45th AIAA Aerospace Sciences Meeting and Exhibit
8 - 11 January 2007, Reno, Nevada

AIAA 2007-248

High-Order Dynamic Overset Procedure Applied to Moving Body Calculations

Kehinde Alabi* and Foluso Ladeinde.†
Thaerocomp Technical Corp., P. O. Box 1527, Stony Brook, NY, 11790-0609

This paper reports on the development and implementation of a high-order dynamic overset method for moving body calculations. The procedure is incorporated within a high order CFD solver system and is based on high-order static overset methods. The dynamic high-order overset formulation for both the blanking and donor-recipient operations are presented. Results using the high-order dynamic overset method show promise compared to low-order methods.

Nomenclature

α_w = wing or airfoil angle of attack
α_f = flap angle of attack
C_p = pressure coefficient
C_L = lift coefficient
C_D = drag coefficient
C = airfoil or wing chord length
M_∞ = free stream Mach number
P = thermodynamic pressure
x, y, w = Cartesian coordinate system
i, j, k = Spatial scheme indices
ξ, η, ζ = Curvilinear coordinate system
u, v, w = velocity components in the x, y, z Cartesian coordinate directions, respectively

I. INTRODUCTION

The overset procedure has been used to solve practical engineering problems where the geometry is irregular and several overlapping meshes are used to approximate the domain. Using finite-difference procedures with overset grids, complete aircrafts and reentry vehicles[1-5] have been analyzed. In particular, the overset method is ideal for moving body calculations where separate grids are generated around each moving segment thus simplifying the application of motion of the bodies and avoiding the distortion in the grids that is a part of most single block procedures. However, overset methods have typically used low-order interpolation to exchange results in the common regions between adjacent grids.[2,3] When used with low-order flow solvers, second-order interpolation may be adequate for exchanging results across boundaries. With high-order flow solvers, low-order interpolation procedures will reduce solution accuracy, particularly in the vicinity of the overset boundaries. Several authors have examined the use of high-order interpolation procedures[6-9] and reported improved solutions. However, to our knowledge, overset procedures that retain a high-order of accuracy for dynamic or moving grid calculations has yet to be reported. The objective of the current paper is to compare the results of high-order moving body calculations combined with high-order overset method to those of low-order procedures.

Moving body calculations are required for many aerospace systems. Calculations such as those of store or payload separation, VTOL aircraft in take-off and landing, booster rocket or external fuel tank separation require dynamic calculations to accurately model the problem. Challenge problems have been issued by the United States Navy related to the store clearance problem during the design of air vehicles.[10-17] The results of the challenge problems and subsequent research have demonstrated the feasibility of utilizing CFD with wind tunnel tests within the framework of a store certification analysis. However, as noted in the summary report,[11] there is room for improvement both in the accuracy and the speed of obtaining moving boundary CFD results.

* Research Engineer, Ph.D, AIAA Member.
† Director of Research, AIAA Life Member & Associate Fellow.

Copyright © 2007 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

Moving boundary calculations are resource intensive as the solutions have to be evolved accurately in time – requiring the equivalent of multiple static calculations. Using high-order methods in the spatial and temporal schemes may provide the ability to utilize less grid points in the spatial models as well as larger time steps. In a previous work, the current authors[18,19] demonstrated using the multi-disciplinary CFD solver - AEROFLO - with static high-order overset calculations that high-order calculations do provide a higher degree of accuracy while utilizing less grid points. In the current paper, we have extended the overset procedure in AEROFLO to include efficient dynamic capabilities that build upon the static overset features. The implementation and results of this work are presented in following sections. Firstly, the static, high-order overset capability within AEROFLO is summarized, and then the high-order dynamic procedures are presented. Results are presented only for problems with prescribed motion. The design of the overset and CFD procedures are developed such that the specification of motion can be prescribed, 6DOF, or constrained 6DOF. To this end, a high-order force moment calculation utility for overset grids was developed for AEROFLO as a separate module. However, details of this implementation are not included in the current report. Finally, results of high-order, moving grid calculations are presented for a sinusoidally pitching airfoil, a moving flap in an airfoil flap configuration, and static calculations for a separating store from a 3D wing. Comparisons of results of the high-order dynamic calculations to the equivalent low-order ones are also presented.

II. Static Overset Formulation

The overset procedure presented in this paper is discussed under three subtopics or tasks:

- Identification of the overset nodes
- Identification of the donors for the overset nodes
- Interpolation and donation of solution values to the overset nodes

Details of these tasks are provided below.

A. Identification of the overset nodes

Overset nodes are the nodes that lie at the boundary of an overlap region. The boundary of an overlap region may be the natural boundary of a grid or an internal boundary exposed following a hole-cutting process. For a high-order solver, the boundary condition at the edge may require a buffer of several nodes to maintain high-order accuracy. In such a case, additional buffer nodes may be treated as overset nodes. For instance, for a fourth-order solver, at least two nodes at the boundary will require boundary conditions, and thus, will be overset nodes at an overlap boundary. For a sixth-order solver, Fig. 1 illustrates the selection of overset nodes at the boundary of a hole-cut in a grid – labeled "Grid 1." Grid 1 overlaps a second grid, "Grid 2," which contains the solid boundary of the object. There is a buffer of three overset nodes at the boundary of Grid 1 that is needed to maintain high order up to the boundary.

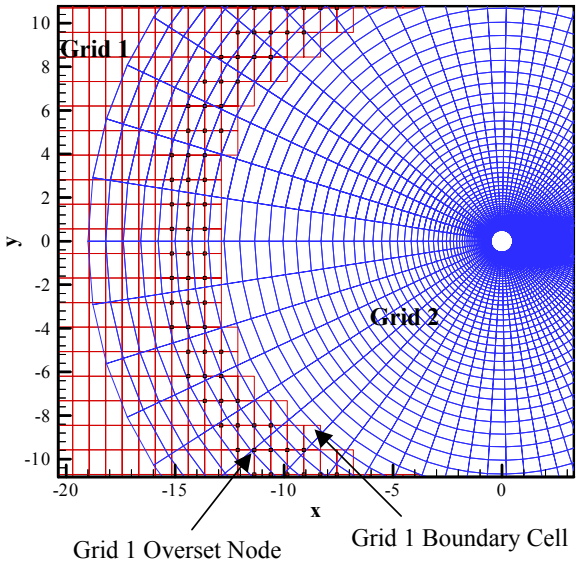


Figure 1. Overset nodes identified as a fringe of nodes at the boundary of a cut region

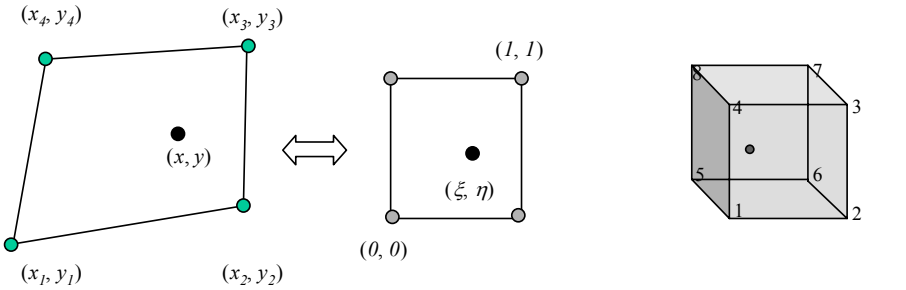
B. Identification of donors for overset nodes

Once the overset nodes are identified, solution results must be provided via interpolation from another grid (overlapping the overset node). (The boundary condition in an overset node is Dirichlett.) The process of identifying

donors involves a search within other grids. Then, the offset of the overset node within the donor cell is calculated. To determine this offset, an inverse transformation procedure is used, as described below.

C. The inverse transformation procedure

Given an overset (recipient) node located within a cell of a donor grid, shown in computational space in Fig. 2, the value of the corresponding offset of the overset node within the donor cell can be obtained via an iterative process. To accomplish this, functions relating the offset in computational and physical spaces are constructed.



(a) 2D Physical Space (b) 2D Computational Space (c) 3D Computational Space

Figure 2. Overset node enclosed within a donor cell in computation space

A Taylor expansion of the functions (f_1, f_2, f_3) relating (ξ, η, ζ) in computational space to (x, y, z) in physical space may be written as:

$$f_{i+1} = f_i + f_i' \Delta(\xi, \eta, \zeta)\big|_i + \ldots .$$

Using Newton-Raphson formula, we obtain

$$f_i' \Delta(\xi, \eta, \zeta)\big|_i = -f_i ,$$

or,

$$-\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}_i = \begin{bmatrix} \frac{\partial f_1}{\partial \xi} & \frac{\partial f_1}{\partial \eta} & \frac{\partial f_1}{\partial \zeta} \\ \frac{\partial f_2}{\partial \xi} & \frac{\partial f_2}{\partial \eta} & \frac{\partial f_2}{\partial \zeta} \\ \frac{\partial f_3}{\partial \xi} & \frac{\partial f_3}{\partial \eta} & \frac{\partial f_3}{\partial \zeta} \end{bmatrix}_i \begin{bmatrix} \Delta\xi \\ \Delta\eta \\ \Delta\zeta \end{bmatrix}_i ,$$

and

$$\begin{aligned} \Delta\xi &= \xi_{i+1} - \xi_i , \\ \Delta\eta &= \eta_{i+1} - \eta_i , \\ \Delta\zeta &= \zeta_{i+1} - \zeta_i . \end{aligned}$$

Solving for ($\Delta\xi$, $\Delta\eta$, $\Delta\zeta$) via a matrix inversion, we obtain the following iterative procedure:

$$\xi_{i+1} = \xi_i - J\left[\left(\frac{\partial f_2}{\partial \eta}\frac{\partial f_3}{\partial \zeta} - \frac{\partial f_2}{\partial \zeta}\frac{\partial f_3}{\partial \eta}\right) f_1 - \left(\frac{\partial f_1}{\partial \eta}\frac{\partial f_3}{\partial \zeta} - \frac{\partial f_1}{\partial \zeta}\frac{\partial f_3}{\partial \eta}\right) f_2 + \left(\frac{\partial f_1}{\partial \eta}\frac{\partial f_2}{\partial \zeta} - \frac{\partial f_1}{\partial \zeta}\frac{\partial f_2}{\partial \eta}\right) f_3\right],$$

$$\eta_{i+1} = \eta_i - J\left[\left(\frac{\partial f_1}{\partial \xi}\frac{\partial f_3}{\partial \zeta} - \frac{\partial f_1}{\partial \zeta}\frac{\partial f_3}{\partial \xi}\right) f_2 - \left(\frac{\partial f_2}{\partial \xi}\frac{\partial f_3}{\partial \zeta} - \frac{\partial f_2}{\partial \zeta}\frac{\partial f_3}{\partial \xi}\right) f_1 - \left(\frac{\partial f_1}{\partial \xi}\frac{\partial f_2}{\partial \zeta} - \frac{\partial f_1}{\partial \zeta}\frac{\partial f_2}{\partial \xi}\right) f_3\right],$$

$$\zeta_{i+1} = \zeta_i - J\left[\left(\frac{\partial f_2}{\partial \xi}\frac{\partial f_3}{\partial \eta} - \frac{\partial f_2}{\partial \eta}\frac{\partial f_3}{\partial \xi}\right) f_1 - \left(\frac{\partial f_1}{\partial \xi}\frac{\partial f_3}{\partial \eta} - \frac{\partial f_1}{\partial \eta}\frac{\partial f_3}{\partial \xi}\right) f_2 + \left(\frac{\partial f_1}{\partial \xi}\frac{\partial f_2}{\partial \eta} - \frac{\partial f_1}{\partial \eta}\frac{\partial f_2}{\partial \xi}\right) f_3\right],$$

where

$$J = \frac{1}{\frac{\partial f_1}{\partial \xi}\frac{\partial f_2}{\partial \eta}\frac{\partial f_3}{\partial \zeta} - \frac{\partial f_1}{\partial \xi}\frac{\partial f_3}{\partial \eta}\frac{\partial f_2}{\partial \zeta} - \frac{\partial f_1}{\partial \eta}\frac{\partial f_2}{\partial \xi}\frac{\partial f_3}{\partial \zeta} - \frac{\partial f_3}{\partial \xi}\frac{\partial f_2}{\partial \eta}\frac{\partial f_1}{\partial \zeta} + \frac{\partial f_2}{\partial \xi}\frac{\partial f_3}{\partial \eta}\frac{\partial f_1}{\partial \zeta} + \frac{\partial f_3}{\partial \xi}\frac{\partial f_1}{\partial \eta}\frac{\partial f_2}{\partial \zeta}} .$$

The functions (f_1, f_2, f_3) in (ξ, η, ζ) may be chosen as the errors associated with the interpolation of x, y, and z:

$$f_1(\xi,\eta,\zeta) = \sum x_i N_i - x,$$
$$f_2(\xi,\eta,\zeta) = \sum y_i N_i - y,$$
$$f_3(\xi,\eta,\zeta) = \sum z_i N_i - z,$$

where $N_i = N_i(\xi,\eta,\zeta)$ are the basis functions for a 64 node finite element.

Note that (x, y, z) are the physical coordinates whose (ξ, η, ζ) offsets are being determined within an enclosing donor cell.

Guess values for (ξ, η, ζ) are obtained from the metrices of the transformed domain as follows:

$$\xi = \xi_x \Delta x + \xi_y \Delta y + \xi_z \Delta z,$$
$$\eta = \eta_x \Delta x + \eta_y \Delta y + \eta_z \Delta z,$$
$$\zeta = \zeta_x \Delta x + \zeta_y \Delta y + \zeta_z \Delta z.$$

Overset nodes for which a donor is not found are referred to as orphan nodes.

1. Projection of Surface Grids

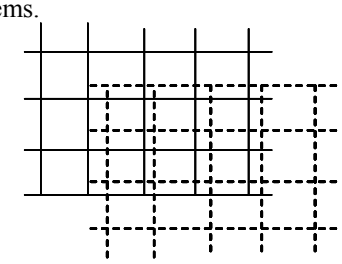
Surface mismatch of overset nodes and donor cells sometimes occur close to wall boundaries due to the error in the linear approximation of surfaces by grids and the high aspect ratio of grids close to the wall. A virtual projection of the surface is carried out to ensure the accurate pairing of overset to donor cells.[8,9]

2. Octree Structure for High-Speed Donor Searches

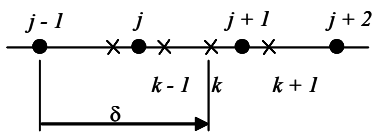
To speed up the donor search process, every grid point in all blocks are initially tagged or mapped to octrees. For this purpose, a balanced digital tree formulation was used. This allows for a fast recursive procedure for allocating points to tree segments which typically requires less than 1% of the computational time for an initial donor/overset matching process. However, the benefits result in significant decrease in the search for a donor for an overset node as only a few cells, within the same tree segment (and possibly neighboring segments), need to be assessed as potential donors.

D. Interpolation and donation of solution values to the overset nodes

After the donor cells have been identified (Section 2.B), the value of the corner node of the donor cell and the offset of the overset node within the cells are stored. During the solution process, when boundary conditions are applied, interpolation is done using a high-order stencil based on the corner node of the donor cell. Although this task needs to be performed as part of the solver process, Tasks 1 and 2 of the overset procedure may be completed as a pre-processing step, except for moving boundary problems.



(a) 2D Overlap



(b)1-D Overlap

Figure 3. Interpolation stencil for an overset node

Consider a non-coincident overlap of two grids illustrated in Fig. 3. The task is to determine the values of solution variables at a nodal point of a receptor grid using the value of known nodal points of a donor grid. The one-dimensional

case is shown for simplicity in Fig. 3b. For the receptor or overset node, *k*, a third-order interpolation utilizing a central stencil would use the nodes *j-1*, *j*, *j+1*, and *j+2* of the donor grid. The equation for the Lagrange interpolation procedure within a unit-spaced grid[12] is

$$\phi = \sum_{k=0}^{m-1} \left(\prod_{\substack{i=0 \\ i \neq k}}^{m-1} (\delta - i) \right) R(k) \phi_k ,$$

where

$$R(k) = \frac{-1^{m+k-1}}{(m-k-1)!\,k!} .$$

Here, ϕ is the solution variable being approximated, m is the pre-determined order of the interpolant and δ is the distance of the interpolated point from the left-most point (*j-1* in Fig. 3b) of the stencil. In three-dimensions, a tensor product of one-dimensional Lagrange interpolation is used.

III. Dynamic Overset Formulation

The dynamic overset procedure is developed with the objective of reducing the time for overset computation at every time step following the pre-processing step. This is accomplished by using the previous overset configuration at every time step similar to dynamic low-order methods described in other implementations.[20,21] In fact, for dynamic calculations, the CFL restriction in time typically restricts the movement of grids such that overset nodes do not usually move more than a cell away from a previous donor cell at any time step. However, for robustness, we have developed the current procedure to accommodate movement beyond one cell by implementing a neighborhood delta formulation.

A. Dynamic Donor Identification

The algorithm for the dynamic donor identification process is presented in Fig. 4. This process is executed every time step in the transient calculation following the pre-processing step.

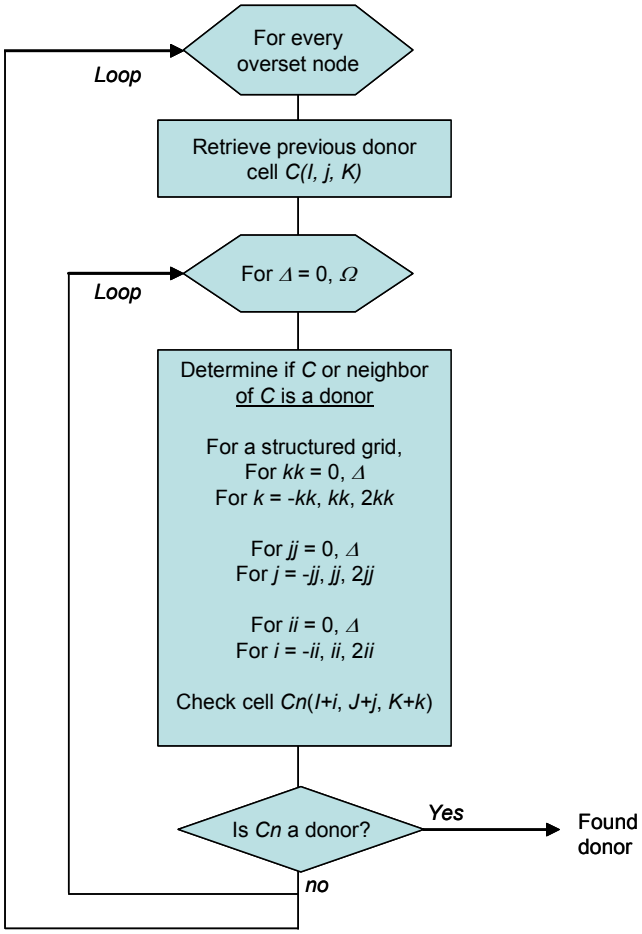


Figure 4. Algorithm for Dynamic Donor Determination Process

Note that in the algorithm in Fig. 4., Ω is an integer (usually 1) that determines the maximum number of cells from a current donor cell that an overset node may move at any time step. The value of Ω determines the speed as well as robustness of the procedure.

B. Dynamic Hole-Cut Procedure

For the high-order static overset implementation,[8,9] blank nodes may be specified via cookie-cutters or by an automatic hole-cut procedure. Dynamic versions of both methods were implemented. In particular, blanking using cookie-cutters is extremely fast. Consequently, procedures were added to the static overset method to allow cookie cutters to move similar to the grid blocks or moving bodies. The algorithms for both dynamic cookie-cutter and automatic cut procedures are presented below.

Dynamic Cookie-Cutter Method

For a moving cookie cutter D_{cookie} in 3D space, the grid point of all blocks indicated to be blanked by the cutter is assessed and blanked as indicated below.

If $\vec{x}(i,j,k) \in D_{cookie}$

Blank node $\vec{x}(i,j,k)$

Dynamic Automatic Hole-Cut Method

In Fig. 5a, the dynamic node blanking process is illustrated using two grid blocks. In this simplified illustration, the inner grid (Grid B), consisting of moving bodies causes a blanking of nodes of the grid labeled "Grid A". Only four nodes and five cells of Grid A are shown for clarity. Figure 5b shows the two grid blocks after a time step and movement of Grid B. In this figure, two nodes have changed their blanking status; a node that was formally blank is no longer a blank node and one that was formally a field (non-blank) node is now a blank node.

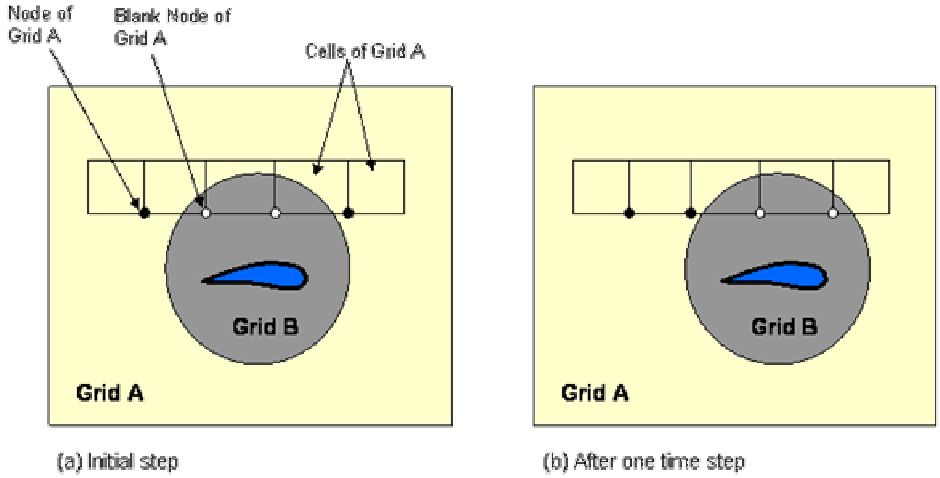


Figure 5. Movement of Blank Nodes

The algorithm to dynamically determine the blanking status of the nodes of a grid block is presented in Fig. 6.

In the implementation, nodes with field nodes within the Ω neighborhood are automatically stacked in an array at every time step.

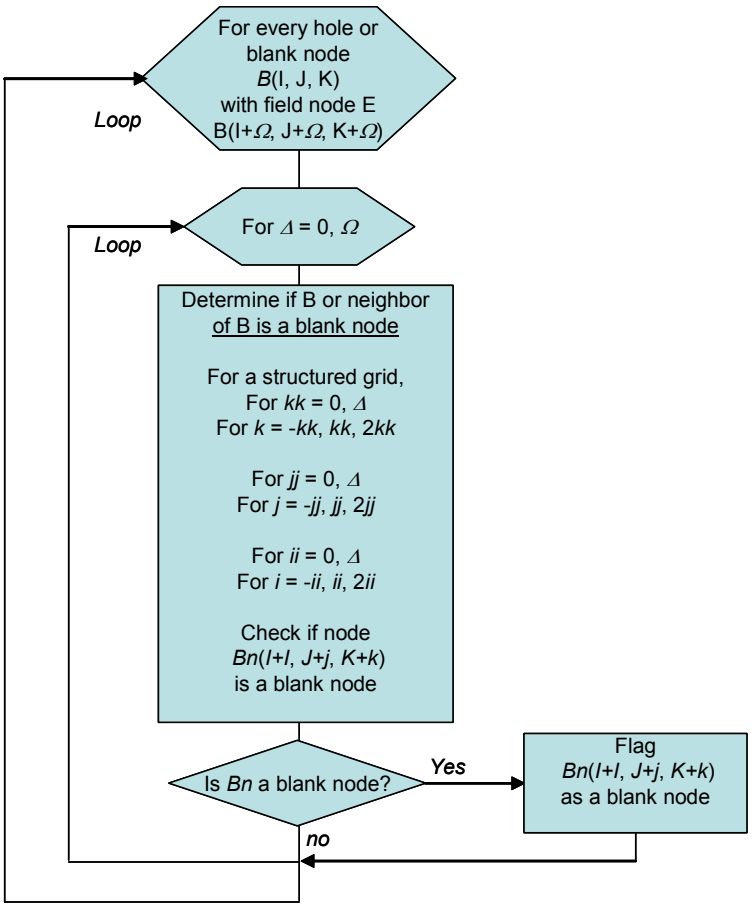


Figure 6. Algorithm for Dynamic Blanking Determination Process

IV. The CFD Procedure

The CFD equations in the high-order solver are those of the compressible Navier-Stokes equations.

$$\frac{\partial Q}{\partial t}+\frac{\partial (F-F_v)}{\partial \xi}+\frac{\partial (G-G_v)}{\partial \eta}+\frac{\partial (H-H_v)}{\partial \zeta}=0, \tag{9}$$

where Q is the vector of solution variables, and F, G, and H are the Euler fluxes:

$$Q=\frac{1}{J}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{bmatrix},\quad F=\frac{1}{J}\begin{bmatrix}\rho U\\ \rho uU+\xi_x p\\ \rho vU+\xi_y p\\ \rho wU+\xi_z p\\ (E+p)U\end{bmatrix},\quad G=\frac{1}{J}\begin{bmatrix}\rho V\\ \rho uV+\eta_x p\\ \rho vV+\eta_y p\\ \rho wV+\eta_z p\\ (E+p)V\end{bmatrix},\quad H=\frac{1}{J}\begin{bmatrix}\rho W\\ \rho uW+\zeta_x p\\ \rho vW+\zeta_y p\\ \rho wW+\zeta_z p\\ (E+p)W\end{bmatrix},$$

and (U, V, W) are the contravariant velocity components defined as

$$U=\xi_x u+\xi_y v+\xi_z w,\qquad V=\eta_x u+\eta_y v+\eta_z w,\qquad W=\zeta_x u+\zeta_y v+\zeta_z w\,.$$

In the above equations, (u,v,w) are the velocity components in the Cartesian coordinate directions (x,y,z), ρ is the density, and p is the pressure. E is the total energy, which can be written as

$$E=\frac{P}{(\gamma-1)}+\rho\frac{u^2+v^2+w^2}{2}\,.$$

The viscous terms in Eq. 1 are

$$F_v = \frac{1}{J}\begin{bmatrix} 0 \\ \tau_{xx}\xi_x + \tau_{yx}\xi_y + \tau_{zx}\xi_z \\ \tau_{xy}\xi_x + \tau_{yy}\xi_y + \tau_{zy}\xi_z \\ \tau_{xz}\xi_x + \tau_{yz}\xi_y + \tau_{zz}\xi_z \\ \left(u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x\right)\xi_x + \left(u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y\right)\xi_y + \left(u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z\right)\xi_z \end{bmatrix},$$

$$G_v = \frac{1}{J}\begin{bmatrix} 0 \\ \tau_{xx}\eta_x + \tau_{yx}\eta_y + \tau_{zx}\eta_z \\ \tau_{xy}\eta_x + \tau_{yy}\eta_y + \tau_{zy}\eta_z \\ \tau_{xz}\eta_x + \tau_{yz}\eta_y + \tau_{zz}\eta_z \\ \left(u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x\right)\eta_x + \left(u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y\right)\eta_y + \left(u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z\right)\eta_z \end{bmatrix},$$

$$H_v = \frac{1}{J}\begin{bmatrix} 0 \\ \tau_{xx}\zeta_x + \tau_{yx}\zeta_y + \tau_{zx}\zeta_z \\ \tau_{xy}\zeta_x + \tau_{yy}\zeta_y + \tau_{zy}\zeta_z \\ \tau_{xz}\zeta_x + \tau_{yz}\zeta_y + \tau_{zz}\zeta_z \\ \left(u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x\right)\zeta_x + \left(u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + q_y\right)\zeta_y + \left(u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + q_z\right)\zeta_z \end{bmatrix},$$

where

$$\tau_{ij} = \frac{1}{\mathrm{Re}}\left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right], \qquad q_i = \frac{1}{(\gamma-1)M^2\,\mathrm{Re}\,\mathrm{Pr}}\frac{\partial T}{\partial x_i}.$$

By using the implicit, approximately-factored finite-difference algorithm of Beam-Warming and employing a Newton-like sub-iteration, we have the following algorithm:

$$\left[J^{-1^{p+1}} + \phi^i \Delta t_s \delta_\xi\left(\frac{\partial \hat{F}^p}{\partial U} - \frac{1}{\mathrm{Re}}\frac{\partial \hat{F}_v^p}{\partial U}\right)\right] J^{p+1} \times \left[J^{-1^{p+1}} + \phi^i \Delta t_s \delta_\eta\left(\frac{\partial \hat{G}^p}{\partial U} - \frac{1}{\mathrm{Re}}\frac{\partial \hat{G}_v^p}{\partial U}\right)\right] J^{p+1} \times$$
$$\left[J^{-1^{p+1}} + \phi^i \Delta t_s \delta_\zeta\left(\frac{\partial \hat{H}^p}{\partial U} - \frac{1}{\mathrm{Re}}\frac{\partial \hat{H}_v^p}{\partial U}\right)\right]\Delta U$$
$$= -\phi^i \Delta t_s\left[J^{-1^{p+1}}\frac{(1+\phi)U^p - (1+2\phi)U^n + \phi U^{n-1}}{\Delta t} - U^p\left(\left(\frac{\xi_t}{J}\right)_\xi + \left(\frac{\eta_t}{J}\right)_\eta + \left(\frac{\zeta_t}{J}\right)_\zeta\right)\right]$$
$$-\phi^i \Delta t_s\left[\delta_\xi\left(\hat{F}^p - \frac{1}{\mathrm{Re}}\hat{F}_v^p\right) + \delta_\eta\left(\hat{G}^p - \frac{1}{\mathrm{Re}}\hat{G}_v^p\right) + \delta_\zeta\left(\hat{H}^p - \frac{1}{\mathrm{Re}}\hat{H}_v^p\right)\right],$$

where

$$\phi^i = \frac{1}{1+\phi}, \Delta U = U^{p+1} - U^p.$$

and superscripts "p" and "n" denote the sub-iteration steps and the outer-loop time steps, respectively. In the above equations, (ξ, η, ς) are the curvilinear coordinate directions and Δt_s is the time step for the sub-iterations. Either a first or second-order temporal accuracy can be specified in the above iterative procedure by selecting $\phi = 0$ or $\phi = 1/2$. For $p = 1$, $U^p = U^n$ and $U^{n+1} = U^p$ at convergence in p.

Note that the default spatial discretization procedure in AEROFLO is based on high-order differencing, while the standard, simple (low-order) schemes are available as options. For the former and assuming subsonic flows, we use the compact Padé approximant method for spatial differencing.[22] Consider the differencing of a flow variable u along the ξ direction (that is, $\partial u/\partial \xi$). The problem is to utilize the known u_i values to estimate the derivative $u' \equiv \partial u/\partial \xi\big|_i$ at each mesh point. In the interior, a centered formula is employed:

$$\alpha u'_{i-1} + u'_i + \alpha u'_{i+1} = b\frac{u_{i+2} - u_{i-2}}{4\Delta\xi} + a\frac{u_{i+1} - u_{i-1}}{2\Delta\xi}$$

$$(\alpha, \mathrm{a}, \mathrm{b}) = \begin{cases} \left(\frac{1}{4}, \frac{3}{2}, 0\right) : 3-pt, 4th-order \\ \left(\frac{1}{3}, \frac{14}{9}, \frac{1}{9}\right) : 4-pt, 6th-order. \end{cases}$$

For high-order differencing of flow fields with shock waves, the weighted essentially non-oscillatory (WENO) procedure[23] is used, which can be summarized as follows, if we consider the ξ direction in Eq. (9) as an example:

$$\left.\frac{\partial \hat{F}}{\partial \xi}\right|_i = \frac{1}{\Delta\xi}\left\{\left[\tilde{R}_{Roe}\cdot\left(\tilde{R}^{-1}_{Roe}\cdot\hat{F}\right)\right]_{i+1/2} - \left[\tilde{R}_{Roe}\cdot\left(\tilde{R}^{-1}_{Roe}\cdot\hat{F}\right)\right]_{i-1/2}\right\},$$

where $\hat{F}_{i\pm 1/2} = \frac{1}{2}\left[\hat{F} \pm \alpha\cdot\hat{U}\right]_{i+1/2}, \quad \alpha = \max_u\left|\hat{F}'(u)\right|,$

and $\hat{F}^{+}_{i+1/2} = \sum_{r=0}^{k-1}\omega_r\hat{F}^{(r)+}_{i+1/2}, \quad \hat{F}^{-}_{i+1/2} = \sum_{r=0}^{k-1}\tilde{\omega}_r\hat{F}^{(r)-}_{c,i+1/2}, \quad \hat{F}^{(r)}_{i+1/2} = \sum_{m=0}^{k-1}c_{rm}\hat{F}_{i-r+m},$

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1}\alpha_s}, \alpha_\gamma = \frac{d_r}{(\varepsilon+\beta_r)}, \quad r = 0,\ldots,k-1,$$

with ω_r as the normalized weights, $\overline{\omega}_r = \overline{\omega}_r(\omega_r)$ and d_γ are constants, ε is a robustness factor that prevents the occurrence of a zero denominator, while β_γ is a smoothness indicator, which is related to the undivided difference. The positive sign indicates upwind, while negative implies downwind. We set $\varepsilon \approx 10^{-14}$ and use spectral radius from coupled equation systems to compute the value of α, as opposed to a component-wise procedure to determine this parameter. Note that c_{rm} are coefficients of Lagrange interpolation formula.[8]

IV. High Order Dynamic Boundary Conditions

The boundary conditions in AEROFLO are implemented to the same high-order as the spatial schemes.[19,22] The dynamic boundary conditions are derived from the static ones except for modifications to moving wall boundary conditions including no-slip and slip conditions. For a no-slip or wall boundary, the boundary itself has a finite velocity value and setting the velocities to zero is not accurate. Generalized equations for the boundaries for slip and no-slip wall are presented below:

A. No slip boundary condition

To maintain the no-slip boundary conditions, the velocity of the fluid at a moving boundary must match the velocity of the boundary. For a boundary which has the surface equation

$$Ax + By + Cx + D = 0, \tag{10}$$

the boundary conditions for the variables, u, v, and w, will be given by $u = \dot{x}, v = \dot{y}, w = \dot{z}$. $\dot{x}, \dot{y}, \dot{z}$ are either specified or calculated from the dynamic solution between the grids at two time steps.

The pressure is derived from the forces on the boundary due to this movement

$$\frac{\partial P}{\partial n} = \frac{\int \rho a dS}{\int n\cdot dS},$$

where a is the acceleration of the surface. For ξ = constant, η = constant, ζ = constant surfaces, this equation becomes:

$$\frac{\partial P}{\partial n} = \frac{\rho J}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}\left[A\ddot{x} + B\ddot{y} + C\ddot{z}\right]$$
$$\frac{\partial P}{\partial n} = \frac{\rho J}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}}\left[A\ddot{x} + B\ddot{y} + C\ddot{z}\right]$$
$$\frac{\partial P}{\partial n} = \frac{\rho J}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}}\left[A\ddot{x} + B\ddot{y} + C\ddot{z}\right] \tag{11}$$

$\ddot{x}, \ddot{y}, \ddot{z}$ are either specified or calculated from the dynamic solution between the grids at two time steps. The density is calculated similar to the static case in 4.2.

B. Slip boundary condition

The form of the density and pressure are the same as above. However, the slip velocities can be calculated by one of the following equations (derived in the same manner as for the static equivalents):

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \begin{bmatrix} A & -\frac{AB}{C} & -\frac{B^2 + C^2}{C} \\ B & \frac{A^2 + C^2}{C} & \frac{AB}{C} \\ C & -B & A \end{bmatrix} \begin{bmatrix} \dot{N} \\ v_{i\pm1}C - w_{i\pm1}B \\ -u_{i\pm1}C + w_{i\pm1}A \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \\ w_{i+1} \end{bmatrix} = \begin{bmatrix} A & \frac{AC}{B} & \frac{B^2 + C^2}{B} \\ B & C & -A \\ C & -\frac{A^2 + B^2}{B} & -\frac{AC}{B} \end{bmatrix} \begin{bmatrix} \dot{N} \\ v_i C - w_i B \\ u_i B - v_i A \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \\ w_{i+1} \end{bmatrix} = \begin{bmatrix} A & -C & B \\ B & -\frac{BC}{A} & -\frac{A^2 + C^2}{A} \\ C & \frac{A^2 + B^2}{A} & \frac{BC}{A} \end{bmatrix} \begin{bmatrix} \dot{N} \\ -u_i C + w_i A \\ u_i B - v_i A \end{bmatrix}, \tag{14}$$

where

$$\dot{N} = A\dot{x} + B\dot{y} + C\dot{z}.$$

Only one of the three equations is needed. For boundaries aligned with the z-plane, for instance, the only non-zero value of A, B, and C, is C and Eq. (12) must be used.

To maintain high-order accuracy, at least two layers of nodes at the boundaries are set to the boundary condition values calculated according to equations. The time dependent metric terms were computed using second order for the high-order calculations and first-order for the low-order calculations as follows. Note that the high-order formulation requires the storage of an additional set of coordinates for the moving grid in time.

$$\frac{\partial x_i}{\partial t} = \frac{1}{2\Delta t}(3x_i^l - 4x_i^{l-1} + x_i^{l-2})$$
$$\frac{\partial x_i}{\partial t} = \frac{1}{\Delta t}(x_i^l - x_i^{l-1})$$

V. Results

The high-order dynamic overset procedure was used to simulate a pitching airfoil. Results were compared to low-order methods. In addition, the procedure was also used to calculate the moving flap in an airfoil/flap configuration. Finally, a 3D calculation of a store separating from a 3D wing was commenced. The static calculations are presented in the current abstract.

A. Sinusoidally Pitching Airfoil

The airfoil has a NACA0012 profile and the sinusoidal pitch was prescribed by the equation

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t)$$

where ω is related to the reduced frequency κ by

$$\kappa = \frac{\omega c}{2U_\infty}.$$

In non-dimensional form, this can be written as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(2M_\infty \kappa t)$$

where the non-dimensional symbol used to represent time, t, has simply replaced the dimensional one.

The calculations were completed for the case CT1 in the AGARD Report: Compendium of unsteady aerodynamic measurements.[23,24] with the following parameters:

$M_\infty = 0.6$
$\alpha_m = 2.89^o$
$\alpha_0 = 2.41^o$
$\kappa = 0.0808$

The grid used for the airfoil contains 245 x 64 nodes. For the low-order calculations, the MUSCL scheme was used for spatial differencing while the Beam-Warming procedure for time integration. For the high order calculations, the WENO scheme was used for spatial differencing while the 4th-order Runge-Kutta procedure for time integration. A time step size of 0.02 was chosen with seven sub-iterations each sub-iteration with a size of 0.0005. The results are presented in Fig. 7 after about 6 cycles. These results include the lift and moment coefficients and show high-order calculations that are in closer agreement with experimental values.

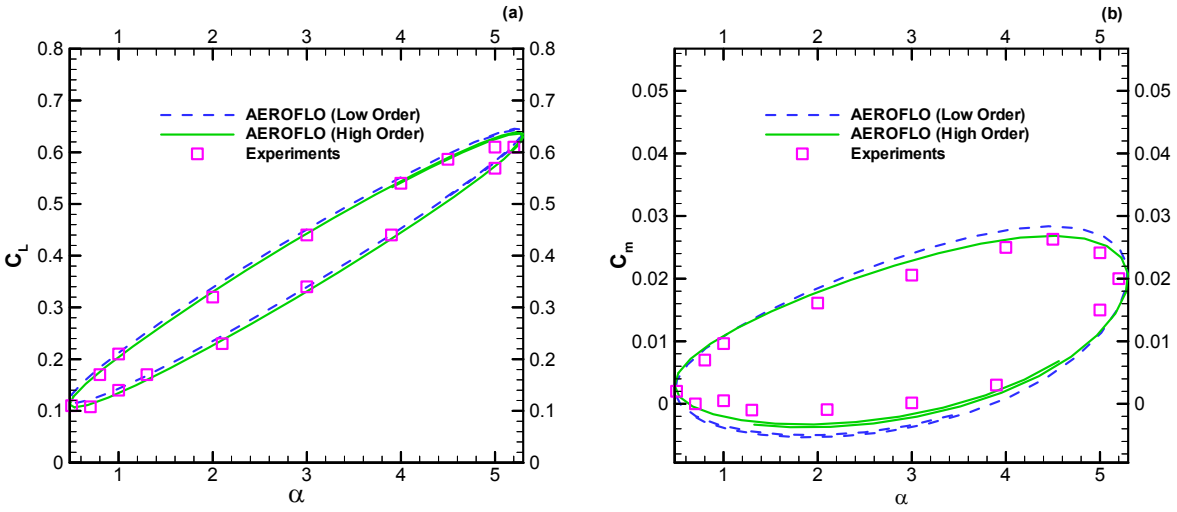


Figure 7. Low and high order results of C_L and C_M for sinuisoidally pitching airfoil calculations compared to experiments.

B. Moving Flap of an Airfoil/Flap Configuration

Calculations were done for flow past an airfoil/flap combination. The physical domain is illustrated in Fig. 8. The dimensions in the figure are in inches. The airfoil has a NLR(1)-0015 profile while the flap has a NACA 64(1)-112 profile. The model is two-dimensional. Static calculations were done at Mach number, $M_\infty = 0.25$, wing angle of attack, $\alpha_w = 0^o$, and three flap angles, $\alpha_f = 0^o$, 10^o, and 20^o. Dynamic calculations were based on a flap movement from 0^o to 20^o. For the low-order calculations, the MUSCL scheme was used for spatial differencing while the Beam-Warming procedure for time integration. For the high order and static calculations, the compact scheme was used for spatial differencing while the 4^{th}-order Runge-Kutta procedure for time integration. The calculations are based on the Euler equations. The airfoil chord length, C, was used as the length scale.

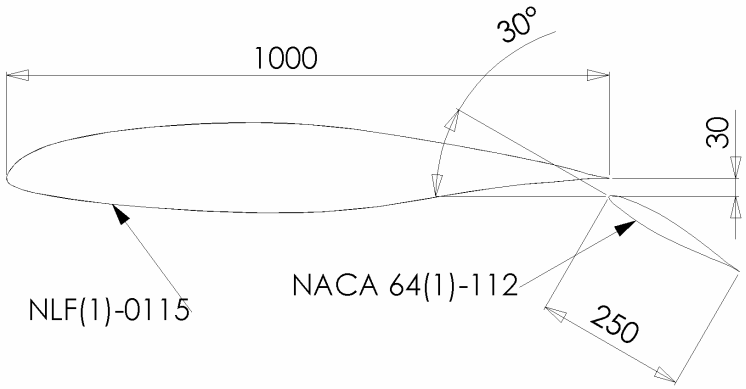


Figure 8. Airfoil/Flap Configuration.

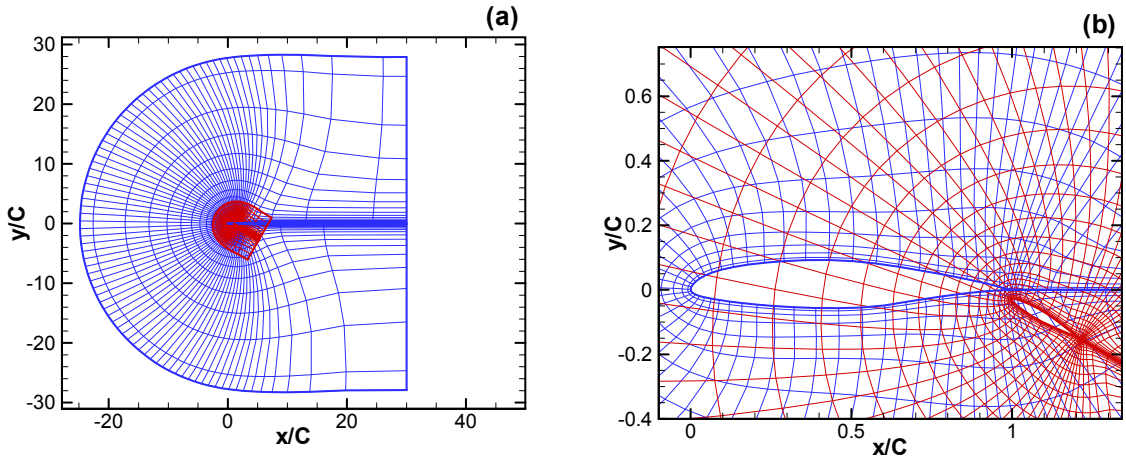


Figure 9. Grids at 30^o flap angles after rigid transformations (a) Airfoil and flap grid after scaling and rotation, (b) Close up view.

The airfoil and flap grids were generated separately and are shown in Fig. 9. The grids were assembled as a pre-processing step transforming appropriately for the different flap angles and position. Following the transformation, hole cuts were made between both grids using the overset procedures and calculations were performed to obtain the results for the full domain (airfoil and flap). The grids following the overset assembly is shown in Fig. 10. 251 x 68 grid points were used in the airfoil grid and 121 x 53 gird points were used for the flap grid. A time step size of 0.05 was used with five sub-iterations of 0.0005. The total motion was imposed over a total non-dimensional time of 500.

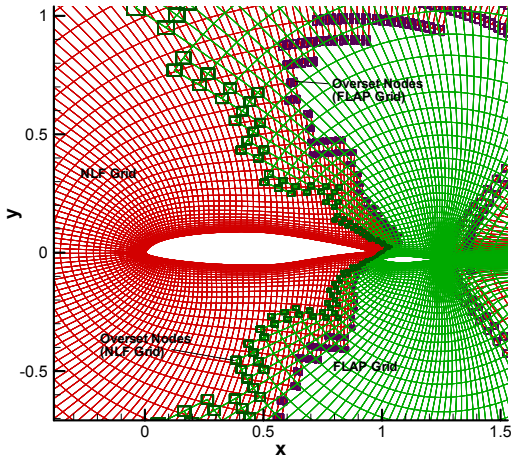


Figure 10. Overset grids following hole cut. Notice the number of fringe nodes and between the grids required to perform high-order interpolation between the grids.

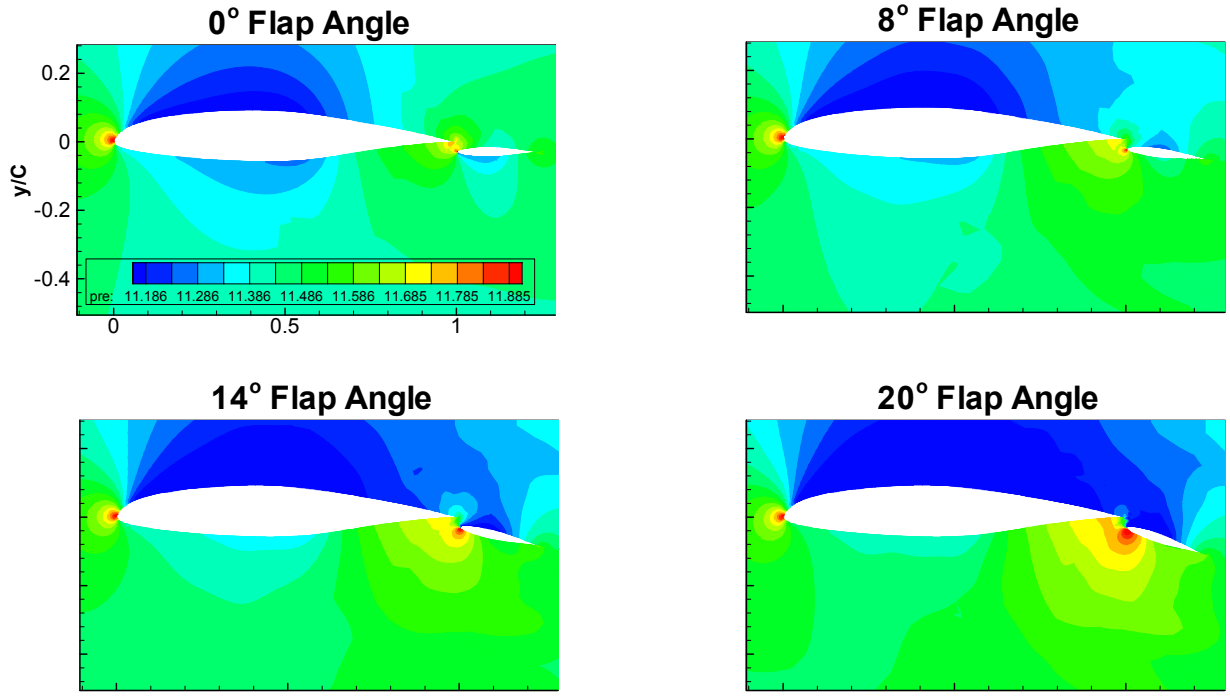


Figure 11. Snapshots of $P/\rho U^2$ contours at different times in the moving flap calculations.

Snapshots of the high-order solutions at selected flap angles are presented in Fig. 11. The results of the dynamic and static calculations are shown in Fig. 12 for the lift and drag coefficients. The high-order calculations appear to provide results that are closer to the static one. Additional analysis is in progress.

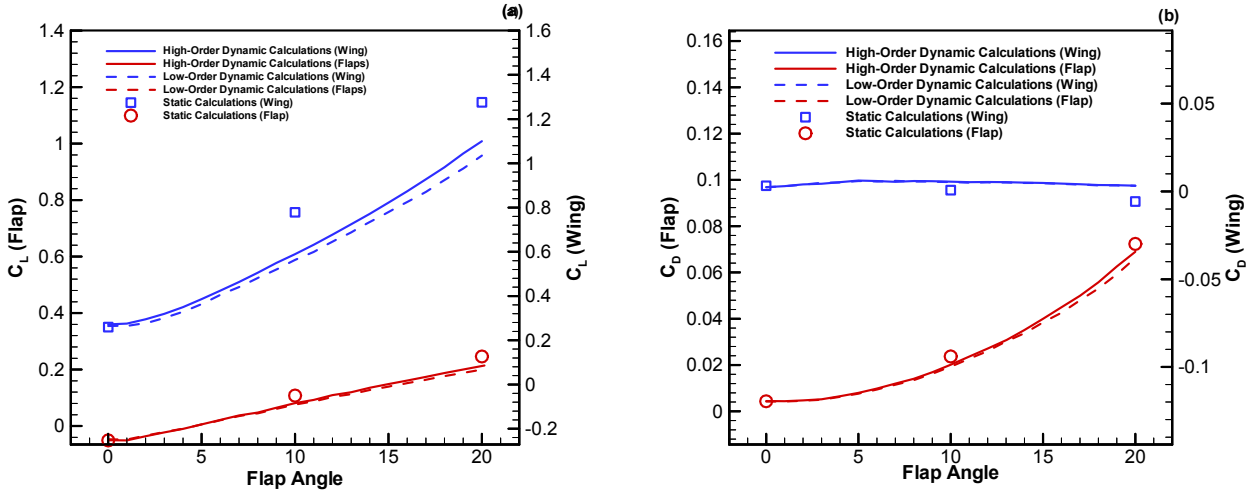


Figure 12. Comparisons of high and low-order moving flap calculations

C. Store Separation from 3D Wing

Static calculations were completed for flow past a wing store combination. The physical domain is illustrated in figure 13. The dimensions in the figure are in inches. The wing has a NACA 0012 profile and is three-dimensional while the store is an ellipsoid.

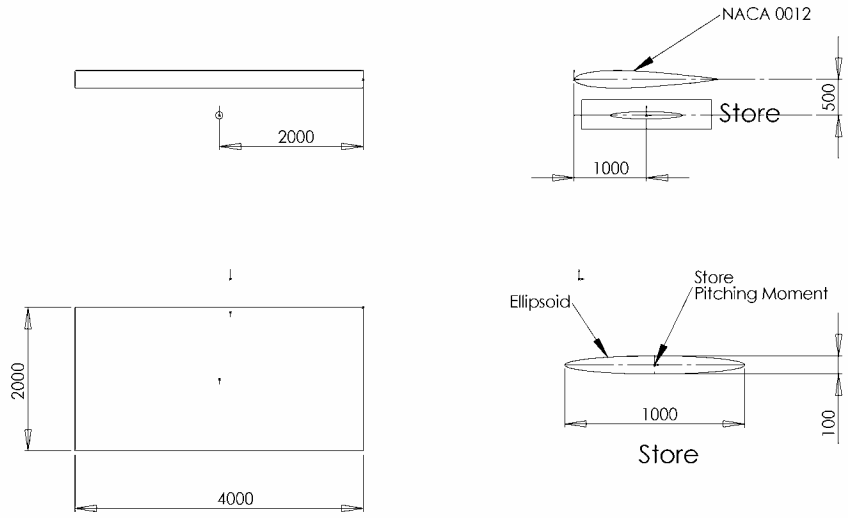
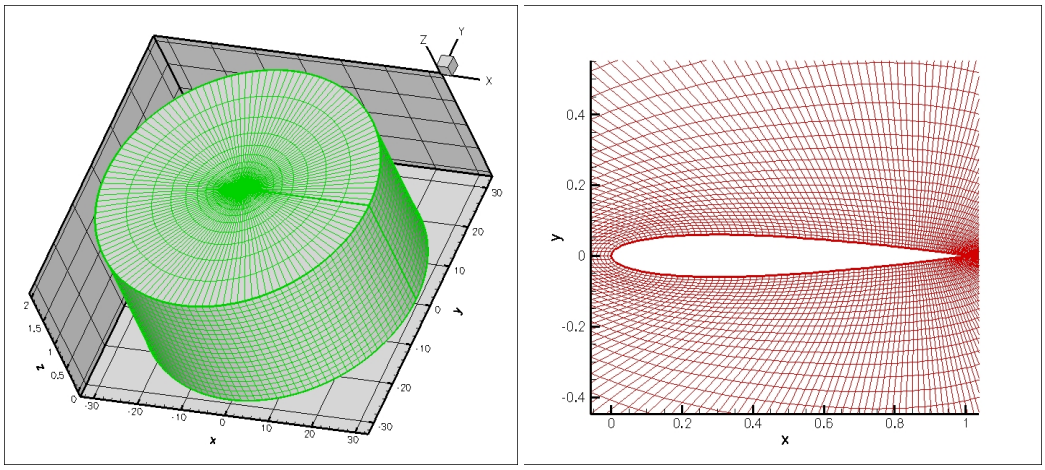


Figure 13. Wing/Store Configuration.

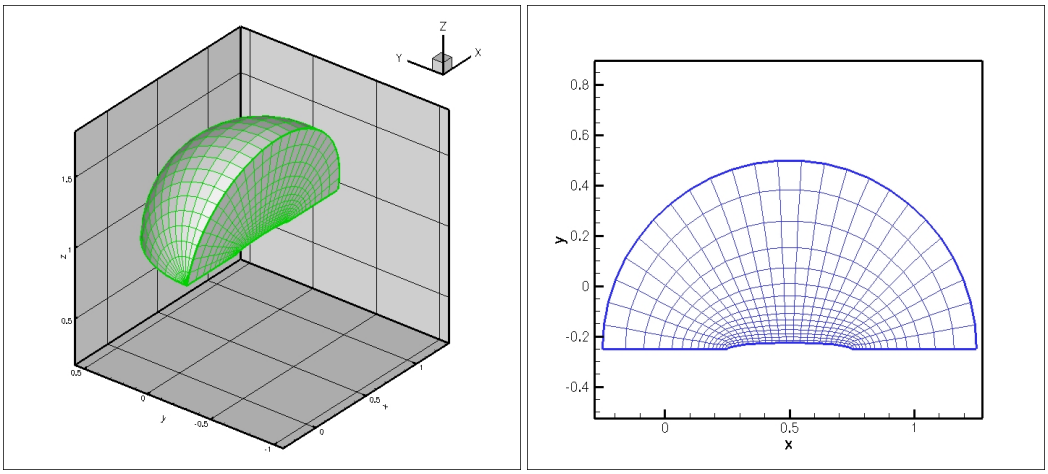
The wing grid uses an O-grid with *249* x *55* x *41* grid points. The store grid is an axisymmetric grid with *51* x *31* x *41* grid points. The grids are shown in Figs. 14 and 15. The grids were generated separately and assembled in the solver using grid transformation procedures to appropriately scale and translate them to the required relative positions. The wing chord length, *C*, is used as the length scale. The overset procedure was used for the full domain calculations to supply boundary values and convey solutions between both grids. Figure 16(a) shows the surface grids after assembly while Fig. 16(b) shows a section through the grids after the overset procedure was completed.

Calculations were completed at Mach number, $M_\infty = 0.8$, wing angle of attack, $\alpha_w = 0^o$, and three store separation distances, $s/C = 0.25$, 0.5, and 1.0. The partially blanked grids (focusing on the store grid) for the three separation distances are presented in Fig. 17. The calculations are based on the Euler equations and utilize the WENO scheme for spatial differentiation and fourth-order Runge-Kutta procedure for time integration.



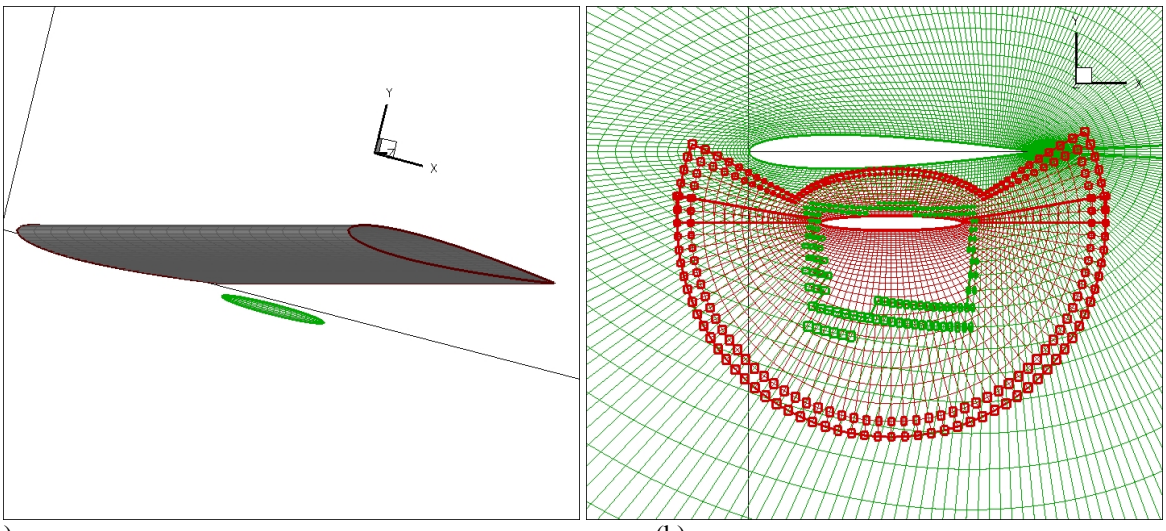
a) b)

Figure 14. Wing mesh (a) Full grid, (b) Plane section through grid at *Z = 1.0*.



a) b)

Figure 15. Store mesh with every other grid point removed (a) Section of full grid, (b) Axisymmetric section through grid.



(a) (b)

Figure 16 Combined grid (a) Surface grid of the wing and the store with every other grid point removed (b) Plane section at *z/C* = 1.0 for s/C = 0.25 showing overset nodes of both grids.

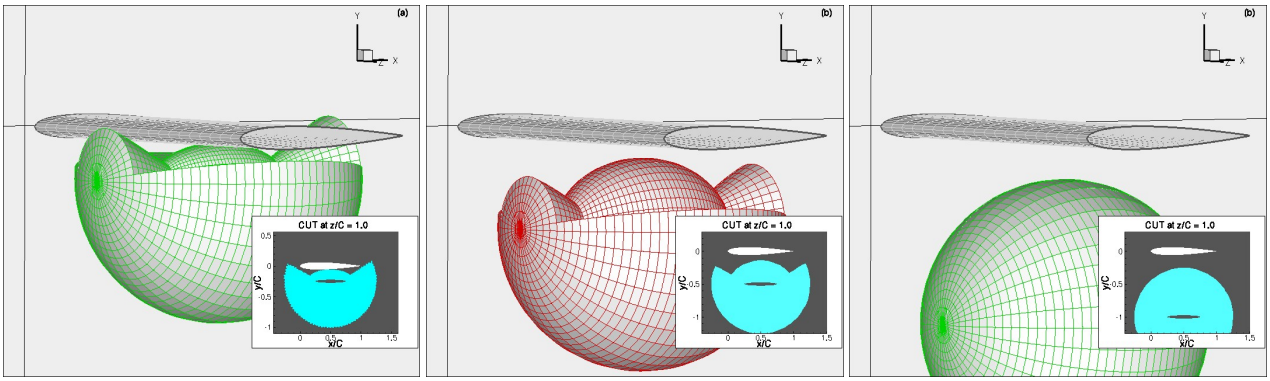


Figure 17 Overset computational domains (a) *S/C* = 0.25, (b) S/C = 0.5, (c) S/C = 1.0.

Figures 18 shows the lift coefficient, C_L, and drag coefficient, C_D, for the static calculations at the three separation distances. The results show decreasing total lift coefficient with store separation distance and a slight increase in the total drag. These results were compared to dynamic calculations of the store motion from *s/C* = 0.25 through 1.0. A time step size of 0.005 was used with five sub-iterations of 0.0004.

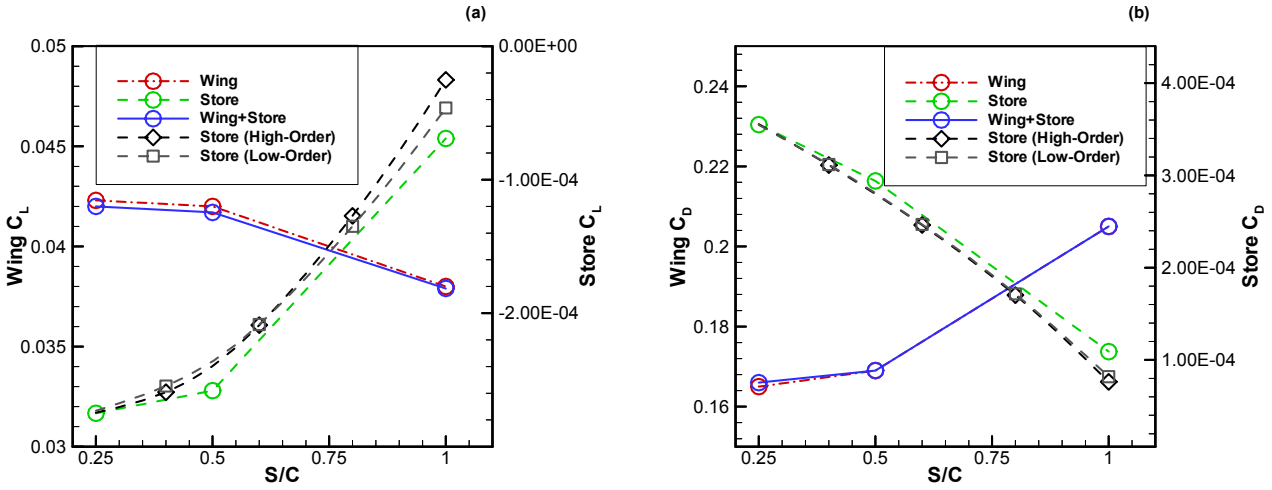


Figure 18. Drag coefficient variation with store separation distance: (a) *S/C = 0.25*, (b) *S/C = 0.5*, (c) *S/C = 1.0* for static calculations.

The results for the static calculations were compared with those of low-order and high-order dynamic calculations for the same store moving at a non-dimensional speed of 0.15 from s/C = 0.25 through 1.0. The results are also presented in Figure 18. Similarity of both results can be observed. These calculations were completed without any measure of the relative superiority of the high-order calculations. Future work is intended to focus on this aspect by carefully selecting problems for which the performance of both procedures can be clearly compared with known outcomes.

V. Conclusions

The current paper presents a procedure for high-order overset calculations for moving grids. Algorithms for extending previously developed high-order static overset formulation to a dynamic one were presented for both the donor search process and the hole cut or blanking procedure. Results comparing low and high-order dynamic overset formulations are presented to show the potential for accurate high-order overset calculations. However, the effect of the high-order formulation for time-dependent metrices can not be determined from the initial applications tested which were based on prescribed motion. Future work will focus on additional testing and analysis of the performance of high-order dynamic procedures compared to low-order ones, as well as evaluating the potential for savings in computational resources by analyzing the accuracy of high-order calculations using coarser grids and larger time step sizes.

Acknowledgments

This work was partially funded by the United States Air Force, Contract F33615-03-C-3315, via the Phase II SBIR Program, with Dr. Datta Gaitonde as Technical Monitor. The authors are very grateful to the Air Force for giving

Thaerocomp the opportunity to develop innovative research tools and to Dr. Gaitonde for the various administrative and technical supports.

References

[1]Davis, M.B., Welterlen, T.J, and Corfeld, K.J. 2003. "Minimized Domain CFD for Store Separation". AIAA Paper 2003-1245.

[2]Cenko, A., Meyer, R., and Tessitore, F. 1986. "Further Development of the Influence Function Method for Store Aerodynamic Analysis.", J. Aircraft, 23, 8.

[3]Rogers S.E., Roth, K., Nash, S.M., Baker, D.M., Slotnick J.P., Whitlock M., Cao H.V. "Advances In Overset CFD Processes Applied to Subsonic High-Lift Aircraft.", AIAA Paper 2000-4216, AIAA Applied Aerodynamics Conference, 2000, Denver Co.

[4]Snyder, R.D. and Scott, J.N., "An Overset Grid Solver with Method Coupling and Feature Tracking," AIAA Paper 99-3290, 1999.

[5]Josyula, E and Gordnier, R.E., " Computational Simulation of the F-22 Aircraft," AIAA Paper 98-0526, 36th Aerospace Sciences Meeting, Reno, NV, January 1998.

[6]Lee, Y. and Baeder, J.D., "High-Order Overset Method for Blade Vortex Interaction," AIAA Paper 02-0559, AIAA 40th Aerospace Sciences Meeting Reno, NV, January 2002.

[7]Sherer, S.E & Scott, J.N., 2002. Development and Validation of a High Order Overset Grid Flow Solver". AIAA Paper 2002-2733.

[8]Alabi, K. & Ladeinde, F., 2004. "Parallel, high-order overset grid implementation for supersonic flows". AIAA Paper 2004-1178.

[9]Alabi, K. & Ladeinde, F., 2005. "Treatment of Hole Nodes in a High-Order Overset Procedure for Supersonic Flows". AIAA Paper 2005-0437.

[10]Cenko, A., 1999. "F/A-18C/JDAM CFD Challenge Wind Tunnel and Flight Test Results," AIAA Paper 1999-0120.

[11]Cenko, A., 2000. Gowanlock, D., Lutton, M., Tutty, M., "F/A-18C/JDAM Applied Computational Fluid Dynamics Challenge II Results," AIAA Paper 2000-0795.

[12]Hall, L., 1999. "Navier-Stokes/6DOF Analysis of the JDAM Store Separation for the F/A-18C Aircraft," AIAA Paper 1999-0121.

[13]Tomaro, R., et. al., 1999. "A solution of the F -18C for Store Separation using COBALT," AIAA Paper 1999-0122.

[14]Woodson, S., Bruner, C., 1999. "Analysis of Unstructured CFD Codes for the Accurate Prediction of A/C Trajectories," AIAA Paper 1999-0123.

[15]Welterlen, T., 1999. "Store Release Simulation on the F/A-18C Using Split Flow," AIAA Paper 1999-0124.

[16]McGroy, W., et. al., 1999. "Store Trajectory Analysis About the F/A-18C Using GUST Unstructured Grid Generation and Flow Solver Package," AIAA Paper 1999-0125.

[17]Fairlie, B., Caldeira, R., 1999. "Prediction of JDAM Separation Characteristics from the F/A-18 Aircraft," AIAA Paper 1999-0126.

[18]Safta, S., Alabi, K., Ladeinde, F., 2006, "Comparative advantages of high-order schemes for subsonic, transonic, and supersonic flows ," AIAA Paper 2006-0299.

[19]Ladeinde, F., Alabi, K., Safta, S., Cai, X., Johnson, F., 2006, "The First High-Order CFD Simulation of Aircraft: Challenges and Opportunities," AIAA Paper 2006-1526.

[20]Meakin, R., Suhs, N., 1989, "Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion," AIAA Paper 89-1996-CP.

[21]On Application of Chimera Grid Schemes To Store Separation, Dougherty, F.L., Benek, J. A., & Steger, J. L., NASA Technical Memorandum 88193, Nvember 1985.

[22]Gaitonde, D. & Visbal, M., "High-Order Schemes for Navier-Stokes Equations: Algorithm and Implementation into FDL3DI," Technical Report AFRL-VA-WP-TR-1998-3060, Air Forcel Research Laboratory, Wright-Patterson AFB, 1998.

[23]Compendium of unsteady aerodynamic measurements. AGARD Report No.702. August 1982.

[24]Compendium of unsteady aerodynamic measurements. AGARD Report No.702 Addendum 1. May 1985.