A Multi-solver Scheme for Viscous Flows Using Adaptive Cartesian Grids and Meshless Grid Communication

Aaron Katz* and Antony Jameson[†]

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

Andrew M. Wissink[‡]

Ames Research Center, Moffett Field, CA

This work concerns the development of an adaptive multi-solver approach for CFD simulation of viscous flows. Curvilinear grids are used near solid bodies to capture boundary layers, and structured adaptive Cartesian grids are used away from the body to fill the majority of the computational domain. An edge-based meshless scheme is used in the interface region to connect the near-body and off-body codes. We show that the combination of a body-fitted grid near the surface coupled with an adaptive Cartesian grid system away from the surface leads to a highly efficient scheme with sharp feature resolution. The use of a meshless flow solver to interface the body-fitted and Cartesian grid systems leads to seamless grid communication without many of the complexities inherent in traditional Chimera overset grid interpolation schemes. The hierarchical structure of the nested Cartesian grids may be exploited to achieve multigrid convergence for steady problems and for use in dual-time stepping algorithms for unsteady problems. Results of two-dimensional steady airfoil calculations are presented.

I. Introduction

MOST modern CFD codes used today are written for a single gridding paradigm, such as structured-Cartesian, structured-body-fitted or unstructured (tetrahedral, hexahedral or prismatic) grids. Each meshing paradigm has specific advantages and disadvantages. For example, Cartesian grids are simple to generate, to adapt, and to extend to higher-order spatial accuracy, but they are not suited for resolving boundary layers around complex geometries. Body-fitted curvilinear grids work well for resolving boundary layers but are difficult to generate around complex geometries. Unstructured grids are well-suited for complex geometries and are easier to generate, but their spatial accuracy is usually limited (to second-order) and they tend to be less computationally efficient than their structured-grid counterparts. Thus, while a single gridding paradigm brings certain advantages in some portions of the flowfield, it also imposes an undue burden on others.

In this work we investigate a mixed-grid solution strategy that couples unstructured grids for the “near-body” region, Cartesian structured grids for the “off-body” region, and a meshless solver in between. The use of a mixed-grid paradigm is not new. The OVERFLOW code of Buning et al¹ employs structured body-fitted grids in the near-body region and multi-level Cartesian grids in the off-body. The Helios code^{2,3} uses a similar paradigm but with general unstructured grids in the near-body, and an implementation that links two separate codes (i.e. an unstructured near-body solver and an adaptive Cartesian off-body solver) through Python. Both use traditional Chimera overset grid connectivity to link the two grid systems by interpolating data in the overset region.

In earlier work,⁴ we introduced the idea of using a meshless solver in the interface region between the near- and off-body grid systems, in place of traditional overset grid interpolation in the overlap region. The

*PhD Candidate, Department of Aeronautics and Astronautics, Stanford, CA, AIAA Member.

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

[‡]Research Scientist, Scaled Numerical Physics LLC, AIAA member

advantage of a meshless solver was that we solved the governing equations at all interface nodes, rather than simply interpolating data as is done in traditional overset domain connectivity schemes. This led to better resolution of features in the overlap region. The meshless interface also avoids many of the complexities associated with hole-cutting and ensuring proper overlap in traditional overset methods. Only clouds of points needed to be identified in place of the rigid donor/recipient structure used in interpolation.

Our earlier work demonstrated the capabilities of the meshless interface approach but was limited to uniform Cartesian off-body grids. In this work we extend these capabilities to include nested Cartesian grids that adapt to flow features. We include viscous terms in all regions of the discretization. These new capabilities will eventually allow us to resolve time-dependent features, such as flow separation, automatically without a priori resolution of the mesh. From an algorithmic standpoint, the nested structure of the Cartesian grid system will allow us to perform a multigrid solution algorithm in the future, enhancing convergence for steady state problems, and for use in dual-time unsteady problems. The structured multigrid on Cartesian grids could be used with the multicloud algorithm of Katz and Jameson,⁵ which has been shown to dramatically increase convergence for meshless algorithms.

The abstract is organized as follows: section II discusses details of the discretization strategy used in the three regions of the flowfield: near-body curvilinear, off-body Cartesian, and meshless in-between. Section III discusses our strategy for adaptivity of the off-body Cartesian grids. Section IV gives the results obtained so far. Finally, Section V provides a summary of the results along with future directions of research.

II. Discretization Strategies

The paradigm of combining multiple solvers to achieve an accurate global simulation capability is truly revolutionary, with the potential to leverage limitless capabilities in a manageable form. In this work we seek to solve the Navier-Stokes equations in two dimensions to provide a proof of concepts and validation of the method. The Navier-Stokes equations may be expressed as

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

where w is the vector of conserved variables, f_e and g_e are the inviscid flux vectors, and f_v and g_v are the viscous flux vectors. As a significant demonstration of the multi-solver technology, we combine three distinct discretization strategies: (1) near-body curvilinear grids, (2) off-body adaptive Cartesian grids, and (3) meshless interface regions. As a matter of implementation, a high level computer language, such as Python, is particularly well-suited to perform grid communication tasks. An example Python script to drive a steady-state simulation is shown in Figure 1. The use of a high-level script has three main advantages. First, preexisting standalone solvers, may be imported as shared object modules. This requires minimal modification to source codes and allows the full capability of various solvers to be accessed directly at run-time. Second, since, solvers are imported as shared objects, all data is held in memory, leading to minimal I/O overhead costs for data transfer. Finally, dividing simulation tasks into distinct modules, allows for ease of modification with the ability to test different schemes with only local coding changes. Such “plug in” coding is invaluable to developers to reduce bugs, greatly expediting code development.

II.A. Near-body

Near solid bodies, body-fitted curvilinear grids are used. The use of such grids accommodates complex geometry and allows for anisotropic grid refinement to capture viscous effects. For the two-dimensional cases tested in this work, C-mesh topologies, shown in Figure 2(a), are used to capture viscous wakes accurately.

The solution method follows closely the viscous flow implementation of Jameson,⁶ which is based on a multistage, multigrid algorithm for steady-state computations. Further details of the algorithm may be found in the thesis of Nadarajah.⁷

II.B. Off-body

The regions of the domain extending beyond the near-body grid to the outer boundaries of the domain are covered by adaptive Cartesian grids, as shown in Figure 2(b). Cartesian algorithms may be made extremely

```
# import necessary modules

import nearBody
nearBody.dimension()
nearBody.initialize()

if (multisolver):
    import cartSolver
    import meshless
    cartSolver.genInitialMesh()
    cartSolver.initialize
    meshless.dimension()
    meshless.initialize()

# iterate until steady state is reached

time = 0
converged = False
while not converged:
    time = time+1
    converged = nearBody.takestep()
    if (multisolver):
        if (adapt): # adapt Cartesian solver
            cartSolver.adaptMesh()
            meshless.dimension()
            meshless.initialize()
            convergedc = cartSolver.takestep()
            convergedm = meshless.takestep()
    else:
        nearBody.farfield()

# post-process results

nearBody.plotresults()
cartSolver.plotresults()
meshless.plotresults()
```

Figure 1. Top level Python driver for hybrid meshless scheme

Figure 2. Near and Off-body grid systems

efficient and accurate thanks to their rigid structure. The scheme used on the Cartesian grids is derived from the near-body algorithm, with modifications to exploit the Cartesian structure. For example grid and metric data need not be stored at each cell. Instead, only the coordinates of the lower left and upper right corners of the Cartesian block, along with the number of cells in the I and J directions need to be stored. Similar gains in efficiency and reduction of memory may be realized elsewhere as well. Cartesian grids also permit a convenient multi-level representation that facilitates numerical algorithms like multigrid.

As can be seen in Figure 2(b), the Cartesian solver must be able to accomodate “holes” arising from the presence of the geometry and near-body grid. 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.⁸ The presence of a hole in the Cartesian mesh is indicated by an array, s , of zeros (blanked points) or ones (field points). The blanked points are not considered part of the computation since the domain occupying their space is covered by the near-body grid. The Cartesian solver has been modified to account for holes. An example of a modification is found in the implicit residual smoothing. In the absence of a hole, the residual, R , is smoothed implicitly to increase the allowable CFL number of the explicit scheme by

$$-\epsilon \bar{R}_{i-1} + (1 + 2\epsilon) \bar{R}_i - \epsilon \bar{R}_{i+1} = R_i, \quad (2)$$

where ϵ is a smoothing parameter and $\bar{R}$ is the smoothed residual. To account for the presence of a hole, the following modification to Eq.(2) is observed:

$$-s_i s_{i-1} \epsilon \bar{R}_{i-1} + (1 + 2s_i \epsilon) \bar{R}_i - s_i s_{i+1} \epsilon \bar{R}_{i+1} = R_i. \quad (3)$$

The form in Equation 3 leads to a stable tri-diagonal system which does not smooth across blanked regions. This avoids drawing information from invalid cells along hole-cut surfaces.

II.C. Meshless Regions

As previously shown by Katz and Jameson,⁴ a meshless scheme is particularly well-suited to interface grid types. Here an edge-based meshless scheme is used to interface the near-body and off-body grid components. The meshless scheme only requires a cloud of neighboring points for each point in the meshless region. The task of determining the meshless clouds may be performed in a stepwise fashion to obtain near and off-body meshless points and their necessary support regions, as shown in Figure 3. Meshless clouds make use of the grid structure of the parent grids when possible. Grid points are also found from neighboring grid components to complete local cloud definition. The details of the cloud definition procedure may be found in our previous work.⁴

The spatial discretization of the Navier-Stokes equations for the meshless scheme is based a least-squares method for obtaining derivatives on an arbitrary cloud of points.⁹ Derivatives of any property ϕ at node i with n surrounding nearest neighbors may be expressed as

(a) Overset region for NACA 0012.

(b) Step 1

(c) Step 2

(d) Step 3

(e) Step 4

Figure 3. Procedure for determining the meshless interface

$$\frac{\partial \phi}{\partial x} \approx \sum_{j=1}^n a_{ij}(\phi_j - \phi_i), \quad \frac{\partial \phi}{\partial y} \approx \sum_{j=1}^n b_{ij}(\phi_j - \phi_i), \quad (4)$$

where a_{ij} and b_{ij} are metric coefficients arising from the solution of a least squares problem. The least squares problem typically arises from Taylor series expansions or polynomial fitting procedures. In light of Equation 4, the Navier-Stokes equations of Equation 1 may be semi-discretized as

$$\frac{\partial w_i}{\partial t} + \sum_{j=1}^n (a_{ij}(f_{ej} - f_{ei}) + b_{ij}(g_{ej} - g_{ei})) = \frac{M_\infty}{Re_\infty} \sum_{j=1}^n (a_{ij}(f_{vj} - f_{vi}) + b_{ij}(g_{vj} - g_{vi})).$$

Using a directed flux, $F = af + bg$, in the direction of the least squares coefficients, the discretization becomes

$$\frac{\partial w_i}{\partial t} + \sum_{j=1}^n (F_{ej} - F_{ei}) = \frac{M_\infty}{Re_\infty} \sum_{j=1}^n (F_{vj} - F_{vi}). \quad (5)$$

Unfortunately, the dissipative nature of the viscous terms present in Equation 5 is not always 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 Jameson¹⁰ is invoked, differencing to the midpoint of each edge in the meshless cloud. Equation 5 becomes

$$\frac{\partial w_i}{\partial t} + 2 \sum_{j=1}^n (F_{eij} - F_{ei}) = \frac{2M_\infty}{Re_\infty} \sum_{j=1}^n (F_{vij} - F_{vi}). \quad (6)$$

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

$$F_{eij} = \frac{1}{2}(F_{ei} + F_{ej}) - \frac{1}{2}d_{ij} \quad (7)$$

The diffusive term may be constructed from any number of schemes including scalar and characteristic schemes. In this work Jameson's CUSP scheme¹¹ is used:

$$d_{ij} = \alpha^* c(w_R - w_L) + \beta(F_{eR} - F_{eL}). \quad (8)$$

Left and right states are formed using SLIP reconstruction.¹⁰

In addition to the Euler flux, the viscous flux must also be computed at the edge midpoint as required by Equation 6. If care is not taken, the viscous terms may support odd/even modes as well. 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 Equation 4. If the midpoint viscous fluxes are computed using an arithmetic average of these gradients, odd/even decoupling could ensue as a results of the recycling of the gradients to form the flux derivatives. A one dimensional analysis of such a procedure shows a five point stencil which excludes the even points. Instead, a modified average of the velocity and temperature gradients may be computed as:¹²

$$\nabla \phi_{ij} = \overline{\nabla \phi_{ij}} - \left(\overline{\nabla \phi_{ij}} \cdot \vec{s}_{ij} - \frac{\phi_j - \phi_i}{|\vec{r}_j - \vec{r}_i|} \right) \vec{s}_{ij}, \quad (9)$$

where $\overline{\nabla \phi_{ij}} = \frac{1}{2}(\nabla \phi_i + \nabla \phi_j)$ is the average of the gradients at i and j using the least squares method of Equation 4, 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, the gradients of which are needed to form the viscous flux.

Finally, the semi-discretized form of the Navier-Stokes equations may be expressed as

$$\frac{\partial w_i}{\partial t} + R_i = 0, \quad (10)$$

where

$$\begin{aligned}
R_i &= Q_i - D_i, \\
Q_i &= \sum_{j=1}^n (F_{ej} - F_{ei}), \quad \text{and} \\
D_i &= \sum_{j=1}^n d_{ij} + \frac{2M_\infty}{Re_\infty} \sum_{j=1}^n (F_{vij} - F_{vi}).
\end{aligned}$$

Equation (10) is advanced to a steady state with the same multistage Runge-Kutta scheme as the near and off-body meshes.

III. Adaptive Cartesian Flow Solver

Adaptive capability is provided to the Cartesian grid flow solver through a Berger and Colella¹³-style multi-level block-structured AMR (SAMR) grid hierarchy. The coarsest level defines the physical extent of the computational domain. Each finer level is formed by selecting cells on the coarser level and then clustering the marked cells together to form block regions that will constitute the new finer level. The result is a hierarchy of nested refinement levels with each level composed of a union of logically-rectangular grid regions (Figure 4). Construction and storage of this grid system is facilitated by the SAMRAI^{14–16} package from Lawrence Livermore National Lab.

Figure 4. Block structured AMR grid composed of a hierarchy of nested levels of refinement. Each level contains uniformly-spaced logically-rectangular regions, defined over a global index space.

SAMRAI manages construction and data communication on the SAMR grid hierarchy and an instance of the single block Cartesian grid flow solver is executed on each block. The solver uses the same multi-stage Runge-Kutta scheme used by the near-body solver, with explicit updates at block boundaries. Fringe cells are used to exchange data across blocks of the same refinement. Blocks of the same resolution have point matching so there is no loss in accuracy at block boundaries. At coarse-fine interfaces, second order interpolation is used. A multi-grid scheme which makes use of the nested refinement levels is not yet implemented in this adaptive version of the off-body solver. However, this is a natural extension, as shown by Kirshman and Liu,¹⁷ that will be examined in future work.

Refinement of the off-body grid system is performed by marking cells on a level and constructing a new grid level from those marked regions. This requires some refinement criteria to be defined; e.g. gradients in flow quantities, computed errors, etc. In this work, we choose to refine to gradients in density, since that indicates the presence of a shock. A general assessment of the appropriate refine criteria to use for problems of this type is an interesting but difficult problem. Once the cells are marked, they are clustered to form blocks that will form a new level in the hierarchy.

As with the single-block version of the Cartesian flow solver solver, the adaptive off-body flow solver is interfaced with the near-body and meshless solvers through Python interfaces (see Figure 1).

IV. Results

The results in this section were obtained using the near-body mesh shown in Figure 5(a) around the NACA 0012 airfoil. The mesh was generated using the mesh generation package of Jameson's FLO103 code,⁶ 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 resolution 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.

Figure 5. Flow over NACA 0012, $M = 0.8$, $\alpha = 1.25^\circ$.

To highlight the capabilities of the meshless interface, we perform a difficult shock capturing test. Figure 5(b) features a shock in transonic flow crossing the meshless interface region. Note the clean capturing of the discontinuity and the seamless transition across the meshless interface. This case highlights the ability of the meshless interface technique to 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 used in other regions of the domain covered by meshes. This is an advantage over interpolated interfaces, which can numerically smear solutions.

Before adding off-body adaptivity, we further validate the meshless interface technique by comparing our results with established methods for a well-known test case. Figure 6 contains solution contours and surface pressure plots for the NACA 0012 at $M = 0.8$, $\alpha = 10^\circ$ and $Re = 500$. The surface pressure curves for the hybrid scheme are overplotted with results from Jameson's FLO103 code, showing excellent agreement. Accompanying the set of plots is a table of lift and drag coefficients due to pressure comparing the hybrid scheme with FLO103 and the viscous scheme of Mavriplis.¹⁸ The lift and drag coefficients compare quite well for all cases tested.

Table 1. Lift and drag coefficients due to pressure

	c_l	c_d
FLO 103	0.4311	0.1421
Mavriplis ¹⁸	0.4469	0.1474
Hybrid	0.4234	0.1420

As the focus of this paper is combining the meshless interface technique with adaptive Cartesian grids, we present results for an adaptive viscous case. Figure 7 shows the calculation on a refined grid system that extends 10 chords on either side of the airfoil in the X direction, and 5 chords in the Y direction. This

Figure 6. Flow over NACA 0012, $M = 0.8$, $\alpha = 10^\circ$, $Re = 500$

grid system contained 3 levels of refinement, with the finest level $\Delta x = 0.03125$ chord. This multi-level grid contained a total of 28K points, approximately 10X fewer points than a uniformly fine grid extending over this domain with the finest level resolution.

The initial solution, shown in Figure 7(c), was refined according to density gradients near the airfoil surface. Figure 8 shows the resulting grid system and solution. Note the enhanced resolution of the density variations on both the top and bottom, and the aft region, of the airfoil. The number of gridpoints increases by about 30% over the non-adapted refined grid case shown in Figure 7, but is still a significant savings over using a uniformly refined grid everywhere.

V. Conclusions and Future Work

The results in this paper have demonstrated the ability of a meshless method to interface near-body curvilinear grids with off-body adaptive Cartesian grids. Our results compare favorably with established methods for two-dimensional laminar flow computations. We showed that point clouds may be obtained

(a) Cartesian block system.

(b) Near-body, meshless, and off-body mesh systems.

(c) Density contours.

Figure 7. Flow over NACA 0012, $M = 0.85$, $\alpha = 0.0^\circ$, $Re = 2000$.

(a) Cartesian block system.

(b) Adaptive Cartesian mesh.

(c) Density contours.

Figure 8. Flow over NACA 0012, $M = 0.85$, $\alpha = 0.0^\circ$, $Re = 2000$, with solution adaptivity.

efficiently from existing distributions in the interface region. The meshless interface solution procedure, along with the adaptive Cartesian grids, enabled sharp resolution of flow features.

While still in the conceptual stages, the meshless interface procedure displays some advantages over conventional interpolation for overset intergrid communication. First, the meshless method is based on the discretized PDEs governing the flow. This enables sharp resolution of flow features without smearing, as was shown in the inviscid shock case. Second, the meshless interface only requires clouds of points, which are derived from existing grid definitions in the intergrid region. The formation of these local clouds appears less demanding than rigorous donor cell identification for interpolation procedures. Finally, the meshless procedure fits conveniently within a multisolver approach, where the meshless solver is simply another discretization strategy, which complements near-body and Cartesian solvers.

The adaptive capability in the Cartesian region reduced grid size by an order of magnitude. Future work will extend the method to three dimensions, where further savings are expected through refinement in one more dimension. Additionally, the nested Cartesian off-body structure will extend naturally to multigrid algorithms to obtain even greater efficiency. Unsteady adaptive flows may then be computed efficiently with dual time stepping algorithms. Future work will also examine appropriate refinement criteria for different problems of interest. Recent work by Kamkar et al¹⁹ using feature detection algorithms to target refinement is one promising option. Improved refinement will allow for greater resolution in regions of interesting features while reducing computational cost in regions with little variation.

VI. Acknowledgements

The first author has benefited greatly from the NDSEG graduate fellowship program administered through the High Performance Computing Modernization Office (HPCMO) of the Department of Defense. The second author has benefited greatly from the long term and continuing support of the AFOSR Computational Mathematics Program directed by Dr. Fariba Fahroo.

References

- ¹Buning, P., Jespersen, D., Pulliam, T., Chan, W., Slotnick, J., Krist, S., and Renze, K., "OVERFLOW Users Manual," *NASA Langley Research Center*, 2003.
- ²Wissink, A. M., Sitaraman, J., Sankaran, V., Pulliam, T., and Mavriplis, D., "A Multi-Code Python-Based Infrastructure for Overset CFD with Adaptive Cartesian Grids," *46th AIAA Aerosciences Conference*, No. AIAA-2008-0927, Reno, NV, Jan 2008.
- ³Sitaraman, 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," *46th AIAA Aerosciences Conference*, No. AIAA-2008-0660, Reno, NV, Jan 2008.
- ⁴Katz, A. and Jameson, A., "Edge-based Meshless Methods for Compressible Viscous Flow with Applications to Overset Grids," *AIAA paper 2008-3989*, AIAA 38th Fluid Dynamics Conference, Seattle, WA, June 2008.
- ⁵Katz, A. and Jameson, A., "Multicloud convergence Acceleration for Complex Applications on Arbitrary Grids," *IEEE Proceedings of the 2008 DoD High Performance Computing Modernization Program*, 2009.
- ⁶Grasso, F., Jameson, A., and Martinelli, L., "A Multistage Multigrid Method for the Compressible Navier-Stokes Equations," *Notes on Numerical Fluid Mechanics*, Vol. 18, 1987, pp. 123–128.
- ⁷Nadarajah, S. K., "The Discrete Adjoint Approach to Aerodynamic Shape Optimization," Phd thesis, Stanford University, 2003.
- ⁸Meakin, R., "Composite Overset Structured Grids," *Handbook of Grid Generation*, CRC Press, 1999.
- ⁹Katz, A. and Jameson, A., "Edge-based Meshless Methods for Compressible Flow Simulations," *AIAA paper 2008-669*, AIAA 46th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2008.
- ¹⁰Jameson, 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.
- ¹¹Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 2 Artificial Diffusion and Discrete Shock Structure," *International Journal of Computational Fluid Dynamics*, Vol. 5, 1995, pp. 1–38.
- ¹²May, G. and Jameson, A., "Unstructured Algorithms for Inviscid and Viscous Flows Embedded and a Unified Solver Architecture: Flo3xx," *AIAA paper 2005-0318*, AIAA 43rd Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2005.
- ¹³Berger, M. J. and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, 1989, pp. 65–84.
- ¹⁴Hornung, R. D., Wissink, A., and Kohn, S., "Managing Complex Data and Geometry in Parallel Structured AMR Applications," *Engineering with Computers*, Vol. 22, No. 3, 2006, pp. 181–195.
- ¹⁵Hornung, R. D. and Kohn, S. R., "Managing application complexity in the SAMRAI object-oriented framework," *Concurrency and Computation: Practice and Experience*, Vol. 14, 2002, pp. 347–368.

- ¹⁶Wissink, A. M., Hornung, R. D., Kohn, S. R., Smith, S. S., and Elliott, N., “Large Scale Parallel Structured AMR Calculations Using the SAMRAI Framework,” *Proc. of SC01*, Denver, CO, November 10–16 2001.
- ¹⁷Kirshman, D. J. and Liu, F., “A Gridless Boundary Condition Method for the Solution of the Euler Equations on Embedded Cartesian Meshes with Multigrid,” *Journal of Computational Physics*, Vol. 201, 2004, pp. 119–147.
- ¹⁸Mavriplis, D. J., Jameson, A., and Martinelli, L., “Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes,” *AIAA paper* 1989-0120, AIAA 27th Aerospace Sciences Meeting, Reno, NV, January 1989.
- ¹⁹Kamkar, S., Jameson, A., and Wissink, A., “Automated Grid Refinement Using Feature Detection for Rotorcraft Simulations,” *AIAA paper* 2009-1496, AIAA 47th Aerospace Sciences Meeting and Exhibit, Orlando, FL, January 2009.