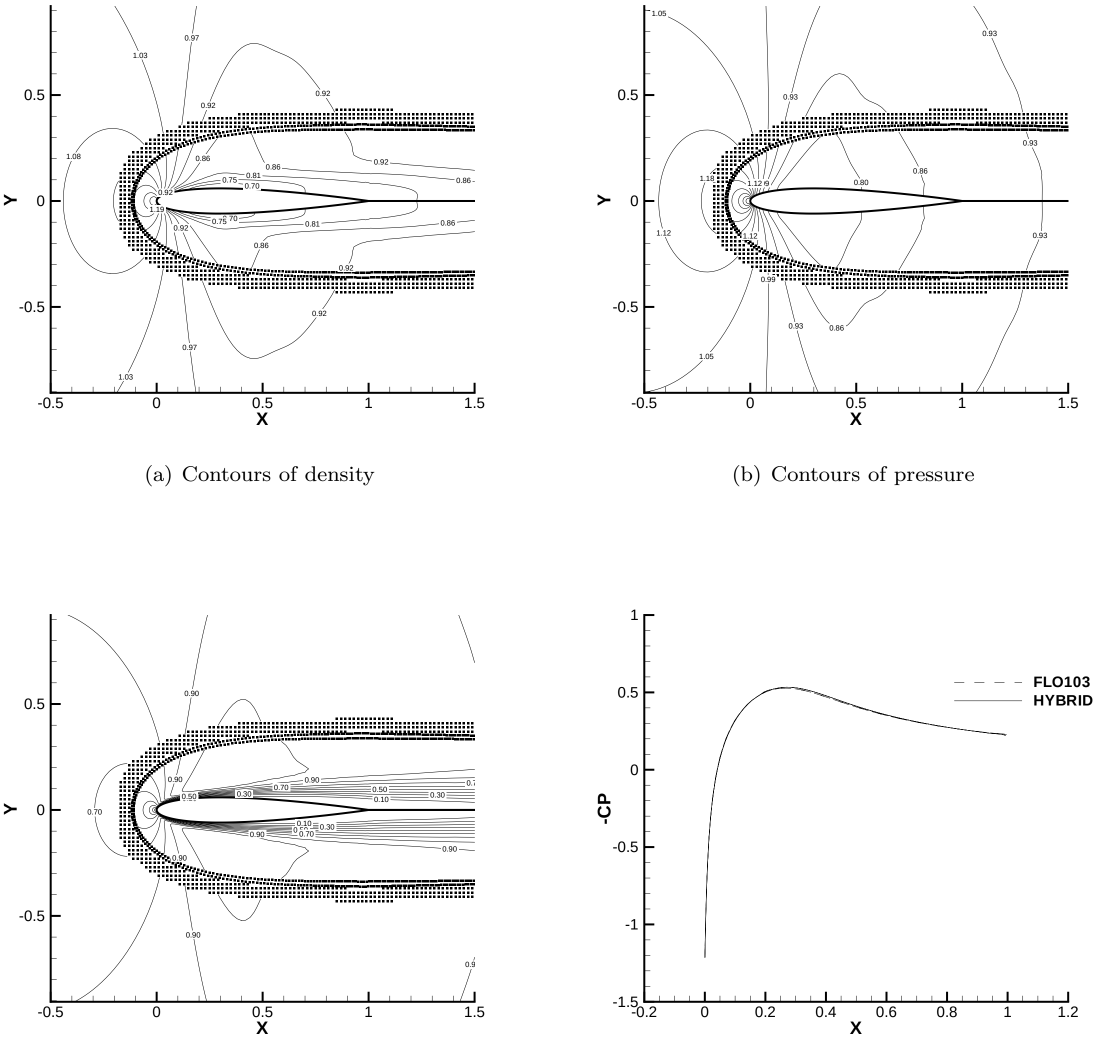
(c) Contours of Mach number

(d) Surface pressure coefficient

Figure 9. Flow over NACA 0012, $M = 0.85$, $\alpha = 0^o$, $Re = 500$

Table 4. Lift and drag coefficients due to pressure

	c_l	c_d
FLO 103	0.0000	0.0832
Hybrid	0.0000	0.0827



(a) Contours of density

(b) Contours of pressure

(c) Contours of Mach number

(d) Surface pressure coefficient

Figure 10. Flow over NACA 0012, $M = 0.85$**,** $\alpha = 0^o$**,** $Re = 2000$

Table 5. Lift and drag coefficients due to pressure

	c_l	c_d
FLO 103	0.0000	0.0608
Hybrid	0.0000	0.0616

VI. Appendix A: Linear least squares coefficients in two dimensions

Retaining only linear terms, a quantity ϕ may be expanded from i to j in a Taylor series as

$$\Delta\phi_{ij} = \Delta x_{ij}\frac{\partial\phi}{\partial x} + \Delta y_{ij}\frac{\partial\phi}{\partial y}. \tag{16}$$

Performing such an expansion at n points, a weighted least squares problem may be expressed as[5]

$$min\sum_{j=1}^{n} w_{ij}\left[\Delta\phi_{ij} - \Delta x_{ij}\frac{\partial\phi}{\partial x} + \Delta y_{ij}\frac{\partial\phi}{\partial y}\right]^2 \quad wrt \quad \frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \tag{17}$$

where $n \geq D+1$, and D is the number of spatial dimensions. A simple inverse distance weighting function of the following form may be used:

$$w_{ij} = \frac{1}{(\Delta x_{ij}^2 + \Delta y_{ij}^2)^{p/2}}, \quad p \geq 0. \tag{18}$$

In this work, a value of $p = 1$ was used. Constructing the normal equations for the least squares problem in Eq.(17) yields

$$A^T A\mathbf{d} \quad = \quad A^T\mathbf{b} \tag{19}$$

$$\begin{bmatrix} \sum w\Delta x^2 & \sum w\Delta x\Delta y \\ \sum w\Delta x\Delta y & \sum w\Delta y^2 \end{bmatrix} \begin{bmatrix} \frac{\partial\phi}{\partial x} \\ \frac{\partial\phi}{\partial y} \end{bmatrix} = \begin{bmatrix} w_{i1}\Delta x_1 & w_{i2}\Delta x_2 & \cdots & w_{in}\Delta x_n \\ w_{i1}\Delta y_1 & w_{i2}\Delta y_2 & \cdots & w_{in}\Delta y_n \end{bmatrix} \begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \\ \vdots \\ \Delta\phi_n \end{bmatrix} \tag{20}$$

Computing $(A^T A)^{-1}A^T$ leads to the following linear approximation to the derivatives:

$$\frac{\partial\phi}{\partial x} \approx \sum_{j=1}^{n} a_{ij}\Delta\phi_{ij}, \quad a_{ij} = \frac{w_{ij}\Delta x_{ij}\sum w\Delta y^2 - w_{ij}\Delta y_{ij}\sum w\Delta x\Delta y}{\sum w\Delta x^2\sum w\Delta y^2 - \left(\sum w\Delta x\Delta y\right)^2} \tag{21}$$

$$\frac{\partial\phi}{\partial y} \approx \sum_{j=1}^{n} b_{ij}\Delta\phi_{ij}, \quad b_{ij} = \frac{w_{ij}\Delta y_{ij}\sum w\Delta x^2 - w_{ij}\Delta x_{ij}\sum w\Delta x\Delta y}{\sum w\Delta x^2\sum w\Delta y^2 - \left(\sum w\Delta x\Delta y\right)^2} \tag{22}$$

It is worth noting that solving the above least squares problem via the normal equations requires the division by the determinant of $A^T A$, which is $O(\Delta x^4)$, potentially leading to numerical instability. However, as pointed out by Mavriplis,[17] inverse distance weighting greatly improves the condition number of $A^T A$, resulting in a determinant which is $O(1)$.

References

[1] Löhner, R. and Onate, E., "An Advancing Front Point Generation Technique," *Commun. Numer. Meth. Engng.*, Vol. 14, 1998, pp. 1097–1108.

[2] Katz, A. and Jameson, A., "Edge-based Meshless Methods for Compressible Flow Simulations," *AIAA paper* submitted, AIAA 46th Aerospace Sciences Meeting, Reno, NV, January 2008.

[3] Sridar, M. and Balakrishnan, N., "An Upwind Finite Difference Scheme for Meshless Solvers," *Journal of Computational Physics*, Vol. 189, 2003, pp. 1–29.

[4] Luo, H., Baum, J., , and Löhner, R., "A Hybrid Cartesian Grid and Gridless Method for Compressible Flows," *AIAA paper* 2005-492, AIAA 43rd Aerospace Sciences Meeting, Reno, NV, January 2005.

[5] Praveen, C., *Develpment and Application of Kinetic Meshless Methods for Euler Equations*, Ph.D. thesis, July 2004.

[6] Ghosh, A. K. and Deshpande, S. M., "Least Squares Kinetic Upwind Method for Inviscid Compressible Flows," *AIAA paper* 1995-1735, AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 1995.

[7] Steger, J., Dougherty, F., and Benek, J., "A Chimera Grid Scheme," Tech. rep., ASME Mini-Symposium on Advances in Grid Generation, Houston, TX, June 1983.

[8] Mavriplis, D., Jameson, A., and Martinelli, L., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," *AIAA paper* 89-0120, AIAA 27th Aerospace Sciences Meeting, Reno, NV, January 1989.

[9] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 1 Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence," *International Journal of Computational Fluid Dynamics*, Vol. 4, 1995, pp. 171–218.

[10] May, G. and Jameson, A., "Unstructured Algorithms for Inviscid and Viscous Flows Embedded in a Unified Solver Architecture: Flo3xx," *AIAA paper* 2005-0318, AIAA 43rd Aerospace Sciences Meeting, Reno, NV, January 2005.

[11] Jameson, A. and Mavriplis, D., "Finite volume Solution of the Two-dimensional Euler Equations on a Regular Triangular Mesh," *AIAA Journal*, Vol. 24, 1986, pp. 611–618.

[12] Meakin, R., "The Chimera Method of Simulation for Unsteady Three-Dimensional Viscous Flow," *Computational Fluid Dynamics Review 1995*, edited by M. Hafez and K. Oshima, John Wiley and Sons Ltd, 1995, pp. 70–86.

[13] Sitaraman, J., Katz, A., Jayaraman, B., Wissink, A., and Sankaran, V., "Evaluation of a Multi-Solver Paradigm for CFD Using Unstructured and Structured Adaptive Cartesian Grids," *AIAA paper* 2008-660, 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2008.

[14] Meakin, R., "On the Spatial and Temporal Accuracy of Overset Grid Methods for Moving Body Problems," *AIAA paper* 1994-1925, AIAA 12th Applied Aerodynamics Conference, Colorado Springs, CO, June 1994.

[15] Meakin, R., Wissink, A., Chan, W., and Pandya, S., "On Strand Grids for Complex Flows," *AIAA paper* 2007-3834, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.

[16] Meakin, R., "Composite Overset Structured Grids," *Handbook of Grid Generation*, CRC Press, 1999.

[17] Mavriplis, D., "Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes," *AIAA paper* 2003-3986, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.