Efficiency of Mixed-Element USM3D for Benchmark Three-Dimensional Flows

Mohagna J. Pandya[1]
NASA Langley Research Center, Hampton, Virginia 23681, USA

Dennis C. Jespersen[2]
NASA Ames Research Center, Moffett Field, California 94035, USA

Boris Diskin[3]
National Institute of Aerospace, Hampton, Virginia 23666, USA

and

James L. Thomas[4], Neal T. Frink[5]
NASA Langley Research Center, Hampton, Virginia 23681, USA

The unstructured, mixed-element, cell-centered, finite-volume flow solver USM3D is enhanced with new capabilities such as line generation for general unstructured grids and optimization of the iterative solver. The efficiency of the new developments is assessed in a massively parallel environment. The assessments are conducted using a baseline method and the recent hierarchical adaptive nonlinear iteration method framework. Three benchmark turbulent flows, namely, a subsonic separated flow around a three-dimensional hemisphere-cylinder configuration, a transonic flow around the ONERA M6 wing, and a supersonic flow through a square duct are considered.

Nomenclature

b = generalized conjugate residual (GCR) search direction vector

A portion of the material in the present article was published as Paper 2019-2333 at the AIAA SCITECH Forum, San Diego, CA, USA, January 7-11, 2019.

[1] Research Aerospace Engineer, Configuration Aerodynamics Branch, Mail Stop 499, Senior Member AIAA.

[2] Computer Scientist (retired), Advanced Computing Branch, Mail Stop 258-5.

[3] NIA Senior Research Fellow, Associate Fellow AIAA.

[4] Senior Research Scientist (retired), Computational Aerosciences Branch, Mail Stop 399, Fellow AIAA.

[5] Distinguished Research Associate, Configuration Aerodynamics Branch, Mail Stop 499, Associate Fellow AIAA.

C_D = drag coefficient

C_L = lift coefficient

n_s = specified maximum number of search directions in hierarchical adaptive nonlinear iteration method

$\boldsymbol{Q}$ = vector of the discrete nonlinear solution

$\boldsymbol{R}(\boldsymbol{Q})$ = discrete nonlinear operator

$\boldsymbol{r}$ = GCR residual vector

r = radius of sphere defining the farfield boundary

V = control volume

x, y, z = spatial coordinates

y^+ = near-wall mesh spacing

γ = projection of the residual vector on the search direction

$\Delta\boldsymbol{Q}$ = vector of solution correction

$\Delta\tau$ = pseudotime step

ϵ = small factor used in Fréchet derivative

μ = inner product of two search directions

$\hat{\nu}$ = field variable used in Spalart-Allmaras (SA) turbulence models

$\frac{\partial\hat{\boldsymbol{R}}}{\partial\boldsymbol{Q}}$ = approximate Jacobian

$\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}$ = exact Jacobian

Superscripts

n = solution iteration

Subscripts

j, k = index of a GCR search direction

I. Introduction

USM3D is a cell-centered Reynolds-averaged Navier-Stokes (RANS) solver that is a part of the NASA Tetrahedral Unstructured Software System (TetrUSS) [1]. USM3D and TetrUSS have been extensively used for configuration aerodynamic research within NASA [1]-[3] and for product development within major airframe companies [4]-[6].

An overriding attribute of the USM3D flow solver has been its speed and robustness in providing solutions for a broad class of vehicles, which makes it a good candidate for further upgrades.

Revolutionary algorithmic improvements in modern computational fluid dynamics (CFD) codes are needed to address major challenges documented in the NASA CFD Vision 2030 report [7]. Time reduction of at least two orders of magnitude achieved through algorithmic advancements is the new goal for USM3D simulation of complex flows around complex geometries. Encouraging initial strides have been made in this direction already. USM3D has been extended to support the solutions on mixed-element grids [8] that provide improved numerical simulation using flow-aligned anisotropic hexahedral or prismatic cells in a boundary layer and isotropic tetrahedral cells away from a boundary layer. A new strong nonlinear solver named Hierarchical Adaptive Nonlinear Iteration Method (HANIM) [9] has been implemented in the mixed-element USM3D to improve the robustness, efficiency, and automation of the RANS solutions. The USM3D HANIM methodology using a point-implicit preconditioner was initially assessed on several turbulent flow benchmark cases [9]. In that study, the Spalart-Allmaras (SA) eddy viscosity turbulence model [10], [11] was used to provide closure for the RANS solutions. The convergence acceleration factor ranging from 1.4 to 13 was demonstrated relative to the USM3D baseline solver.

A further enhancement to the efficiency of the mixed-element USM3D HANIM was achieved by implementing a line-implicit preconditioner [12]. The line generator used in that study was limited to only prismatic and/or hexahedral cells and imposed restriction that cells be of the same topology within a given line. An assessment of the iterative convergence of HANIM with point- and line-implicit preconditioners for steady-state RANS solutions [12] demonstrated that only line-implicit HANIM met convergence targets on all grids. An assessment of time to solution normalized by degrees of freedom showed that the case-to-case variation in the performance of the line-implicit HANIM is significantly less than the corresponding variation in the performance of the point-implicit HANIM. The performance of the line-implicit HANIM is also less sensitive to an increase in degrees of freedom. The aforementioned studies were performed using a sequential version of the mixed-element USM3D.

The current paper reports on new developments in the parallel mixed-element USM3D [13]. This paper provides the following new contributions: (1) a line-generation algorithm suitable for general unstructured grids, (2) a cost-effective implementation of a Krylov method with many search directions, and (3) a detailed solver efficiency study, including assessments of the effects of grid topology, grid resolution, and convergence criteria. The HANIM efficiency of the parallel mixed-element USM3D is assessed on three benchmark three-dimensional (3D) configurations, namely, a hemisphere-cylinder (HC) configuration, the ONERA M6 (OM6) wing, and a square-duct (SD) configuration. Grids as large as 1.6 billion cells are used for the present assessments. The benchmark configurations and flow conditions are introduced in Section II. Section III overviews the principal elements of the preceding versions of the mixed-element USM3D [8], [9], [12]. Section IV provides a detailed description of the newly implemented features of USM3D. The computational results are reported in Section V. Concluding remarks are presented in Section VI.

II. Benchmark Turbulent Flows

In this paper, the efficiency of the parallel mixed-element USM3D is assessed on three benchmark 3D flows, namely, a subsonic separated flow around an HC configuration [14], a transonic flow around an OM6 wing [15], and a supersonic flow through an SD configuration [16]. These benchmark flows are described in detail at the NASA Turbulence Modeling Resource (TMR) website[*] in the "Cases and Grids for Turbulence Model Numerical Analysis" section. Below is a brief summary of the geometry and flow parameters.

Figure 1 shows geometry and boundary conditions (BCs) for the HC configuration. Both cylinder and hemisphere have diameters of unity. The combined length of the configuration is 10. The apex of the hemisphere is located at the origin of the coordinate system. The outflow BC is assigned at a plane that is orthogonal to the cylinder axis that coincides with the x-axis and contains the cylinder base located at $x = 10$. The symmetry BC is assigned at the vertical plane corresponding to $y = 0$. The farfield boundary is a quadrant of a sphere ($(x - 10)^2 + y^2 + z^2 = r^2$, $(10 - r) \leq x \leq 10,\ 0 \leq y \leq r,\ -r \leq z \leq r$) with the radius $r = 100$. The flow corresponds to the reference (freestream) Mach number of 0.6, the Reynolds number of 3.5×10^5 based on the unit length, the angle of attack of 19°, and the reference temperature of 540°R.

The OM6 wing geometry and BCs are illustrated in Fig. 2. The root chord aligns with the x-axis. The leading edge is at $x = 0.0$ and the trailing edge is at $x = 1.0$. The approximate values for the semispan and the mean aerodynamic chord are 1.47602 and 0.80167, respectively. The farfield boundary is a hemisphere ($x^2 + y^2 + z^2 = r^2,\ y \geq 0$) with the radius $r = 100$. The symmetry BC is assigned at the plane ($y = 0$) containing the root airfoil. More details of the wing geometry are available at the TMR website. The flow corresponds to the reference Mach number of 0.84, the Reynolds number of 1.46×10^7 based on the root chord, the angle of attack of 3.06°, and the reference temperature of 540°R.

[*] https://turbmodels.larc.nasa.gov/

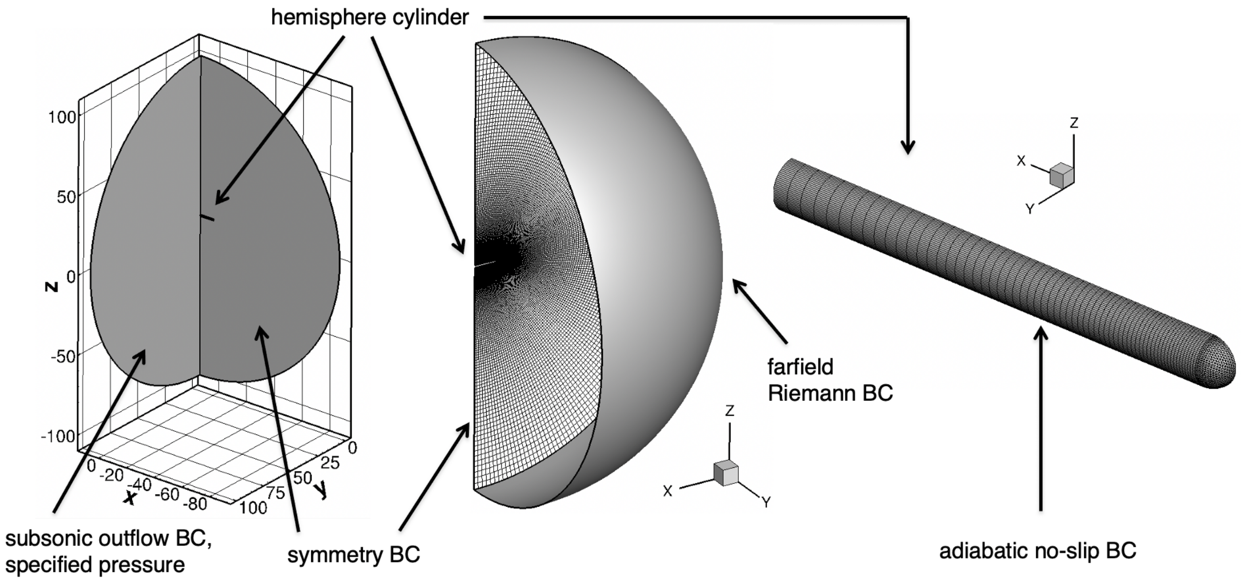


Fig. 1 HC configuration: geometry and boundary conditions.

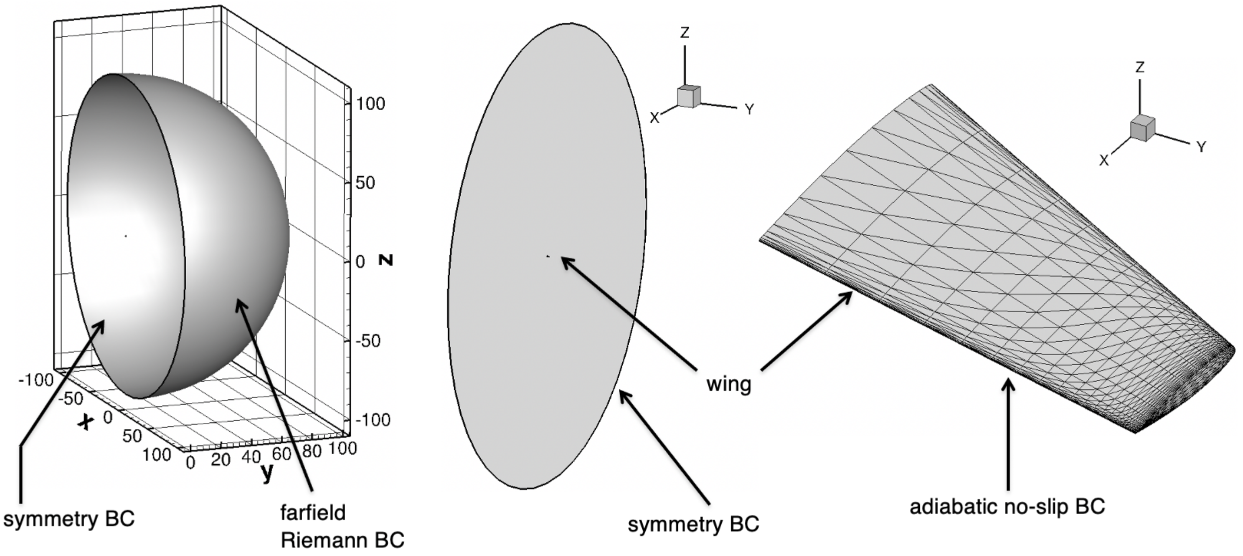


Fig. 2 OM6 wing: geometry and boundary conditions.

The primary focus of the SD case is to assess nonlinear turbulence models for the supersonic internal flow through a geometry with corners. In such cases, anisotropy of turbulence stresses can be important because normal stress differences induce flowfield behavior that cannot be captured with linear eddy-viscosity models. In the present study, the Spalart-Allmaras Quadratic Constitutive Relation (SA-QCR)-2000 model [17] is used. The geometry includes a duct with a constant square cross-sectional area of nondimensional height and width of one. The duct nondimensional length is 52. The SD configuration geometry and BCs are illustrated in Fig. 3. The origin of the coordinate system is set at the lower-left corner of the front section of the duct. The positive direction of the x-axis is the streamwise direction along the duct. The duct dimensions are $0 \le x \le 52; 0 \le y \le 0.5;\ 0 \le z \le 0.5$. Adiabatic-wall conditions are set at the duct boundaries corresponding to $0 \le x \le 52; y = 0;\ 0 \le z \le 0.5$ and $0 \le x \le 52; 0 \le y \le 0.5;\ z = 0$. The computational domain is extended upstream with a small segment of symmetry boundaries at $-1.26829 \le x \le 0$. The flow conditions correspond to the reference Mach number of 3.9, the Reynolds number of 5.08×10^5 based on the unit length, the angle of attack of 0°, and the reference temperature of 520°R. Supersonic inflow conditions are set at $x = -1.26829$ corresponding to the reference conditions. Supersonic outflow conditions are set at $x = 52$. Symmetry conditions are also specified at the boundaries corresponding to $y = 0.5$, and $z = 0.5$, thus, only one quarter of the duct is computed. The reference area is unity.

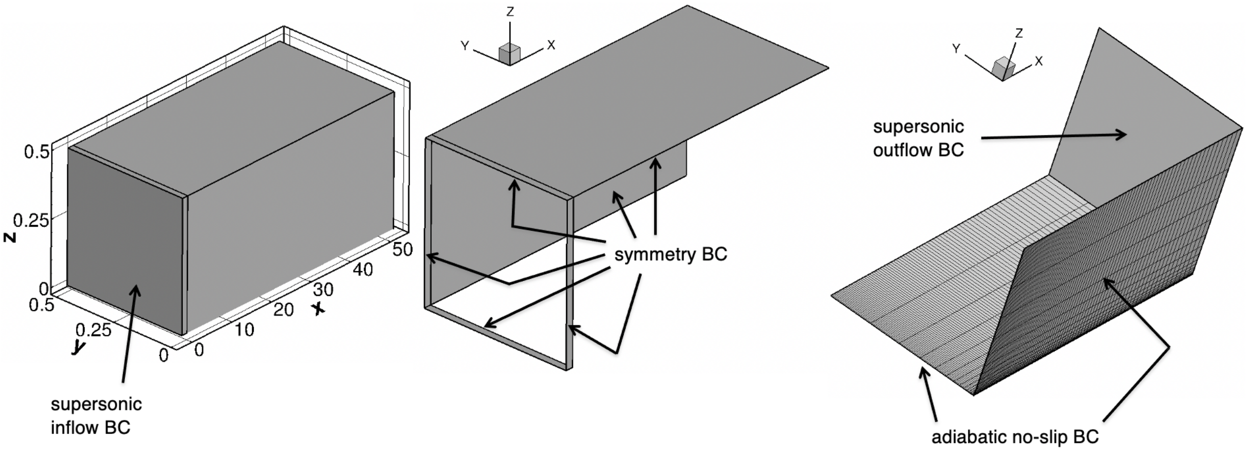


Fig. 3 SD configuration: geometry and boundary conditions.

III. Review of Preceding Mixed-Element USM3D

The salient features of the preceding versions of the mixed-element USM3D [8], [9], [12] are described here. USM3D solves a system of nonlinear flow equations that can be formally represented as

$$\boldsymbol{R}(\boldsymbol{Q}) = 0. \tag{1}$$

The discrete nonlinear operator, $\boldsymbol{R}(\boldsymbol{Q})$, represents a cell-centered discretization of the steady-state RANS equations. The term "cell-centered" means that the independent variables used by the governing equations are the flow variables defined at the centroid of each cell. The solution variables at the grid nodes and boundary faces, as well as the cell and face gradients, are computed solely from the current solution variables defined at the cell centers.

A reconstruction process based on solution gradients computed within cells is used to obtain the USM3D second-order spatial discretization of inviscid fluxes. The cell gradients for inviscid fluxes are computed by a Green-Gauss integration procedure using solution values interpolated to grid nodes. Inviscid fluxes are computed at each cell face using various upwind schemes, such as Roe's Flux Difference Splitting (FDS) [18], van Leer's Flux Vector Splitting (FVS) [19], Harten, Lax, van Leer, Einfeldt (HLLE) [20], or Harten, Lax, van Leer – Contact (HLLC) [21] schemes, among others. The standard [10] and negative (SA-neg) [11] variants of the SA one-equation turbulence model are available. The convective term of the SA turbulence model equation can be approximated with either second- or first-order accuracy. Face gradients are needed for the diffusion fluxes of the meanflow and turbulence model equations. A face gradient is evaluated from Mitchell's stencil [22], [23]. A face-area weighted average of the face velocity gradients provides the cell-based gradients for the SA source term.

The cell-centered solution values are required to satisfy realizability constraints, such as positive density and pressure. For second-order spatial accuracy, a realizability check is implemented for the reconstructed solution to avoid a catastrophic failure (an error condition associated with a square root of a negative number) in computing meanflow inviscid fluxes. If a realizability violation is detected at a face during any nonlinear iteration, the face is temporarily designated as "first-order". The "first-order" designation for a face is removed after the second-order reconstruction values have been realizable for 20 consecutive nonlinear iterations.

A preconditioner is the central component of USM3D's nonlinear iteration strategies. The USM3D preconditioner computes a solution correction, $\Delta\boldsymbol{Q}$ using a defect correction scheme

$$\frac{V}{\Delta\tau}\Delta\boldsymbol{Q} + \frac{\partial\widehat{\boldsymbol{R}}}{\partial\boldsymbol{Q}}\,\Delta\boldsymbol{Q} = -\boldsymbol{R}(\boldsymbol{Q}^n), \tag{2}$$

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^n + \Delta\boldsymbol{Q}. \tag{3}$$

Here, $\boldsymbol{Q}^n$ and $\boldsymbol{Q}^{n+1}$ are the solutions at iterations n and $n+1$, respectively. The term $\frac{\partial\widehat{\boldsymbol{R}}}{\partial\boldsymbol{Q}}$ approximates the Jacobian $\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}$, V is a control volume and $\Delta\tau$ is a pseudotime step, which is set through a Courant-Friedrichs-Lewy (CFL) number specification. The meanflow and turbulence model preconditioner equations are loosely coupled. The approximate Jacobian for the meanflow equations is formed using the linearization of the first-order FVS or FDS inviscid fluxes and a thin-layer approximation for the viscous fluxes. The approximate Jacobian for a turbulence model equation includes the contributions from the advection, diffusion, and source terms. The advection term is linearized with a first-order approximation. A thin-layer approximation is used for the diffusion term. The entire contribution from the linearized source term is added to the diagonal. A positivity check for the diagonal values is conducted before adding the pseudotime term. Negative diagonal values are substituted by their absolute values. An option to use single precision for the approximate Jacobian off-diagonal terms and for the solution updates is available to reduce the memory footprint. The preconditioner equation (Eq. (2)) is solved using either point- or line-implicit Gauss-Seidel (GS) iterations. Point-implicit GS iterations use a lexicographic order to update solutions at cells; line-implicit GS iterations follow a multicolor order.

USM3D offers two iteration methodologies to advance a nonlinear solution. In the baseline methodology, designated as Preconditioner-Alone (PA) solver, the correction computed by the preconditioner is directly applied to update the current nonlinear solution. The preconditioner is based on a linearization of a simplified discrete formulation and uses a fixed number of a point-implicit Gauss-Seidel (GS) iterations. CFL is often ramped over a specified number of nonlinear iterations to a specified maximum value and then kept unchanged. The other solution methodology is HANIM. HANIM provides two additional hierarchies around the preconditioner of USM3D. The HANIM hierarchies are a matrix-free linear solver for the exact linearization of RANS equations and a nonlinear control of the solution updates. The goal of these hierarchies is to enhance the iterative scheme with a mechanism for an automatic adaption of the pseudotime step to increase convergence rate and overcome instabilities and limit cycles. The matrix-free linear solver [24]-[28] uses Fréchet derivatives and a Generalized Conjugate Residual (GCR) method [25] from the family of Krylov methods. The nonlinear update methodology is close to the one discussed in Ref. [28]. HANIM prescribes the residual reduction targets for the preconditioner, GCR, and nonlinear solution updates and specifies the maximum number of linear iterations allowed in the preconditioner and the maximum number of search directions allowed in GCR. Unlike in the PA method, the HANIM CFL update strategy is adaptive. HANIM increases CFL if all the HANIM hierarchies have reported success. On the other hand, if any of preconditioner, GCR, or nonlinear solution update hierarchies fails, HANIM discards the suggested correction and aggressively reduces CFL. The flowcharts of the PA and HANIM iterations are shown in Ref. [9] and supplemental Figure_S1.pdf and Figure_S2.pdf, respectively. The mixed-element USM3D has recently been parallelized for Central Processing Unit (CPU)-based high-performance computing architectures using the message passing interface paradigm [13].

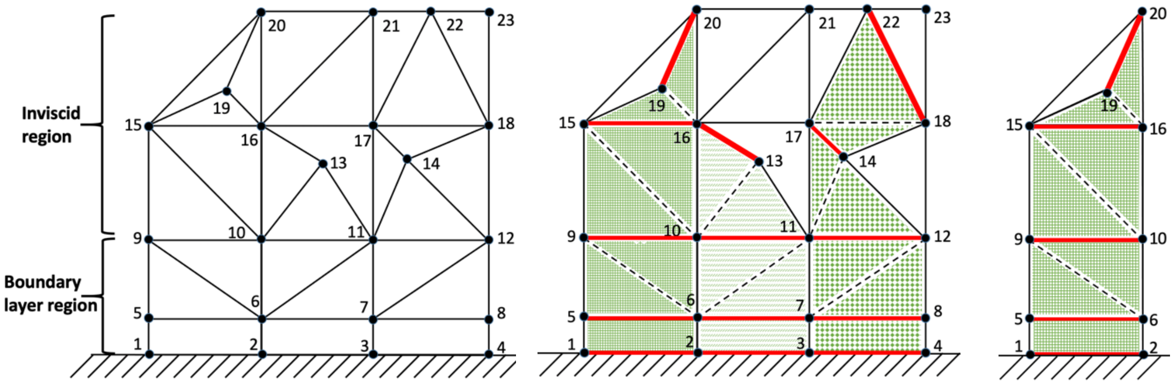
IV. New Developments

A. Line Generation for General Unstructured Grids

A line generation algorithm for general unstructured grids has been developed and implemented within USM3D. Currently, lines are generated in a standalone serial preprocessing step that produces a line file on a disk. The line file serves as an input to the highly scalable USM3D solver [13]. The line construction is based solely on local grid connectivity, i.e., no geometrical data (such as grid coordinates, normal direction, face area, or cell volume) are used.

The lines are generated from a prescribed set of faces, typically on a computational boundary, such as a no-slip aerodynamic surface. The line generation process is not confined to a boundary layer. In a grid that has inherent advancing-layer structure over the entire computational domain, lines are expected to span the entire grid. In a general unstructured grid that has an advancing layer structure only within a boundary layer, lines can span a significant portion of the grid outside of a boundary layer, but some cells may not belong to any line.

An introduction to line terminology and overview of the line generation algorithm are presented using Fig. 4. A notional unstructured grid consisting of 23 nodes is presented in Fig. 4(a) that shows grid layout within and outside of the boundary layer region. In the 2D illustration, a face is represented by an edge that connects two nodes. Fig. 4(b) highlights various grid lines that are identified by the present line generation algorithm.



(a) Grid

(b) Multiple lines

(c) Single line

Fig. 4 Schematic of an unstructured grid and identification of lines.

USM3D generates lines using a recursive advancing face-layer algorithm. A concept of global face layers is introduced first using Figure 4(b). Faces (1, 2), (2, 3), and (3, 4) belong to the first (prescribed) global layer of faces from where lines originate. Faces (5, 6), (6, 7), and (7, 8) belong to the second global layer of faces. Faces (9, 10), (10, 11), and (11, 12) belong to the third global layer of faces. The fourth global layer of faces includes faces (15, 16), (16, 13), and (17, 14). The fifth global layer of faces includes faces (19, 20) and (22, 18). The last face within each line is called a terminal face. In Fig. 4(b) the faces (19, 20), (16, 13), and (18, 22) are the terminal faces where the left, middle, and right lines end, respectively. Any face that belongs to a global face layer is called a "layer face". A global node layer consists of the nodes of the layer faces that belong to a specific global face layer. For example, nodes 1, 2, 3, and 4 belong to the first global layer of nodes; nodes 5, 6, 7, and 8 belong to the second global layer of nodes; nodes 15, 16, 13, 17, 14 belong to the fourth global layer of nodes; and nodes 19, 20, 22, and 18 belong to the fifth global layer of nodes.

Figure 4(c) shows an individual line and its cell and face structures. By definition, each grid line is an ordered sequence of face-connected cells. An interior cell within a line is face-connected to two neighboring cells of the same line, namely, the preceding cell and the succeeding cell. The first and last cells in a line are each face-connected to only one cell in the line. One can also view a grid line as advancing layers of faces. Each line starts from a bottom face and ends with a terminal face. The layer faces of the same line do not intersect. The cells between two consecutive layer faces of a line form a cell layer of the line. A cell layer of a line can be composed of a single cell or several face-connected cells. For example, a cell layer between two quadrilateral layer faces can be composed of a single hexahedron or of two side prisms; a cell layer between two triangular layer faces can be composed of a single prism or three tetrahedral cells. The cells constituting a cell layer of a line have all of their nodes on the two consecutive layer faces of the line that bound the cell layer. The faces connecting cells within a cell layer are interior faces and not layer faces of a line. In Figure 4(c), the faces (1, 2), (5, 6), (9, 10), (15, 16), and (19, 20) are layer faces. The face (19, 20) is the terminal face of the line. The first cell layer includes a single cell (1, 2, 6, 5), the second cell layer includes cells, (5, 6, 9) and (9, 6, 10), the third cell layer includes cells (9, 10, 15) and (15, 10, 16), and the fourth cell layer includes cells (15, 16, 19) and (19, 16, 20). Faces (9, 6), (15, 10), and (19, 16) are interior faces within a line.

The guidelines for developing the USM3D line-generation algorithm are noted below:

- A line that is attached to a physical (not symmetry) boundary at the current layer should preferably remain attached to the same physical boundary at the next layer.
- The gaps between lines should be minimized. Lines that are attached at the current layer should preferably be attached at the next layer as well.

Given a global set of *current* layer faces, the core procedure of the line generation algorithm is to identify a global set of *next* layer faces and try to pair each *current* layer face with a *next* layer face. The outcome of the procedure is that each current layer face is either paired or becomes terminal. This procedure is identified as "pairing between *current*- and *next*-layer faces". A specific advancing-front procedure called "sequencing of current-layer faces for pairing" is devised to consider each *current*-layer face for pairing with the *next*-layer face, following the aforementioned guidelines. Details of the line generation algorithm and assessment of its performance are reported in Ref. [13] and a supplemental file, Data_S1.pdf.

B. Revised GCR Implementation

USM3D's original GCR implementation has been revised to reduce the time to solution. The revisions are pertinent to the GCR with multiple search directions. The revisions produce an early termination of a loop through multiple search directions when either convergence stall or an unsatisfactory rate of convergence is detected. In HANIM-n_s, the GCR method uses up to a maximum n_s search directions to approximately solve the linear equations

$$\frac{V}{\Delta\tau}\Delta\boldsymbol{Q} + \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}\Delta\boldsymbol{Q} = -\boldsymbol{R}(\boldsymbol{Q}^n). \tag{4}$$

The nonlinear residual operator is linearized around the current solution, $\boldsymbol{Q}^n$. The matrix-vector product in Eq. (4) is approximated by the Fréchet derivative

$$\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}\Delta\boldsymbol{Q} \approx \frac{\boldsymbol{R}\left(\boldsymbol{Q}^n + \epsilon\frac{\Delta\boldsymbol{Q}}{|\Delta\boldsymbol{Q}|}\right) - \boldsymbol{R}(\boldsymbol{Q}^n)}{\epsilon}|\Delta\boldsymbol{Q}|. \tag{5}$$

Here, $|\cdot|$ denotes the l_2 norm of a vector.

The initial solution is set as $\Delta\boldsymbol{Q} = 0$, thus the initial GCR residual is

$$\boldsymbol{r}_0 = -\boldsymbol{R}(\boldsymbol{Q}^n). \tag{6}$$

For the k-th search direction ($1 \leq \mathrm{k} \leq n_s$), a solution correction, $\Delta\boldsymbol{Q}_k$, is computed as described below. The initial correction is typically provided by a preconditioner. The new search direction, $\boldsymbol{b}_k$, is computed and normalized as

$$\boldsymbol{b}_k = \frac{V}{\Delta\tau}\Delta\boldsymbol{Q}_k + \frac{\boldsymbol{R}\left(\boldsymbol{Q}^n + \epsilon\frac{\Delta\boldsymbol{Q}_k}{|\Delta\boldsymbol{Q}_k|}\right) - \boldsymbol{R}(\boldsymbol{Q}^n)}{\epsilon}|\Delta\boldsymbol{Q}_k|; \quad \Delta\boldsymbol{Q}_k = \frac{\Delta\boldsymbol{Q}_k}{|\boldsymbol{b}_k|}, \ \boldsymbol{b}_k = \frac{\boldsymbol{b}_k}{|\boldsymbol{b}_k|}. \tag{7}$$

Then the $\boldsymbol{b}_k$ vector is orthonormalized against previously stored search directions, $\boldsymbol{b}_j, 1 \leq j < k,$ simultaneously updating $\Delta\boldsymbol{Q}_k$. For each $\boldsymbol{b}_j$ $(j < k)$,

$$\mu = \boldsymbol{b}_k^T\boldsymbol{b}_j, \qquad \boldsymbol{b}_k = \boldsymbol{b}_k - \mu\boldsymbol{b}_j, \quad \Delta\boldsymbol{Q}_k = \Delta\boldsymbol{Q}_k - \mu\Delta\boldsymbol{Q}_j, \qquad \Delta\boldsymbol{Q}_k = \frac{\Delta\boldsymbol{Q}_k}{|\boldsymbol{b}_k|}, \qquad \boldsymbol{b}_k = \frac{\boldsymbol{b}_k}{|\boldsymbol{b}_k|}. \tag{8}$$

After completing the orthonormalization procedure, the projection, γ_k, of the current residual on the kth search direction is computed, and the GCR correction and the linear residual are updated as

$$\gamma_k = \boldsymbol{b}_k^T\boldsymbol{r}_{k-1}, \ \Delta\boldsymbol{Q} = \Delta\boldsymbol{Q} + \gamma_k\Delta\boldsymbol{Q}_k, \ \boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \gamma_k\boldsymbol{b}_k. \tag{9}$$

The GCR method repeats until either the target residual reduction, μ_{gcr}, has been achieved (i.e., $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_0\| < \mu_{gcr}$, $\|\cdot\|$ denotes the root-mean-squares (rms) norm) or the specified maximum number of search directions has been used. The GCR method is deemed to have failed if, at the end, the residual reduction target is not achieved.

HANIM often provides the best efficiency (fastest time to solution) when only one search direction is specified for the GCR method. HANIM-1 is a memory-efficient algorithm. The computer memory for storing search directions is allocated statically and therefore a larger number for the maximum search directions significantly increases the memory footprint of HANIM. However, there are problems where multiple search directions are needed. In such cases, it has been found useful to implement several early-termination sensors to avoid wasteful computations that may be incurred in a convergence-stall scenario or when the GCR method appears to have no reasonable chance to meet the residual reduction target. The stall condition occurs when vectors $\boldsymbol{b}_k$ and $\boldsymbol{r}_{k-1}$ are (nearly) orthogonal, implying $\gamma_k \ll |\boldsymbol{r}_{k-1}|$. The stall has been observed even for scalar elliptic equations. Under this condition, the solution update is negligible (almost none). The stall detection sensor compares the maximum linear residual update with a representative value of the previous residual. The GCR stall condition is detected if

$$\max(ABS(\gamma_k \boldsymbol{b}_k)) < \ \|\boldsymbol{r}_{k-1}\|, \tag{10}$$

where $\gamma_k \boldsymbol{b}_k$ is the residual correction applied in Eq. (9), and the maximum magnitude is computed over all vector components. Even when stall is not detected, the residual reduction from a new search direction may be too small to attain an eventual residual reduction target. The decision for an early termination of multiple search direction loops is based on the reduction rate of the current linear residuals. The early termination occurs if $\|\boldsymbol{r}_k\|/\|\boldsymbol{r}_{k-1}\| < \mu_{gcr}^{1/n_s}$, where $\boldsymbol{r}_{k-1}$ and $\boldsymbol{r}_k$ are the linear residuals before and after the k-th search direction, respectively.

To assess the benefits of the revised GCR implementation, computations have been performed using two different tetrahedral grids around the OM6 wing configuration. Both grids are generated exclusively using advancing layers [29]. The coarse and fine grids have 5,677,056 and 45,416,448 cells, respectively. The coarse-grid solutions are computed using 24 partitions and the fine-grid solutions are computed using 2,560 partitions. For each grid, three variants of the HANIM solver are applied, namely, HANIM-1, HANIM-30-org, and HANIM-30. For HANIM-1, GCR uses only one search direction. For HANIM-30-org, a maximum of 30 search directions are allowed for the GCR. As the name implies, this is USM3D's original implementation that potentially uses all 30 search directions if a linear residual reduction target is not achieved. HANIM-30 employs a revised GCR that potentially uses all 30 search directions but can terminate early using the convergence-stall and slow-convergence sensors described previously.

The specified linear residual reduction target within GCR is $\mu_{gcr} = 0.92$ for the HANIM-1 solver, and $\mu_{gcr} = 0.5$ for the HANIM-30-org and HANIM-30 solvers.

A HANIM solution is converged when the rms value of the combined meanflow and SA turbulence model residuals reduces to a level of 10^{-13}. The assessment is based on two preconditioners that are also used for the computations reported in Section V. One preconditioner is a mixed point- and line-implicit preconditioner; for 30% of the grid around the wing, the line-implicit preconditioner is applied and for the rest of the grid, the point-implicit preconditioner is applied. The second preconditioner is a point-implicit preconditioner.

Table 1 summarizes various aspects of convergence using the mixed preconditioner on the coarse grid and the point-implicit preconditioner on the fine grid. The number of nonlinear iterations and the wall-clock time needed to converge are shown. Additionally, the table also shows the number of search directions and the number of preconditioner GS iterations en route to the nonlinear solution convergence. For both grids, the HANIM-30-org solver converges in the least number of nonlinear iterations and encounters the fewest GCR failures, as expected. However, it also uses the most search directions and preconditioner GS iterations resulting in the most time to solution. Comparatively, the HANIM-30 solver needs an increased number of nonlinear iterations and encounters more GCR failures but uses a significantly smaller number of search directions and preconditioner GS iterations. The HANIM-30 solver needs the least time to solution. The HANIM-30 time is approximately half of the HANIM-30-org time. Relative to the two HANIM solvers with a maximum of 30 search directions, HANIM-1 uses many more nonlinear iterations and encounters many more GCR failures.

Table 1 Solution convergence statistics for two tetrahedral grids around OM6 wing.

	Grid	Nonlinear	Wall-clock	GCR search	GCR	GS iterations	
		iterations	time, hours	directions	failures	meanflow	SA model
HANIM-1		40,010	35.82	52,051	12,039	875,845	295,765
HANIM-30-org	coarse	13,542	46.25	58,281	512	1,382,915	1,041,805
HANIM-30		16,157	24.22	35,944	4,692	734,595	340,800
HANIM-1		707,288	49.53	920,187	212,833	9,223,710	8,549,755
HANIM-30-org	fine	161,807	84.27	1,532,221	2,800	32,222,795	49,707,575
HANIM-30		264,513	43.34	843,645	79,384	14,342,390	12,287,850

V. Results

A. Grids

For the 3D HC configuration, a family of prismatic-hexahedral grids has been generated using a set of Fortran programs available at the TMR website under the “Cases and Grids for Turbulence Model Numerical Analysis” section, “3D Hemisphere Cylinder (new)” subsection. The grid-generation program is described in Ref. [29]. The finest-grid target near-wall spacing corresponds to a $y^+ = 0.5$. The finest prismatic-hexahedral grid in the family has been generated using the following input parameters [29]: 512 elements along the cylinder axis, 128 elements from the hemisphere apex to its base, and 2,560 elements in the radial direction. This grid has eight times more cells than the finest prismatic-hexahedral grid used in the previous study [30]. A grid-coarsening program (also available at the TMR website and described in Ref. [29]) is used to extract nested coarser grids in the grid family. The coarsening program applies an isotropic coarsening in the extraction of coarser grids. Table 2 provides statistics of the HC configuration prismatic-hexahedral grid, including the number of grid partitions used in the computations.

Table 2 Statistics for family of prismatic-hexahedral grids for HC configuration.

Grid	Partitions	Prisms	Hexahedra	Total cells	Nodes
5	1	30,720	122,880	153,600	147,637
4	5	245,760	983,040	1,228,800	1,143,081
3	42	1,966,080	7,864,320	9,830,400	8,995,153
2	336	15,728,640	62,914,560	78,643,200	71,368,353
1	2,520	125,829,120	503,316,480	629,145,600	568,585,537

For the OM6 wing, a tetrahedral grid family has been generated using a set of Fortran programs available at the TMR website under the “Cases and Grids for Turbulence Model Numerical Analysis” section, “3D ONERA M6 Wing” subsection. In the family, the finest-grid target near-wall spacing corresponds to a $y^+ = 0.5$. Using a grid-coarsening program (available at the TMR website) additional three nested isotropically-coarsened grids are extracted within the family. Table 3 provides statistics for the OM6 wing tetrahedral grids, including the number of grid partitions used in computations.

Table 3 Statistics for family of tetrahedral grids for OM6 wing.

Grid	Partitions	Tetrahedra	Total cells	Nodes
4	3	709,632	709,632	121,841
3	24	5,677,056	5,677,056	960,225
2	192	45,416,448	45,416,448	7,625,153
1	1536	363,331,584	363,331,584	60,777,345

A family of hexahedral grids has been generated using Fortran programs available at the TMR website under the "Cases and Grids for Turbulence Model Numerical Analysis" section, "3D Modified Supersonic Square Duct Validation Case" subsection. The grid family includes four nested hexahedral grids. Table 4 provides grid statistics. The grids are stretched in the wall-normal directions with a constant stretching factor. Grid 1 has 2688 cells in the streamwise direction and 768 cells in the spanwise and vertical directions, the minimum nondimensional spacing at the wall of approximately 0.0000085, and the average $y^+ = 0.022$.

Table 4 Statistics for family of hexahedral grids for SD configuration.

Grid	Partitions	Hexahedra	Total cells	Nodes
4	24	3,096,576	3,096,576	3,641,283
3	192	24,772,608	24,772,608	25,068,577
2	1,536	198,180,864	198,180,864	199,362,625
1	6,144	1,585,446,912	1,585,446,912	1,590,169,729

B. Efficiency Assessment

The efficiency of the parallel mixed-element USM3D flow solver is assessed in this section. The HANIM speedup over the baseline PA solver is presented. In all evaluations, the wall-clock time required for solution to satisfy the prescribed convergence criterion is used as the principle measure of iterative convergence. The CPU time can be calculated as the product of the wall-clock time and the number of partitions. Several solution convergence criteria, such as machine-zero residuals, six significant digits of accuracy for aerodynamic force coefficients, and an engineering accuracy for aerodynamic coefficients, are considered.

The iterative convergence of parallel mixed-element USM3D HANIM is assessed for benchmark flows over the HC configuration and OM6 wing and through an SD configuration. The benchmark flows and corresponding grids are described in Sections II and V.A, respectively. The HANIM solver technology outlined in Sections III and IV.B is assessed relative to the baseline PA method. The PA method computes solutions with a point-implicit preconditioner. In this study, HANIM uses a hybrid point- and line-implicit preconditioner for the prismatic-hexahedral grids and a point-implicit preconditioner for the tetrahedral and hexahedral grids. In the prismatic-hexahedral grids for the relatively simple H-C configuration, grid lines have been generated from the viscous surface through the entire domain. To represent practical grids, grid lines have been truncated to a near-body region containing the boundary layer. Only 30% of the cells in each grid line use the line-implicit preconditioner; other cells use the point-implicit preconditioner. Line-implicit GS preconditioner iterations are performed in a multicolor order whereas, point-implicit GS iterations are performed in the lexicographic order. Two types of HANIM solvers are used: HANIM-1 is used to compute solutions on the prismatic-hexahedral and hexahedral grids and HANIM-30 is used to compute tetrahedral-grid solutions. HANIM-30 uses the early-termination sensors described in Section IV.B. Unless specified otherwise, solutions are obtained using the same set of common input parameters. Some of these parameters are identified based on past experience, judgment, and studies [9], [12]. Meanflow and SA-model initial CFL for the present HANIM solutions are established based on precursor studies aimed at a faster convergence. The following is a brief summary of the solver input parameters:

1) Turbulence model: SA-neg (HC configuration and OM6 wing) and SA-neg-QCR2000 (SD configuration).
2) Meanflow convection terms: Roe's FDS, second order, no limiters.
3) SA-model convection term: upwind, first order.
4) Nodal solutions: nodal averaging scheme based on adaptive-dimensional pseudo-Laplacian method.
5) Cell gradients for meanflow convection terms: Green-Gauss based on nodal solutions.
6) Face gradients for meanflow and SA model diffusion terms: Mitchell's method.
7) Meanflow diffusion terms: full Navier-Stokes, second order.
8) SA-model diffusion term: full Navier-Stokes, second order.
9) Meanflow approximate-Jacobian convection terms: the PA method uses linearization of first-order van Leer's FVS fluxes, HANIM uses linearization of first-order Roe's FDS fluxes.
10) Approximate-Jacobian diffusion terms: thin-layer approximation.
11) Cell gradients for SA-model source term: area-weighted face gradients.

12) Residual reduction target for preconditioner: 0.5 for hybrid point- and line-implicit preconditioner, 0.1 for point-implicit preconditioner.

13) Preconditioner failure conditions (for HANIM): (a) the rms norm of residuals increases by more than a factor of two, or (b) the rms norm at the end of preconditioner iterations exceeds 95% of the maximum residual recorded during the GS iterations.

14) Maximum number of GS iterations for preconditioner: 15 for the PA method, 500 for HANIM.

15) Early termination of GS iterations: for the PA method, termination occurs if residual reduction target is met in the HC and OM6 solutions. For the SD case, the PA method always uses the specified maximum number of GS iterations. For HANIM, termination occurs if either residual reduction target is met or a failure condition is detected.

16) Frequency of sampling preconditioner residual: after the 5th, every 10th, and last GS iteration.

17) Residual reduction target for GCR: 0.92 for HANIM-1 and 0.5 for HANIM-30.

18) Residual reduction target for nonlinear control: 0.96 for HANIM-1 and 0.9 for HANIM-30.

19) CFL: the PA method ramps CFL from 1 to 150 over 150 iterations; uses maximum CFL of 25 for cells with large aspect ratio. HANIM updates CFL adaptively with initial CFL = 3 for the meanflow and initial CFL = 1 for the SA model. HANIM CFL is increased by a factor of two, if all hierarchies report success; decreased by a factor of 10, if any hierarchy reports failure. There is no HANIM CFL limit.

20) Initial conditions: freestream flow.

21) Initial iterations with first-order meanflow inviscid fluxes: 500 for the PA method, none for HANIM on HC configuration; 500 for HANIM on OM6 wing and 200 for HANIM on the SD case.

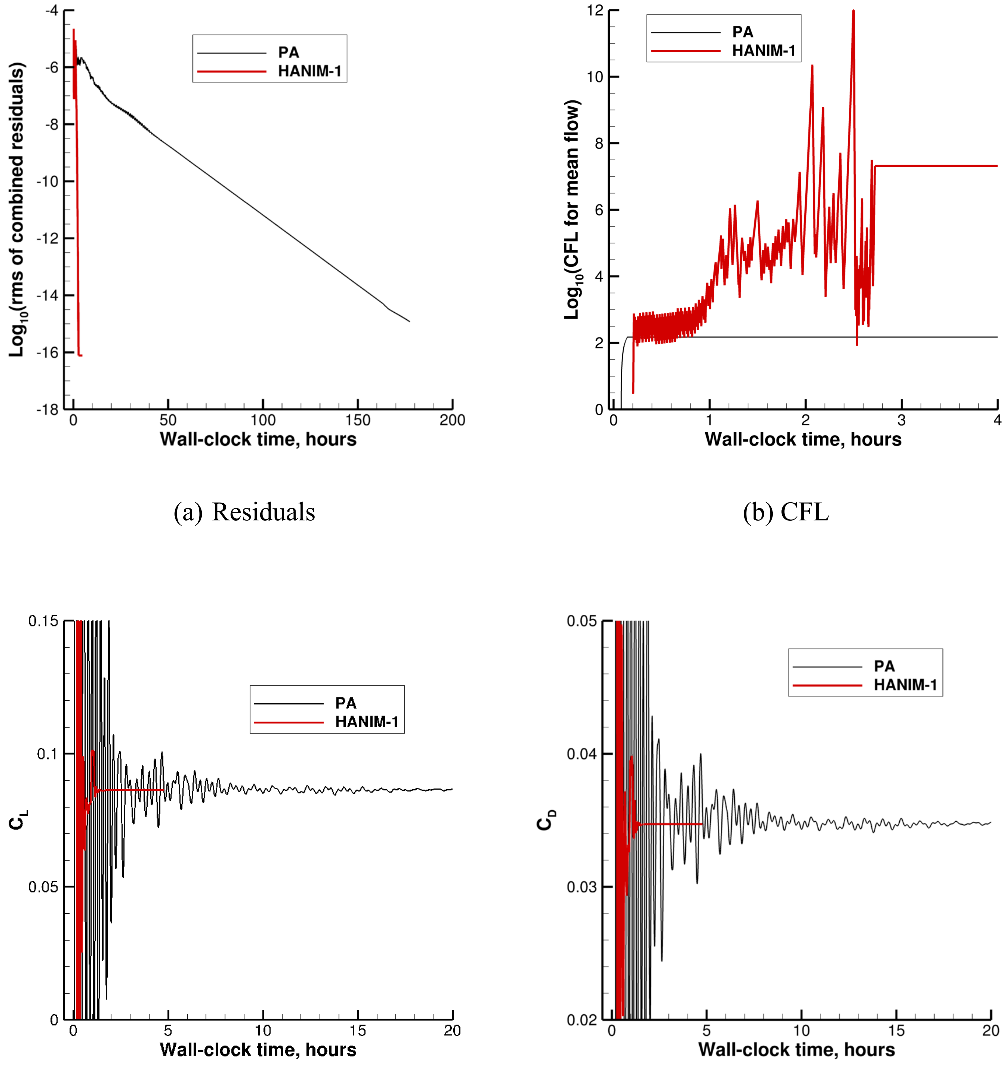
22) SA-model variable in farfield: $\hat{\nu}_{farfield} = 3$ × laminar kinematic viscosity in the freestream.

23) Prandtl number for diffusion fluxes: 0.72 for meanflow, 0.90 for SA model.

Solution convergence is monitored using the residuals and integrated surface quantities. Iterative convergence is demonstrated in a separate figure for each of the two finest grids in the family. Each figure includes plots for variation of four quantities, namely: (1) the combined meanflow and SA-model residuals, (2) CFL, (3) lift coefficient (for the HC and OM6 configurations), and (4) drag coefficient. Variation is shown versus wall-clock time. More quantitative data are presented in tables for all the grids in a family. The preprocessing time is excluded from the wall-clock time shown in the tables. The tables summarize nonlinear iterations, meanflow GS iterations, and the wall-clock time taken by each method to meet the specified convergence criteria. Three types of convergence criteria are considered: (1) reduction of the rms norm of the combined residuals to the target machine-zero level of 10^{-13}, (2) convergence of lift and drag coefficients to six significant digits, and (3) convergence of lift and drag coefficients to an engineering accuracy. Machine-zero residuals and six significant digits of accuracy represent stringent convergence criteria whereas engineering accuracy targets for aerodynamic forces represent lenient convergence criteria. In this study, the engineering accuracy of the lift coefficient is specified as convergence within 0.5% of its final value. The engineering accuracy of the drag coefficient is specified as convergence within one drag count of its final value, which is approximately 0.29%, 0.59%, and 0.1% margin for the HC, OM6-wing, and SD cases, respectively. HANIM speedup relative to the PA method is calculated as a ratio of the wall-clock convergence times of the PA and HANIM solutions. For the residual convergence, HANIM speedup is shown in the tables.

Hemisphere-cylinder (HC) solutions on prismatic-hexahedral grids

Solutions on the five prismatic-hexahedral grids from the HC family (Table 2) are shown below. Although not shown here, the HC configuration solutions on tetrahedral grids are discussed in detail in Ref. [13]. The prismatic-hexahedral grid solutions are computed using the PA method and HANIM-1. The solutions are computed on the NASA Advanced Supercomputing (NAS) facility's Pleiades supercomputer Broadwell nodes consisting of Intel Xeon E5-2680v4 processors. Each Broadwell node consists of two fourteen-core processors per node. A qualitative iterative-convergence assessment of the PA method and HANIM is presented first using the solutions on the two finest grids in the family. Figure 5 illustrates the solution convergence versus wall-clock time on the second-finest grid (Grid 2) of about 78 million cells, whereas, Fig. 6 shows the solution convergence versus wall-clock time on the finest grid (Grid 1) of about 629 million cells. The rms norm of the HANIM-1 solution residuals rapidly drops to the target level. HANIM-1 operates at a much higher CFL as compared to specified maximum CFL of 150 for the PA method, after the solution has evolved beyond the initial transients (after an hour on the second-finest grid and after about 4 hours on the finest grid). Within the transient stage, the HANIM CFL is comparable to the PA CFL. The convergence of aerodynamic coefficients is superior in the HANIM-1 solutions even during the transient stage. Within the plotting accuracy, the lift and drag coefficients in the HANIM-1 solutions appear converged in about 1.5 hours on the second-finest grid and about 6 hours on Grid 1. At these instances, the lift and drag coefficients in the PA solutions exhibit large oscillations with the amplitudes that exceed the entire range of the plotted data.



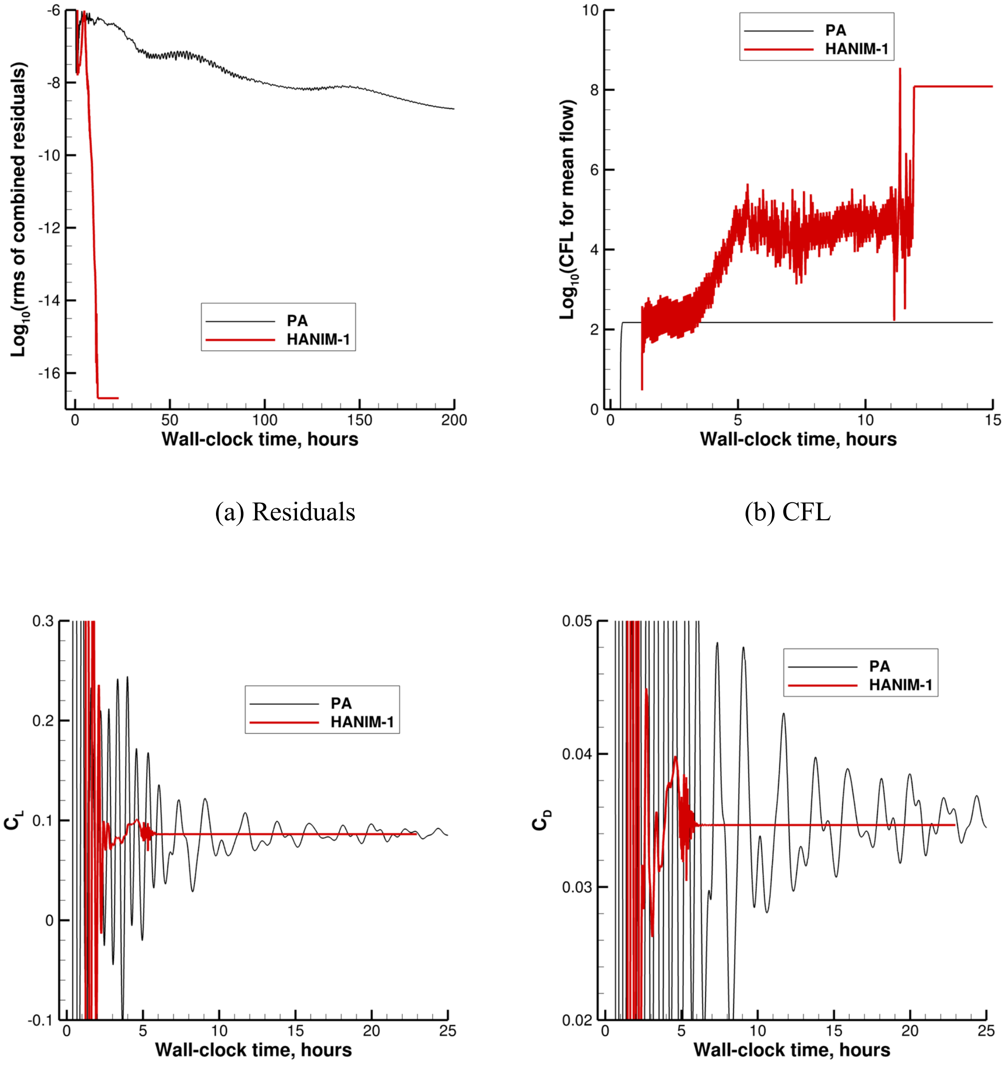
(a) Residuals

(b) CFL

(c) Lift coefficient

(d) Drag coefficient

Fig. 5 HC configuration: iterative convergence on the second-finest prismatic-hexahedral grid with 78,643,200 cells and 336 partitions.



(a) Residuals

(b) CFL

(c) Lift coefficient

(d) Drag coefficient

Fig. 6 HC configuration: iterative convergence on the finest prismatic-hexahedral grid with 629,145,600 cells and 2,520 partitions.

Tables 5-7 summarize the iterative convergence observed in the PA and HANIM solutions on the five prismatic-hexahedral grids. The tables characterize solutions that satisfy the five convergence criteria. Only HANIM-1 solutions satisfy the convergence criteria on all grids. HANIM-1 significantly outperforms the PA method on all grids in all convergence metrics, and the HANIM speedup relative to the PA method increases on finer grids. The HANIM speedup is larger when stringent convergence criteria are applied. HANIM-1 speedup over the PA method across all grids exceeds a factor of 10 for all convergence criteria, including engineering-accuracy criteria. Note, that the PA solution on the finest grid (Grid 1) is suspended before residuals reduce to the machine-zero level and force coefficients converge to six significant digits of accuracy to mitigate the demand on computational resources. After 200 wall-clock hours, in the suspended PA solution, the rms norm of the combined residual has reached 10^{-9} level and engineering accuracy is attained in the lift and drag coefficients. Because HANIM operates at a higher CFL than the PA method, HANIM-1 uses orders-of-magnitude fewer nonlinear iterations to converge. The ratio of the GS preconditioner iterations used by the PA method and HANIM-1 ranges from 13 (Grid 5) to 35 (Grid 2, the finest grid where the PA method has fully converged). Both the PA method and HANIM-1 experience grid-dependent convergence to a varying extent. The number of iterations to convergence grows significantly on finer grids resulting in a significant increase of wall-clock time to convergence despite using similar partition sizes across all grids.

Table 5 Convergence of residuals to machine-zero level for HC prismatic-hexahedral grids.

	Nonlinear iterations		GS iterations, meanflow		Wall-clock time, hours		HANIM
Grid	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1	speedup
5	5,689	188	75,030	5,825	1.41	0.12	11.8
4	11,864	257	159,640	12,780	4.99	0.38	13.2
3	41,504	283	513,870	28,930	21.84	0.86	25.3
2	270,935	442	2,773,240	79,415	136.74	2.23	61.2
1	338,877*	1,289	3,432,675*	234,210	200.91*	9.07	N/A

**PA residual on Grid 1 did not reduce to 10^{-13} level*

Table 6 Convergence of lift for HC prismatic-hexahedral grids.

	Six significant digits				Engineering accuracy			
Grid	Nonlinear iterations		Wall-clock time, hours		Nonlinear iterations		Wall-clock time, hours	
	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1
5	4,000	110	1.03	0.07	1,618	45	0.40	0.03
4	9,045	109	3.83	0.19	1,834	60	0.79	0.11
3	31,103	185	16.74	0.61	7,507	109	4.25	0.39
2	193,256	374	98.04	1.85	33,867	279	18.14	1.13
1	338,877*	1,214	200.91*	8.21	153,837	896	95.09	4.63

**PA lift on Grid 1 did not converge to six significant digits*

Table 7 Convergence of drag for HC prismatic-hexahedral grids.

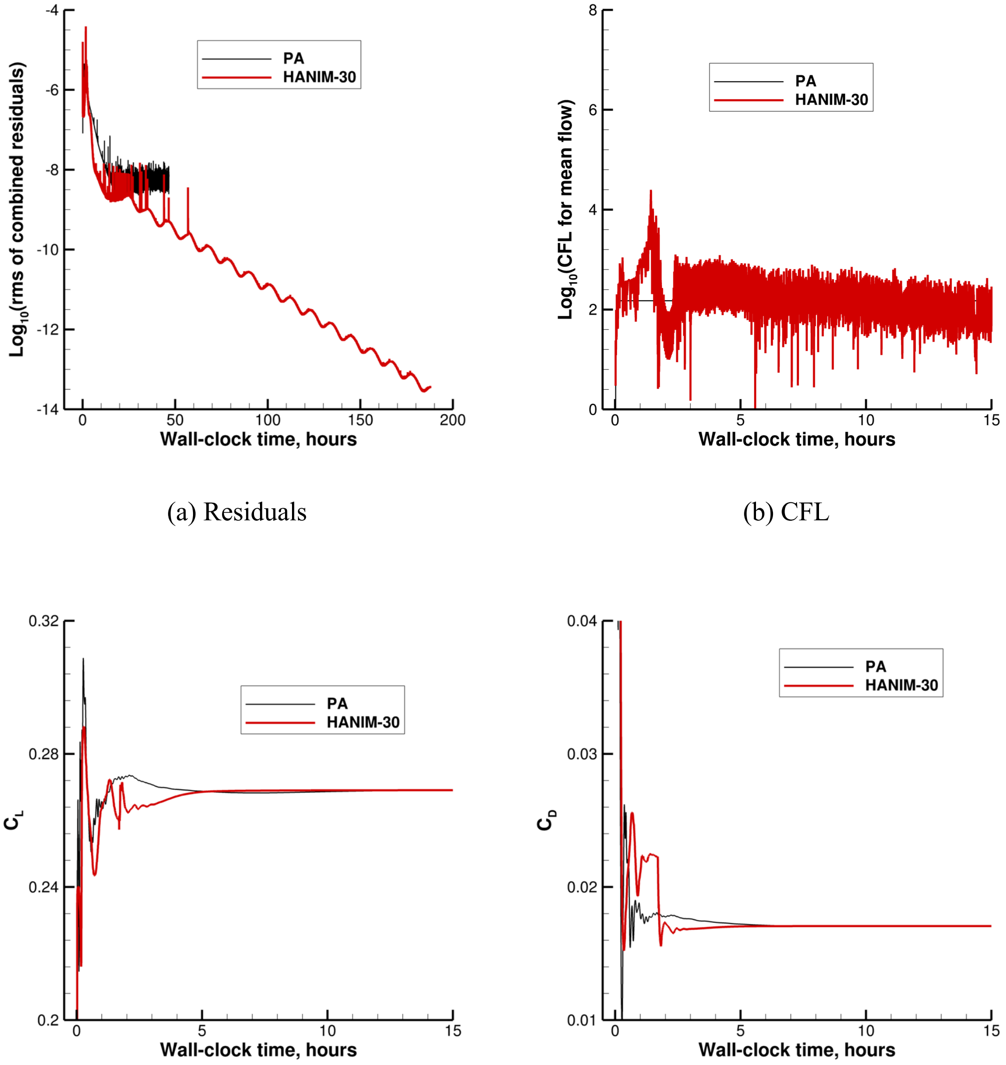
Grid	Six significant digits				Engineering accuracy			
	Nonlinear iterations		Wall-clock time, hours		Nonlinear iterations		Wall-clock time, hours	
	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1
5	3,826	110	0.99	0.07	1,777	47	0.45	0.03
4	9,567	175	4.06	0.26	3,289	61	1.43	0.11
3	30,649	180	16.52	0.60	8,042	115	4.55	0.40
2	207,691	386	105.24	1.93	47,341	299	24.92	1.34
1	338,877*	1,212	200.91*	8.18	168,339	916	103.49	4.87

**PA drag on Grid 1 did not converge to six significant digits*

ONERA M6 (OM6) wing solutions on tetrahedral grids

Solutions on the four grids from the OM6 tetrahedral grid family (Table 3) are shown below. Although not shown here, the OM6 wing solutions on prismatic-hexahedral grids are discussed in Ref. [13]. The tetrahedral-grid solutions are computed using the PA method and HANIM-30. The solutions are computed on the NASA Advanced Supercomputing (NAS) facility’s Pleiades supercomputer Haswell nodes consisting of Intel Xeon E5-2680v3 processors. Each Haswell node consists of two twelve-core processors per node. Figure 7 illustrates the solution convergence on the second-finest grid (Grid 2) of about 45 million cells and Fig. 8 shows the solution convergence on the finest grid (Grid 1) of about 363 million cells.

The PA residuals on the second-finest grid are stalled at an rms norm of 10^{-8}. On the same grid, HANIM is able to converge the rms norm of residuals to the target 10^{-13} level. On the finest grid, the PA solution is not attempted because of an anticipated surge in time to solution for the PA method. On the finest grid, HANIM is unable to reduce the rms norm of residuals below 10^{-9}. On both grids, HANIM operates with a low CFL that oscillates around 100.



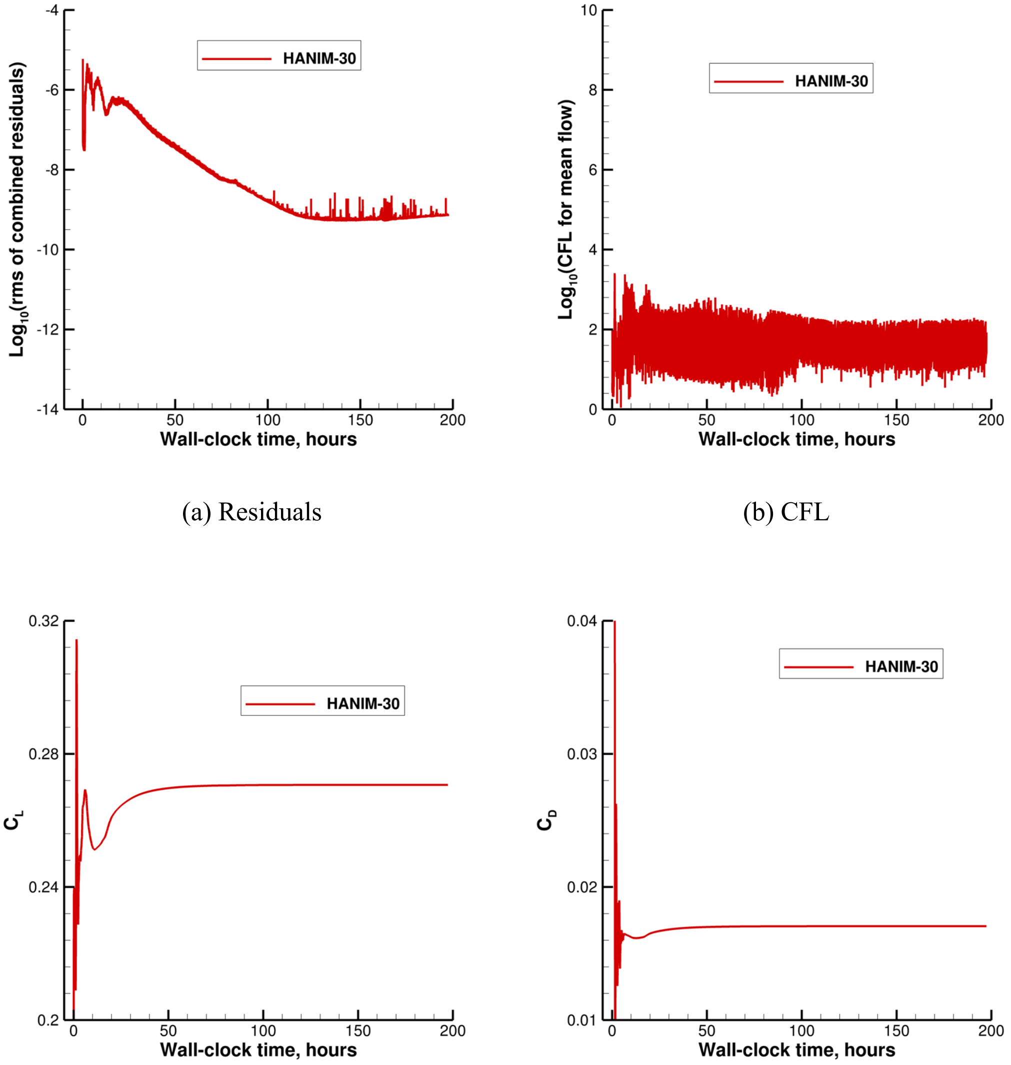
(a) Residuals

(b) CFL

(c) Lift coefficient

(d) Drag coefficient

Fig. 7 OM6 wing: iterative convergence on the second-finest tetrahedral grid with 45,416,448 cells and 192 partitions.



(a) Residuals

(b) CFL

(c) Lift coefficient

(d) Drag coefficient

Fig. 8 OM6 wing: iterative convergence on the finest tetrahedral grid with 363,331,584 cells and 1536 partitions.

Tables 8-10 summarize the iterative convergence observed in the PA and HANIM solutions on the four tetrahedral grids. The tables show that HANIM efficiency is suboptimal. On the two coarsest grids, HANIM performance is similar to that of the PA method. On tetrahedral grids, HANIM’s diminished efficiency may be attributed to an aberrant behavior of USM3D’s simplified preconditioner. It is postulated that on tetrahedral grids, there may be an increased discrepancy between the target nonlinear solution and the solution of the simplified preconditioner equations. This disparity may be impairing the preconditioner’s ability to generate effective search directions for the

GCR linear solver. Consequently, GCR frequently fails to meet residual reduction targets necessitating lower levels of CFL. HANIM's inability to sustain CFL levels that are significantly higher than the level used in the baseline PA solver is one of the possible explanations for the reduced efficiency of HANIM on tetrahedral grids. On prismatic-hexahedral grids, the same simplified preconditioner appears to be adequate in providing effective search directions for the GCR linear solver resulting in better performance of HANIM. Additional studies are needed to understand and address the barriers that hinder iterative convergence on tetrahedral grids.

Table 8 Convergence of residuals to machine-zero level for OM6 wing tetrahedral grids.

	Nonlinear iterations		GS iterations, meanflow		Wall-clock time, hours		HANIM
Grid	PA	HANIM-30	PA	HANIM-30	PA	HANIM-30	speedup
4	16,015	9,438	166,165	211,125	4.42	5.20	0.85
3	63,475	19,332	646,545	538,500	22.70	15.34	1.48
2	128,142*	99,296	1,192,465*	2,200,540	46.57*	170.03	N/A
1	N/A	147,169*	N/A	2,764,785*	N/A	197.30*	N/A

**PA residual on Grid 2 and HANIM-30 residual on Grid 1 did not reduce to 10^{-13} level*

Table 9 Convergence of lift for OM6 wing tetrahedral grids.

	Six significant digits				Engineering accuracy			
Grid	Nonlinear iterations		Wall-clock time, hours		Nonlinear iterations		Wall-clock time, hours	
	PA	HANIM-30	PA	HANIM-30	PA	HANIM-30	PA	HANIM-30
4	9,789	5,488	2.72	3.27	2,391	2,049	0.71	1.58
3	25,373	11,937	9.16	9.79	3,400	3,380	1.35	3.61
2	128,142*	36,051	46.57*	57.86	7,367	2,799	3.09	4.22
1	N/A	147,169*	N/A	197.30*	N/A	48,482	N/A	45.25

**PA lift on Grid 2 and HANIM-30 lift on Grid 1 have not converged to six significant digits*

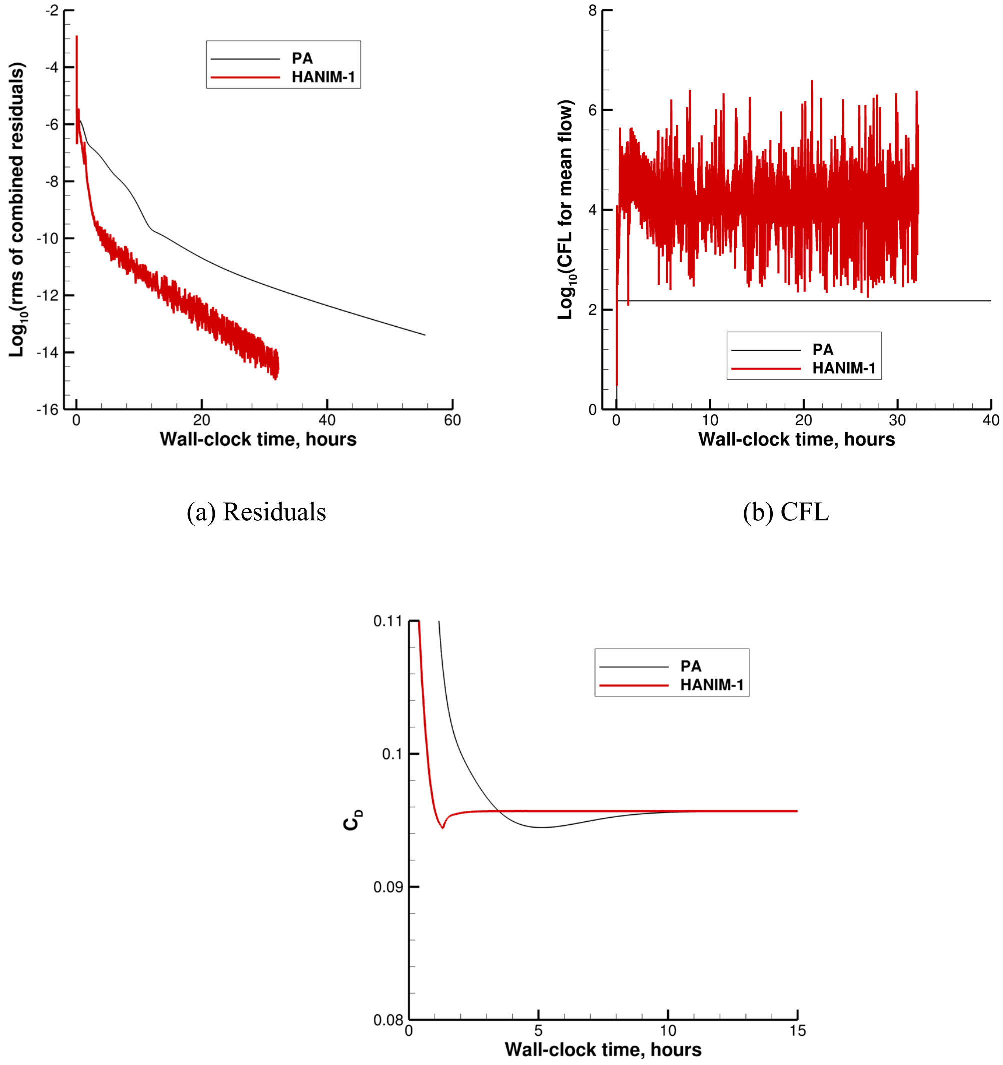
Table 10 Convergence of drag for OM6 wing tetrahedral grids.

Grid	Six significant digits				Engineering accuracy			
	Nonlinear iterations		Wall-clock time, hours		Nonlinear iterations		Wall-clock time, hours	
	PA	HANIM-30	PA	HANIM-30	PA	HANIM-30	PA	HANIM-30
4	9,783	5,477	2.72	3.27	1,505	1.909	0.47	1.51
3	24,165	10,675	8.73	8.87	4,663	3,146	1.80	3.44
2	128,142*	16,504	46.57*	24.52	13,658	2,623	5.43	4.02
1	N/A	147,169*	N/A	197.30*	N/A	45,100	N/A	42.36

**PA drag on Grid 2 and HANIM-30 drag on Grid 1 have not converged to six significant digits*

Square-duct (SD) solutions on hexahedral grids

Solutions on the four hexahedral grids from the SD family (Table 4) are computed using the PA method and HANIM-1. The solutions are computed on the NASA Advanced Supercomputing (NAS) facility's Pleiades supercomputer Haswell nodes consisting of Intel Xeon E5-2680v3 processors. The finest-grid HANIM-1 solution is computed with the initial CFL = 1 for the meanflow and the turbulence model. A qualitative iterative-convergence assessment of the PA method and HANIM is presented using the solutions on the two finest grids in the family. Figure 9 illustrates iterative convergence on the second-finest grid (Grid 2) of about 198 million cells, whereas, Fig. 10 shows iterative convergence on the finest grid (Grid 1) of about 1.6 billion cells. The figures show residual convergence, CFL variation, and convergence of the drag coefficient versus the wall-clock time. HANIM operates at a much higher CFL as compared to a specified maximum CFL of 150 for the PA method. HANIM converges the rms norm of the residual to the target level of 10^{-13} on all grids. The PA solution on the finest grid was terminated after about 115 hours; the rms norm of residuals at termination was close to 10^{-11}. Convergence rates of both HANIM and PA methods slow down after the rms norm of residuals reduced to the level of 10^{-10}. The drag coefficients of HANIM solutions converge significantly faster than the drag coefficients of the PA solutions.

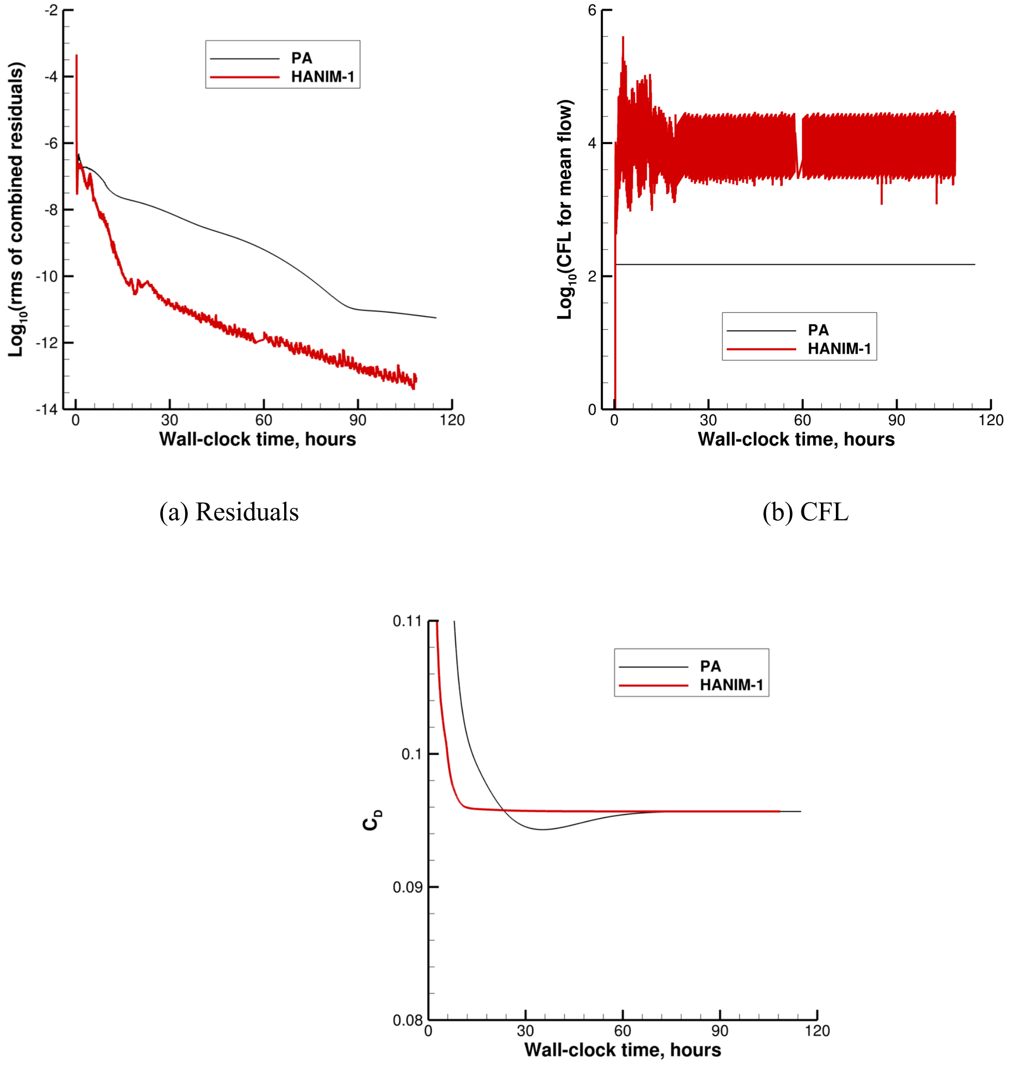


(a) Residuals

(b) CFL

(c) Drag coefficient

Fig. 9 SD configuration: iterative convergence on the second-finest hexahedral grid with 198,180,864 cells and 1536 partitions.



(a) Residuals

(b) CFL

(c) Drag coefficient

Fig. 10 SD configuration: iterative convergence on the finest hexahedral grid with 1,585,446,912 cells and 6144 partitions.

Tables 11 and 12 summarize the iterative convergence observed in the PA and HANIM-1 solutions. On all grids, HANIM uses fewer nonlinear iterations but more GS iterations. For all convergence criteria, the HANIM speedup factor increases on finer grids. HANIM outperforms the PA method on all grids for stringent convergence criteria but needs more time to converge to engineering accuracy on the coarsest grid (Grid 4). The lack of performance improvement on the coarsest grid is in part explained by an excessive time HANIM spends in the initial first-order iterations. Table 13 shows the wall-clock time spent on the initial iterations with first-order meanflow inviscid fluxes.

For the PA method, the first-order iterations take a negligible portion of the solution time. HANIM spends a significant portion of solution time in the first-order iterations, especially on coarser grids. On the two coarser grids, HANIM spends an hour or more in first-order iterations, which is most of the time it takes to converge solutions to engineering accuracy. In fact, after switching discretization to full second order on Grids 3 and 4, HANIM converges to engineering accuracy in less than 50 nonlinear iterations that take 7.2 and 6 minutes, respectively. The PA method spends 10 minutes or less in first-order iterations on these grids. The striking difference between times spent in the first-order iterations is explained by the difference in the preconditioner iterations. The PA method uses a fixed number (15) of GS iterations, while the HANIM preconditioner requires residual reduction by an order of magnitude. Such a strict tolerance on preconditioner convergence is probably counterproductive and should be relaxed, at least in first-order iterations.

Table 11 Convergence of residuals to machine-zero level for supersonic SD hexahedral grids.

	Nonlinear iterations		GS iterations, meanflow		Wall-clock time, hours		HANIM
Grid	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1	speedup
4	15,949	1,102	239,235	496,870	5.25	4.45	1.18
3	42,866	1,576	642,990	882,440	14.64	8.43	1.74
2	118,531	2,877	1,777,965	1,835,285	49.52	19.25	2.57
1	140,577*	6,332	2,108,655*	4,101,940	114.72*	93.54	N/A

**PA residual on Grid 1 did not reduce to 10^{-13} level*

Table 12 Convergence of drag for SD hexahedral grids.

	Six significant digits				Engineering accuracy			
Grid	Nonlinear iterations		Wall-clock time, hours		Nonlinear iterations		Wall-clock time, hours	
	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1	PA	HANIM-1
4	9,064	462	2.98	1.77	2,138	247	0.70	0.97
3	28,253	1,010	9.70	5.43	6,826	243	2.33	1.24
2	83,121	2,426	34.81	16.21	23,029	338	9.77	2.07
1	140,577*	6,892	114.72*	101.25	81,239	1,587	66.29	21.89

**PA drag on Grid 1 did not converge to six significant digits*

Table 13 First-order nonlinear iterations for SD hexahedral grids.

Grid	Nonlinear iterations		Wall-clock time, hours	
	PA	HANIM-1	PA	HANIM-1
4	500	200	0.16	0.87
3	500	200	0.17	1.12
2	500	200	0.24	1.26
1	500	200	0.65	2.80

VI. Conclusions

This paper reports on further algorithmic enhancements of USM3D. Previously, a hierarchical adaptive nonlinear iteration method (HANIM) had been implemented in the NASA unstructured-grid cell-centered mixed-element flow solver, USM3D, to improve the robustness, efficiency, accuracy, and automation of RANS solutions. New contributions of this paper are the development of a line-generation algorithm suitable for general unstructured grids, a cost-effective implementation of the Generalized Conjugate Residual (GCR) method with many search directions, and efficiency assessment of HANIM on substantially larger grids for three-dimensional benchmark turbulent-flow cases. The benchmark cases are a separated subsonic flow over a hemisphere-cylinder (HC) configuration, a transonic flow with shocks around the ONERA M6 (OM6) wing, and a supersonic flow through a square-duct (SD) configuration. The USM3D solutions are computed on families of prismatic-hexahedral, tetrahedral, and hexahedral grids of up to 1.6 billion cells. The Spalart-Allmaras negative (SA-neg) turbulence model is used for the HC and OM6-wing computations; the SA-neg-QCR2000 turbulence model is used in the SD computations.

For the efficiency studies, the iterative convergence of the mixed-element USM3D HANIM and the PA method is compared in terms of the wall-clock time to solution, number of nonlinear iterations, and number of cumulative GS iterations. Five different convergence criteria are considered: residual reduction below 10^{-13} representing the machine-zero level, lift convergence to six significant digits, lift convergence to engineering accuracy ($\pm 0.5\%$ of the final value), drag convergence to six significant digits, and drag convergence to engineering accuracy (± 1 count around the final value).

HANIM-1 (one search direction is applied by the GCR method) with a hybrid point- and line-implicit preconditioner is used on prismatic-hexahedral grids. HANIM-30 (up to 30 search directions are applied by the GCR method) with a point-implicit preconditioner is used on tetrahedral grids. HANIM-1 with a point-implicit preconditioner is used on hexahedral grids. The PA method with a point-implicit preconditioner is used on all grids.

For prismatic-hexahedral grid solutions, HANIM-1 operates at a much higher Courant-Friedrichs-Lewy (CFL) number than the PA method. For the HC configuration, the HANIM speedup over the PA method across all grids and solution convergence metrics ranges from five (engineering accuracy for drag on a coarse grid) to 62 (six-significant-digits accuracy for lift on a fine grid). HANIM speedup factor increases on finer grids and exceeds 7.5 for all metrics on the two finest grids. Only HANIM reduces the solution residuals to machine-zero level and converges the forces to six digits of accuracy on the finest HC prismatic-hexahedral grid of 629 million cells.

For the OM6 wing tetrahedral-grid solutions, HANIM efficiency is not better than the efficiency of the PA method. The degraded efficiency is indicated by HANIM CFL, which is lower than the prescribed CFL of the PA method. On the finest tetrahedral grid, neither solver is able to converge residuals to a machine-zero level.

HANIM uses a Krylov method to solve linearization of a second-order discretization of the RANS equations. The efficiency of a Krylov method is critically dependent on the quality of preconditioner. The current study highlights the fact that the USM3D preconditioner is effective on prismatic-hexahedral grids but less effective on tetrahedral grids. This is exemplified by superior HANIM performance on prismatic-hexahedral grids in an absolute sense (faster time to solution) and a relative sense (an order of magnitude speedup over the PA method). Additional investigations are needed to understand and remove barriers hindering faster iterative convergence on tetrahedral grids.

For all hexahedral grid solutions for the supersonic SD configuration, HANIM-1 operates at a much higher CFL than the PA method. However, for convergence to machine-zero residuals and six digits of drag accuracy, the HANIM speedup over the PA method is modest, about a factor of two. The speedup is higher for convergence to engineering accuracy on finer grids. The engineering-accuracy speedup could have been higher, especially on coarser grids, had HANIM been started with second-order iterations from the beginning. This conclusion is drawn based on the observations that HANIM consumes an excessive time in 200 initial first-order iterations, and the engineering accuracy target is met within a few dozen second-order iterations. On the finest grid of 1.6 billion cells, only HANIM achieves convergence in stringent accuracy metrics (machine-zero residuals and six-digit accuracy of the drag coefficient). On that grid, HANIM reduced the residuals to the machine-zero level in about 90 hours using 6,144 partitions.

Acknowledgments

The work reported in this article was funded by the National Aeronautics and Space Administration. Specifically, the Transformative Tools and Technologies project of the Transformative Aeronautics Concepts Program and the Airspace Technology Demonstrations project of the Airspace Operations and Safety Program within the NASA Aeronautics Research Mission Directorate funded the work. The third author was supported by the NASA Langley Research Center under the cooperative agreement 80LARC17C0004 with the National Institute of Aerospace. All computations were performed on the NASA Advanced Supercomputing facility's Pleiades supercomputer.

VII. References

[1] Pandya, M. J., Frink, N. T., Abdol-Hamid, K. S., Samareh, J. A., Parlette, E. B., and Taft, J. R., "Enhancements to TetrUSS for NASA Constellation Program," *Journal of Spacecraft and Rockets*, Vol. 49, No. 4, 2012, pp. 617-631. https://doi.org/10.2514/1.A32089

[2] Bauer, S. X., Krist, S. E., and Compton, W. B. "Generation of the Ares I-X Flight Test Vehicle Aerodynamic Data Book and Comparison to Flight," AIAA Paper 2011–0011. https://doi.org/10.2514/6.2011-11

[3] Abdol-Hamid, K. S., Ghaffari, F., and Parlette, E. B., "Ares I Vehicle Computed Turbulent Ascent Aerodynamic Data Development and Analysis," *Journal of Spacecraft and Rockets*, Vol. 49, No. 4, 2012, pp. 596-608. https://doi.org/10.2514/1.A32112

[4] Warwick, G., "Lockheed Martin Refines Hybrid Wing-Body Airlifter Concept," Aviation Week & Space Technology, February 17, 2014, p.40.

[5] Wick, A. T., Hooker, J. R., and Zeune, C. H., "Integrated Aerodynamic Benefits of Distributed Propulsion," AIAA Paper 2015-1500. https://doi.org/10.2514/6.2015-1500

[6] Hooker, J. R., Wick, A., Zeune, C., and Agelastos, A., "Over Wing Nacelle Installations for Improved Energy Efficiency," AIAA Paper 2013-2920. https://doi.org/10.2514/6.2013-2920

[7] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA/CR-2014-218178, 2014.

[8] Pandya, M. J., Frink, N. T., Ding, E., and Parlette, E. B., "Toward Verification of USM3D Extensions for Mixed Element Grids," AIAA Paper 2013-2541. https://doi.org/10.2514/6.2013-2541

[9] Pandya, M. J., Diskin, B., Thomas, J. L., and Frink, N. T., “Improved Convergence and Robustness of USM3D Solutions on Mixed Element Grids,” *AIAA Journal*, Vol. 54, No. 9, 2016, pp 2589-2610. https://doi.org/10.2514/1.J054545

[10] Spalart, P. R. and Allmaras, S. R., “A One-Equation Turbulence Model for Aerodynamic Flows,” *Recherche Aerospatiale*, No. 1, 1994, pp. 5-21.

[11] Allmaras, S. R., Johnson, F. T., and Spalart, P. R., “Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model,” ICCFD7-1902, Seventh International Conference on Computational Fluid Dynamics, Big Island, Hawaii, 2012.

[12] Pandya, M. J., Diskin, B., Thomas, J. L., and Frink, N. T., “Assessment of USM3D Hierarchical Adaptive Nonlinear Iteration Method Preconditioners for Three-Dimensional Cases,” *AIAA Journal*, Vol. 55, No. 10, 2017, pp 3409-3424. https://doi.org/10.2514/1.J055823

[13] Pandya M. J., Jespersen D. C., Diskin B., Thomas J. L., and Frink N. T., “Accuracy, Scalability, and Efficiency of Mixed-Element USM3D for Benchmark Three-Dimensional Flows,” AIAA Paper 2019-2333. https://doi.org/10.2514/6.2019-2333

[14] Tsieh, T., “An Investigation of Separated Flow About a Hemisphere Cylinder at 0- to 19-Deg Incidence in the Mach Number Range of 0.6 to 1.5,” AEDC-TR-76-112, 1976.

[15] Schmitt, V. and Charpin, F., “Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers,” AGARD AR-138, May 1979.

[16] Davis, D. O. and Gessner, F. B., "Further Experiments on Supersonic Turbulent Flow Development in a Square Duct," *AIAA Journal*, Vol. 27, No. 8, 1989, pp. 1023-1030. https://doi.org/10.2514/3.10216

[17] Spalart, P. R., "Strategies for Turbulence Modelling and Simulation," *International Journal of Heat and Fluid Flow*, Vol. 21, 2000, pp. 252-263. https://doi.org/10.1016/S0142-727X(00)00007-2

[18] Roe, P., “Characteristic Based Schemes for the Euler Equations,” *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337-365. https://doi.org/10.1146/annurev.fl.18.010186.002005

[19] Van Leer, B. “Flux-Vector Splitting for the Euler Equations,” ICASE Report 82-30, 1982.

[20] Einfeldt, B., “On Godunov-Type Methods for Gas Dynamics,” *SIAM Journal on Numerical Analysis*, Vol. 25, No. 2, 1988, pp. 294-318. https://doi.org/10.1137/0725021

[21] Toro, E. F., Spruce, M., and Speares, W., “Restoration of the Contact Surface in the HLL-Riemann Solver,” *Shock Waves*, Vol. 4, Issue 1, 1994, pp. 25–34. https://doi.org/10.1007/BF01414629

[22] Mitchell, C. R., “Improved Reconstruction Schemes for the Navier-Stokes Equations on Unstructured Meshes,” AIAA Paper 1994-0642. https://doi.org/10.2514/6.1994-642

[23] Frink, N. T., “Tetrahedral Unstructured Navier-Stokes Method for Turbulent Flows,” *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1975-1982. https://doi.org/10.2514/2.324

[24] Knoll, D. A. and Keyes, D. E., “Jacobian-free Newton–Krylov Methods: A Survey of Approaches and Applications,” *Journal of Computational Physics*, 193 (2004) 357–397. https://doi.org/10.1016/j.jcp.2003.08.010

[25] Van der Vorst, H. A. and Vuik, C. “GMRESR: A Family of Nested GMRES Methods,” *Numerical Linear Algebra with Applications*, Vol. 1, pp. 369-386, 1994. https://doi.org/10.1002/nla.1680010404

[26] Lucas, P., van Zuijlen, A. H., and Bijl, H., “Fast Unsteady Flow Computations with a Jacobian-free Newton-Krylov Algorithm," *Journal of Computational Physics*, Vol. 229, No. 24, 2010, pp. 9201-9215. https://doi.org/10.1016/j.jcp.2010.08.033

[27] Ceze, M. and Fidkowski, K., “A Robust Adaptive Solution Strategy for High-Order Implicit CFD solvers," AIAA Paper 2011-3696. https://doi.org/10.2514/6.2011-3696

[28] Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., Johnson, F. T., Melvin, R. G., Tinoco, E. N., Venkatakrishnan, V., Wigton, L. B., and Young, D. P., “Algorithm Issues and Challenges Associated with the Development of Robust CFD Codes,” In: Variational Analysis and Aerospace Engineering, Springer Optimization and Its Applications, Vol. 33, Springer, New York, NY, 2009, pp. 1-19. https://doi.org/10.1007/978-0-387-95857-6_1

[29] Nishikawa, H., and Diskin, B., “Customized Grid Generation Codes for Benchmark Three-Dimensional Flows,” AIAA Paper 2018-1101. https://doi.org/10.2514/6.2018-1101

[30] Diskin, B., Anderson, W. K., Pandya, M. J., Rumsey, C. L., Thomas, J. L., Liu, Y., and Nishikawa, H., “Grid Convergence for Three-Dimensional Benchmark Turbulent Flows,” AIAA Paper 2018-1102. https://doi.org/10.2514/6.2018-1102

[31] Frink, N. T., “Recent Progress toward a Three-Dimensional Unstructured Navier-Stokes Flow Solver,” AIAA Paper 1994-0061. https://doi.org/10.2514/6.1994-61