

# Estimation of Captive Flight Loads Using Grid-Free Euler Solver

Vaibhav Shah,\* Rohit R. More,\* S. Srinivasa Raju,\* K. Anandhanarayanan,† R. Krishnamurthy,\*  
and Debasis Chakraborty\*

*Defence Research and Development Laboratory, Hyderabad 500 258, India*

DOI: 10.2514/1.C031261

**The structural integrity of a store and its attachments to a fighter aircraft has to be verified at various aircraft maneuvering conditions in its flight envelope for the clearance of the store to integrate with aircraft. The captive loads of a flight vehicle mounted on a fighter aircraft have been estimated at various Mach numbers, angles of attack, and sideslip angles using an indigenously developed 3-D grid-free Euler solver. The point distributions required to apply grid-free solver are obtained using chimera cloud technique, and the connectivity required for the solver is obtained using gradient search method. Forces and moments acting on the fighter aircraft with launchers and flight vehicles have been estimated using computational fluid dynamics simulations at different flow conditions, and load distributions on flight vehicle have been obtained. The aerodynamic loads obtained from computational fluid dynamics calculations form the basis of the flight testing.**

## I. Introduction

**I**NTEGRATION of a new weapon to a highly agile aircraft is an exigent task, as the weapon and its supporting systems should structurally endure during stern maneuvers of the aircraft. Fighter aircraft has maneuvering capability of about 9 g and, therefore, the support systems of flight vehicles (launcher, launcher rail, launch shoe, and adapter) on the fighter aircraft should be designed to withstand the loads experienced by the vehicle at critical maneuvering conditions. Guidelines for weapon-aircraft integration are provided through military specification MIL-STD-8591 [1]. The specification provides weapon design load requirements for unusual flight conditions that sometime arise on specific aircraft. It is based on using actual or predicted aircraft flight performance data to examine the full flight envelope and generate consistent set of inertial and aerodynamic loads [2]. In the present work, aerodynamic characterization of the flight vehicle mounted on a fighter aircraft has been carried out using an in-house developed 3-D grid-free Euler solver [3]. Further, the numerical simulations are carried out to estimate the captive loads acting on the flight vehicle integrated with fighter aircraft at specified flow conditions and the maximum loading conditions are identified. The load distribution is extracted at that condition to verify the structural integrity of the flight vehicle and supporting systems.

Grid generation around a complex fighter aircraft is the most time consuming task in the numerical simulation and presence of flight vehicle poses a formidable challenge. A grid-free q-LSKUM [3] (entropy variables-based least-squares kinetic upwind method) Euler solver has been used for the computational fluid dynamics (CFD) analysis. The grid-free solver requires a cloud of points around the configuration and a set of neighbors around each point. The cloud of points was obtained by overlapping grids around fighter aircraft and flight vehicles. CAD models of flight vehicle and fighter aircraft were imported, repaired, and cleaned up to obtain watertight solid models suitable for grid generation. Simple unstructured grids around fighter aircraft and flight vehicles are generated independently and then the

grids are overlapped to get the distribution of points around the full configuration. An efficient preprocessor [4] has been developed and applied to generate the connectivity. A parallel version of the 3-D grid-free Euler solver q-LSKUM has been used to obtain the aerodynamic characteristics of the aerospace vehicle configuration.

Inviscid steady flow simulation for fighter aircraft has been carried out using q-LSKUM code. The simulations at various Mach numbers, angles of attack, and sideslip angles have been carried out to obtain aerodynamic forces and moments acting on the integrated fighter aircraft with flight vehicles. The captive loads on all the four flight vehicles are computed at specified critical flow conditions.

## II. Geometric Details and Grid Generation

The full geometry consists of fighter aircraft and four flight vehicles placed at two different locations on either side of the wing as shown in Fig. 1. The full geometry, including canards, wings with slats and ailerons, vertical and horizontal tails, and air-intakes, is considered for the simulation. The rail launcher and adapter to accommodate flight vehicles are also attached to the fighter aircraft wing at proper positions. The isometric view of the aircraft geometry showing all the geometric features is shown in Fig. 2. The aircraft geometry has been imported in ParaSolid format to generate a tetrahedral grid using a commercial grid generator. Making use of the symmetry of the configuration, only half of the geometry is considered for grid generation. The gaps and surface overlaps are identified and repaired to create a watertight surface for generating a grid with optimal size. The physical small gaps are closed to avoid generating tiny cells which would lead to a large number of cells. The flight vehicle has cruciform low-aspect-ratio wings, fins, two sets of wire-tunnels, and three launch shoes. The flight vehicle geometry is also repaired similar to aircraft geometry and placed at appropriate positions as shown in Fig. 3.

The grids around the fighter aircraft and flight vehicle are generated separately. A large computational domain around the fighter aircraft is considered to cater subsonic and supersonic flows. The computational domain consists of inflow and far-field boundaries at nearly a length of the aircraft from its nose and outflow boundary at twice the length of the aircraft from the nose. Full tetra mesh has been generated around the fighter aircraft configuration with proper clustering at desired regions. The volume grids are generated with sizes 0.77, 2.05, and 3.5 million points to carry out the grid convergence study.

Relatively, a smaller computational domain has been used for the flight vehicle configuration as it forms a subregion in the large computational domain around fighter aircraft. Typical grids with 0.84 and 1.33 million points have been generated around flight vehicles. The grids on symmetry planes of the integrated fighter

Received 4 October 2010; revision received 18 January 2011; accepted for publication 13 March 2011. Copyright © 2011 by V. Shah, R. More, S. Srinivasa Raju, K. Anandhanarayanan, R. Krishnamurthy, and D. Chakraborty. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0021-8669/11 and \$10.00 in correspondence with the CCC.

\*Computational Fluid Dynamics Division, Directorate of Computational Dynamics.

†Computational Fluid Dynamics Division, Directorate of Computational Dynamics; kanand\_cfd@yahoo.com (Corresponding Author).

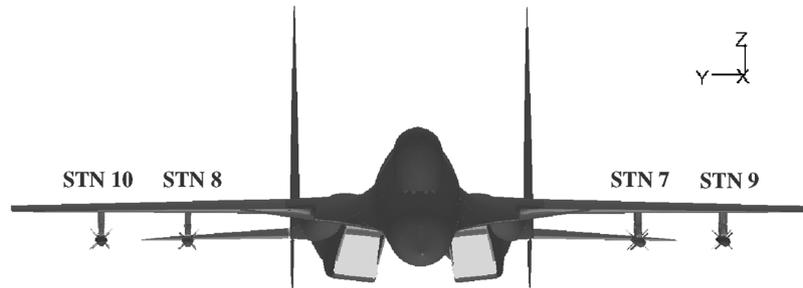


Fig. 1 Front view of fighter aircraft with flight vehicles.

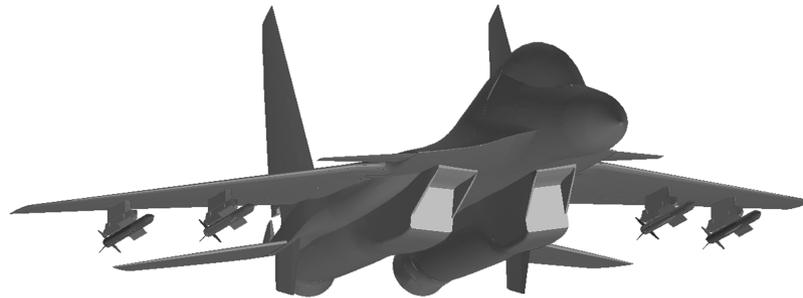


Fig. 2 Isometric view of fighter aircraft with flight vehicles.

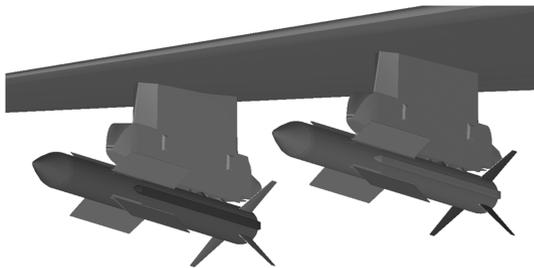


Fig. 3 Flight vehicles integrated with wing of fighter aircraft.

aircraft and flight vehicles are shown in Fig. 4. These grids are overlapped to get the distribution of points within the domain. A set of neighbors around each point is required to be specified from the distribution of points to apply the grid-free solver. The connectivity set is generated by a preprocessor. The method of generating data structure is described in the following section.

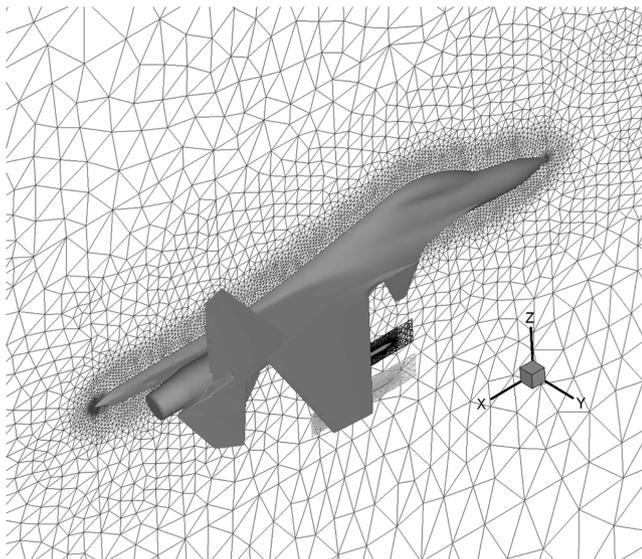


Fig. 4 Grids on symmetry planes of fighter aircraft and flight vehicles.

### III. Generation of Data Structure

The grid-free solver requires just a distribution of points and a set of neighbors, called connectivity, around each point. The distribution of points can be obtained by two methods, namely simple cloud method and chimera cloud method. In simple cloud method, the point distribution can be obtained using the grid generated around the body and leaving the grid lines. In chimera cloud method, the complex geometry is subdivided into geometrically simpler shapes and clouds of points are generated around these individual components. The simple clouds are then overlapped to get the distribution of points over the entire computational domain. The chimera cloud method basically uses grids to get the distribution of points and connectivity, but the present method is different and efficient from the chimera grid method [5].

An efficient preprocessor [3] has been developed to generate the connectivity using overlapped structured grids, and recently the preprocessor has been extended to generate the connectivity using overlapped unstructured grids. The preprocessor accepts multiple unstructured grids and overlaps the unstructured grids as per the geometry position. Because of overlapping of multiple grids, certain nodes of one grid may lie inside other components and these nodes should be removed or blanked. This procedure is generally known as hole cutting. In our approach the surface grids are used for hole cutting, i.e., the nodes that lie inside the solid bodies are identified using the triangles that bound the surface. The advantage of the method is that the hole cutting is exact and no extra human interaction is required to define the hole-surface. There are generally four possible types of cells due to overlapping of multiple grids as shown in Fig. 5. The first type of cells lie completely inside the solid body. The second type of cells is cut by the cutting surface but some nodes of those cells are inside the body and some nodes are outside. The third type of cells are also cut cells that are cut by a thin surface, but all the nodes are outside the solid body, i.e., the cells pierce through the body. The fourth type of cells is fully outside of the cutting surface. Handling of the first and fourth types of cells in our procedure is the same as that of the chimera grid method. The nodes of cells of the first type are removed from the computation and the nodes are referred to as blanked nodes. The nodes of cells of the fourth type are available for computation; the nodes are referred to as field nodes and the cells are called field cells. In the chimera grid method the second and third type of cells are also blanked and the field cells adjacent to the blanked cells are flagged as fringe cells. The flowfield of these fringe cells is not computed, but interpolated from the field cells of the other

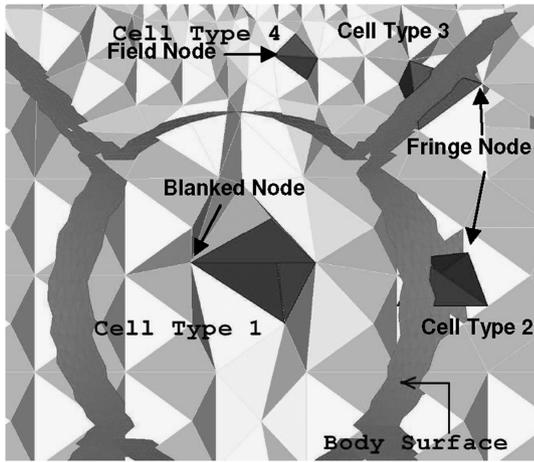


Fig. 5 Four possible types of cells in the overlapped grids.

overlapping grid. This condition implies that some field cells should be available between the two hole-cutting surfaces in both the grids for interpolation of flowfield for fringe cells of other grid, which may be difficult for the bodies separated by thin gaps like aircraft launcher and store or missile body with the deflected fins. In our approach, the nodes of cells of the second type that are outside are also used in the computation, and those nodes are called fringe nodes whose connectivity should include neighbors from the overlapping grids. The nodes inside the solid body are removed from the neighbor list of the fringe nodes. The nodes of cells of third type are also called fringe nodes, but the nodes on the same side of the surface are only included in their respective connectivity list. Unlike chimera grid method, the fringe nodes also take part in computation and therefore no interpolation is required. Major advantages of this important difference are 1) the same upwind discretized equations, which satisfy local characteristics of the flow, are solved at these points, 2) there is no time lag in the unsteady computations between various clouds, 3) uniform treatment for all the points, which simplifies parallelization and implementation of convergence acceleration methods, and 4) most important, a very thin gap can be modeled without much consideration on the grid size of the component grids. After classifying the nodes as blanked, field, and fringe, the connectivity is generated for each node. The blanked node does not require any connectivity; neighbors for field nodes are obtained from the cell connectivity information and for the fringe nodes, neighbors from the same side of the hole-cutting surface of the same grid and some nodes from the overlapping grids are also included. The neighbor nodes in the overlapping grids are the vertices of the cell in the overlapping grid that contains the fringe node. The overall connectivity generation process involves identification of cutcell, classification of cutcells and solid cells, classification of solid nodes,

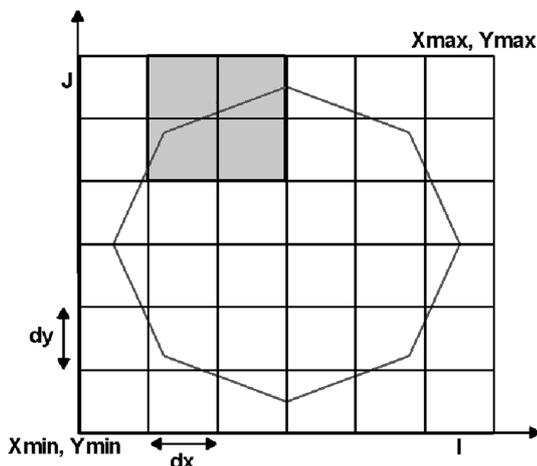


Fig. 6 Intersecting bins of component faces.

field nodes, and fringe nodes, and connectivity generation for the above nodes.

**A. Identification of Cutcell**

Each component is associated with a bounding box as shown in Fig. 6 (for clarity, a 2-D example is shown) and the bounding box is divided into small uniform bins. Each bin is identified with indices  $(i, j)$ . A bin that contains a point  $(x, y)$  can be easily located using the bounding box size  $(x_{min}, x_{max}, y_{min}, y_{max})$  and the number of bins along each coordinate direction  $(I, J)$ . For example

$$i = \frac{x - x_{min}}{dx}; \quad dx = \frac{x_{max} - x_{min}}{I}$$

$$j = \frac{y - y_{min}}{dy}; \quad dy = \frac{y_{max} - y_{min}}{J}$$

The surface triangles of the component are stored in the bins that intersect the bounding box of the triangles. Typical intersecting bins (shaded) for an edge are also shown in the figure.

In the cutcell identification procedure, each component grid is checked for possible intersection with other components using respective bounding box intersections. Should there be a possible intersection, then each cell of the grid is checked for possible intersection with the component. Checking each cell of a grid with each surface triangle of possible intersecting components is time consuming. Instead, each cell is checked only with the triangles stored in the bins that are intersecting with the bounding box of the cell. A cell can intersect with a triangle only if any edge of the cell intersects the triangle. It leads to finding the intersection of a triangle with an edge. Typical intersection of an edge and a triangle is shown in Fig. 7. The parametric representation of the edge and triangle are

$$Q = Q_1 + (1 - s)Q_2 \quad P = P_1 + u(P_2 - P_1) + v(P_3 - P_1)$$

The intersection point can be obtained by equating the above two parametric equations. It constitutes a system of three equations and three unknowns namely,  $s, u,$  and  $v$  to be solved. The intersection of the edge with the triangle is possible only if

$$0 \leq s, u, v, u + v \leq 1$$

If the above conditions are satisfied, then the intersecting cell is flagged as cutcell and each vertex of the cutcell is checked to classify the vertex as solid node or fringe node using the surface normal test, which will be discussed later.

**B. Classification of Active, Solid Cells, and Blanked Cells**

The active cells are cells that are outside all the components. The active cells are identified using the recursive method. In this method, first one cell in each component grid is identified by choosing a cell near the far-field boundary, confirmed that the cell is outside the bounding box of all other components and flagged as an active cell. Starting from this active cell, each of its neighbors is visited and checked. The unflagged neighbor cells (neither active cell nor cutcell) are flagged as active cells. In turn, their neighbors are checked and flagged as active cells, if they are also unflagged. This

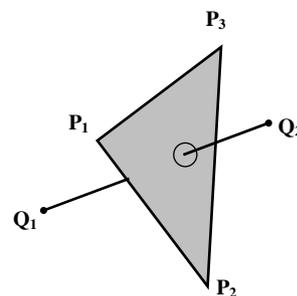


Fig. 7 Intersection of an edge with a triangle.

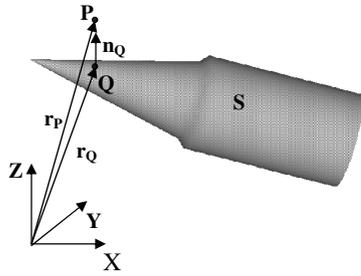


Fig. 8 Surface Normal Test.

procedure is repeated recursively until all the active cells are flagged. Finally, the cells that are neither active nor cutcell are flagged as blanked cells.

C. Classification of Solid Nodes and Fringe Nodes

The vertices of blanked cells are flagged as solid nodes. The vertices of cutcells that lie inside other component are flagged as solid nodes and those that are outside the component are flagged as fringe nodes. The surface normal test is used to check whether a node is inside the component and the discrete surface grid of the component is used to represent the surface for this purpose. Consider a point  $P$  in the domain as shown in Fig. 8. A point  $Q$  on the surface  $S$ , closest to the point  $P$ , is obtained by projecting  $P$  onto  $S$ . Let  $r_p$  be the position vector of point  $P$ ,  $r_Q$  be the position vector of the point  $Q$  on  $S$ , and  $n_Q$  be the unit outward normal vector at point  $Q$ . Then, the point  $P$  is considered to be a solid point of the component, if the following condition is satisfied

$$(r_p - r_Q) \cdot n_Q < 0 \tag{1}$$

Otherwise, the point is flagged as fringe node.

D. Connectivity Generation

The connectivity for an active node is obtained using the grid information. A typical connectivity for an active node  $P$  is shown in Fig. 9. Vertices of all the cells connected to the node  $P$  are considered

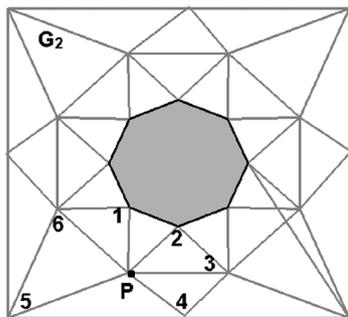


Fig. 9 Connectivity for an active node.

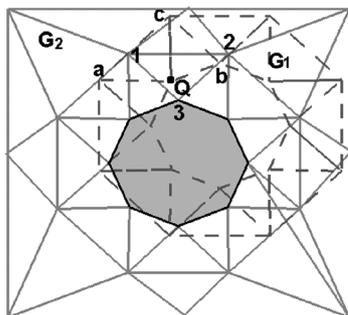


Fig. 10 Connectivity for a fringe node.

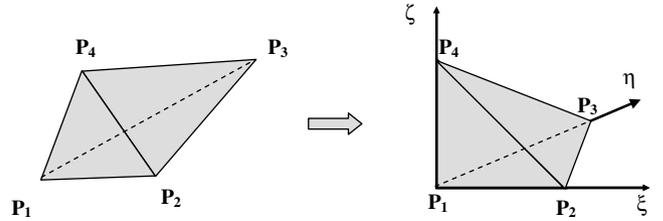


Fig. 11 Mapping of a tetrahedral cell.

as its neighbors. For the fringe nodes, similar to active node, connectivity is first obtained using grid information excluding the vertices that are blanked or the vertices that are on the other side of the surface, as mentioned earlier. A typical connectivity for a fringe node  $Q$  in grid  $G_1$  is shown in Fig. 10. The connectivity includes nodes  $a$  and  $b$  of same grid  $G_1$ . The connectivity for a fringe point should also include nodes in the overlapping grid. This is achieved by finding a cell, called a donor cell, in the overlapping grid  $G_2$  that contains the point  $Q$  as shown in Fig. 10. The vertices of the donor cell (1, 2, and 3) are added to the connectivity of the fringe node  $Q$ . All the vertices are added only when the donor cell is an active cell. If the donor cell is a cutcell, then each vertex is passed through surface normal test before it is added to the connectivity of  $Q$ .

A gradient search method [6] is used to obtain the donor cell. In the gradient search method, physical coordinates  $\mathbf{x}(x, y, z)$  of each cell are mapped to a reference frame of coordinates  $s(\xi, \eta, \zeta)$  of uniform size as shown in Fig. 11. A trilinear function is used for mapping and is given by

$$x(s) = x_1 + (-x_1 + x_2)\xi + (-x_1 + x_3)\eta + (-x_1 + x_4)\zeta \tag{2}$$

Let  $x(s_p)$  be the coordinates of a point  $P$  as a function of the computational space coordinate  $s$  of a candidate donor cell. Values of  $x$  and  $s$  are known for the four vertices of the candidate donor cell. The computational space coordinate  $s_p$  of point  $P$  is obtained by solving Eq. (2) with the known value of  $x$  of the point  $P$ . If the point  $P$  is inside the cell, values of  $s_p$  will be bounded by 0 and 1. If any of the components of  $s_p$  are outside these bounds, then the point is outside the cell and the search must be continued. However, the direction, in computational space, to the cell that bounds  $P$  is indicated by  $s_p$ . Therefore, it is possible to traverse through the neighbors to reach the donor cell shortly as shown in Fig. 12. Once the bounding cell of  $P$  is identified, the vertices of the donor cell are added to the connectivity of point  $P$  as mentioned earlier. A preprocessor has been developed incorporating the various procedures discussed above. Unstructured grids are generated around aircraft and two flight vehicles (at STN 8 and 10) are generated separately. The grids are overlapped to get the distribution of points within the computational domain. The data structure for chimera clouds of points of half-symmetry configuration is obtained using the preprocessor and the data structure for the full aircraft configuration is obtained by reflection. This data structure is used in simulation at different Mach number, angles of attack, and sideslip freestream conditions.

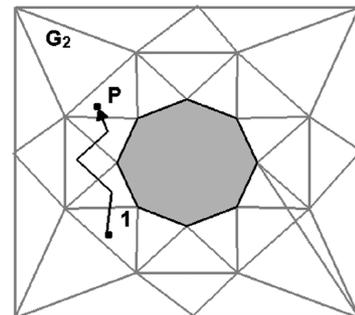


Fig. 12 Gradient search method.

**Table 1** Grid dependence studies and intercode comparison for aerodynamic force and moment coefficients for fighter aircraft

Mach no.	Angle of attack	Code	Grid size (millions)	Normal force coefficient $C_N$	Pitching moment coefficient $C_m$
0.8	10°	q-LSKUM	0.77 (nodes)	0.872	-3.250
			2.05 (nodes)	0.966	-3.595
			3.50 (nodes)	0.968	-3.601
		FLUENT <sup>®</sup>	4.31 (cells)	1.041	-3.848

#### IV. Grid-Free Euler Solver

The flow simulations are carried out on distribution of points using a grid-free Euler solver. The grid-free solver uses LSKUM [7] for spatial discretization and backward Euler time step is used for temporal discretization. The LSKUM is based on the kinetic flux vector splitting (KFVS) [8] scheme, which exploits the connection between the Boltzmann equation of kinetic theory of gases and the governing equations of fluid dynamics using a moment method strategy. More specifically, Euler equations are obtained by taking  $\Psi$ -moments of the Boltzmann equation with Maxwellian as velocity distribution function. In LSKUM, the spatial derivatives of the Boltzmann equation are discretized using a weighted least-squares method and the upwinding is enforced by choosing split substencils from the connectivity based on signs of the molecular velocity to evaluate the spatial derivatives. Finally, taking  $\Psi$ -moments leads to the LSKUM numerical scheme. The higher-order accuracy in space is achieved using a defect correction method [9] in which the lower-order spatial errors are removed using an iterative strategy. An improved version of LSKUM is  $q$ -LSKUM, in which the entropy variables, also called  $q$ -variables [10], are used in the defect correction step to achieve higher-order accuracy in space at all points including boundary points [11]. Kinetic characteristic boundary condition (KCBC) and kinetic outer boundary condition (KOBC) [12] are implemented for treating wall and far-field boundaries, respectively. The  $q$ -LSKUM Euler solver is parallelized in the message passing interface (MPI) environment to run on distributed computing systems [13]. The flow solver has been verified against standard test cases and validated [14].

#### V. Flow Simulation

The overlapped cloud of points is generated around aircraft and flight vehicles. Then, the grid-free  $q$ -LSKUM Euler solver is applied on the cloud of points to estimate aerodynamic loads on aircraft and flight vehicles. The diameter and base area of the flight vehicle are considered as reference length and reference area for the flight

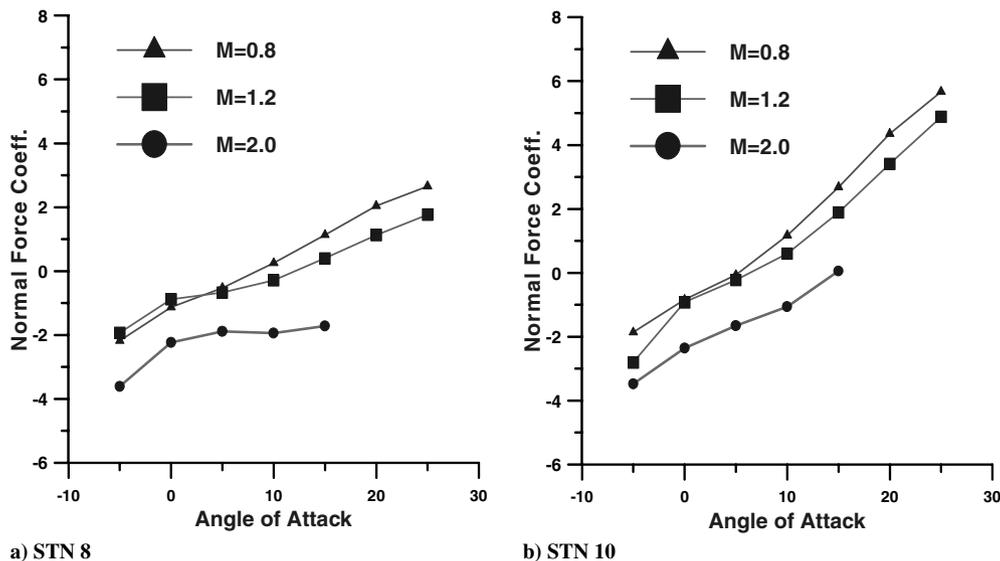
vehicle loads. Similarly, the mean aerodynamic chord and surface area of the wing are considered as the reference length and the reference area for the aircraft force and moments. The flow is allowed to pass through air-intake by applying KOBC with first-order extrapolation of neighboring point to construct incoming Maxwellian. This avoids modeling of inlet as wall and formation of strong shock ahead of intake. This boundary condition works well in the subsonic range of Mach numbers also. The simulations have been carried out on a Linux cluster using 100 nodes of Intel dual Xeon at 3.8 GHz with 4 GB RAM processor. Each case took about 5–8 h of computation time for iterative convergence.

##### A. Grid Dependence Study and Intercode Comparison

The grid dependence study has been carried out for the half-symmetry aircraft configuration at freestream Mach number 0.8 and angle of attack 10° using three different grids with 0.77, 2.05, and 3.5 million nodes. The aerodynamic normal force and pitching moment coefficients obtained using these grids are presented in Table 1, and it can be seen that they vary marginally from medium grid (2.05 million nodes) to fine grid (3.5 million nodes). Thus the medium grid has been used for further computations, as this grid has optimal grid size for the transonic flow conditions. An inviscid simulation has been carried out with 4.5 million cells using the finite volume code FLUENT [15], and the aerodynamic coefficients obtained from FLUENT and  $q$ -LSKUM compare within 7% as presented in Table 1.

##### B. Aerodynamic Characterization of Fighter Aircraft with Flight Vehicles

The full configuration (full fighter aircraft with four flight vehicles) has been simulated with a grid of 7.46 million nodes. Here, the 2.05 million nodes around half-aircraft and 0.84 million nodes grid around each flight vehicle have been used. The flow simulations are carried out for the freestream conditions with Mach number ( $M_\infty$ ) = 0.8, 1.2 and 2.0, angles of attack ( $\alpha$ ) = -5 to 25° and sideslip angles ( $\beta$ ) = 0 and 5°. The variation of normal force

**Fig. 13** Normal force coefficient of flight vehicle with angle of attack.

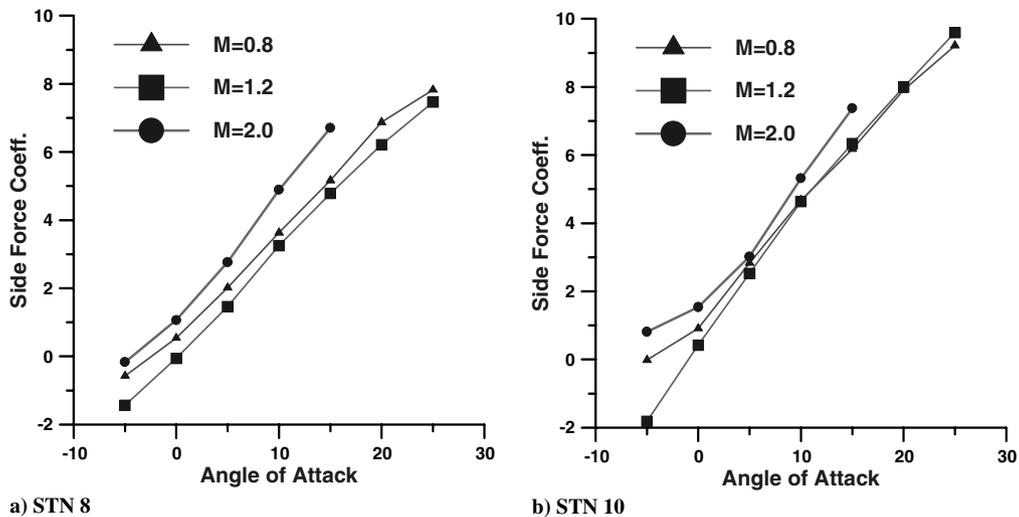


Fig. 14 Side force coefficient of flight vehicle with angle of attack.

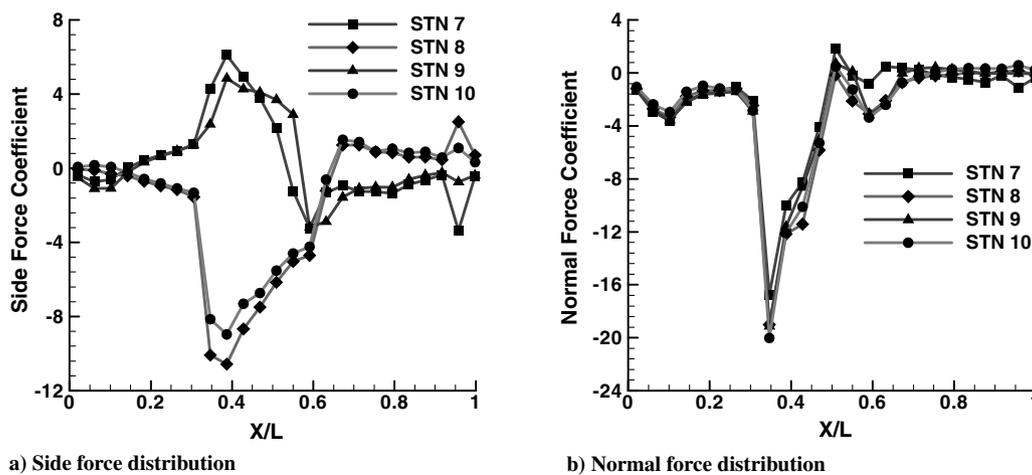


Fig. 15 Captive loads distribution along the nondimensional axial length of the flight vehicle at  $M_\infty = 1.2$ ,  $\alpha = -12.6^\circ$ ,  $\beta = 5.2^\circ$ .

coefficients with angles of attack for vehicles at station (STN) 8 and 10 at various Mach numbers are plotted in Figs. 13a and 13b, respectively. It is observed from Fig. 13a that the normal force coefficient of flight vehicle at STN 8 is negative up to angle of attack of  $10^\circ$  at Mach number 0.8 and 1.2. At Mach 2, the downward force is observed at even higher positive angles of attack. This shows that stronger compression takes place between the launcher and flight vehicle. It implies that during the pitch-up maneuver of aircraft both inertia loads and aerodynamic loads are in the same direction. Even though a similar trend is observed for flight vehicle at STN 10 as shown in Fig. 13b, due to wing tip relieving effect, the slopes are better compared with other STN. The side force coefficients for vehicles at STN 8 and 10 have been plotted with angles of attack in Figs. 14a and 14b respectively. It is clear from the figures that the forces act away from the aircraft for positive angles of attack and toward the aircraft at negative angles of attack at all Mach numbers. The side forces are mainly due to the sweepback angle of the aircraft wing and wing tip effect. As suggested in the MIL-STD 8591, the flow conditions corresponding to critical maneuvers such as pullout, pushover, roll pullout, and pushover are simulated. The aerodynamic loads on the flight vehicles are obtained for those flight conditions. The nondimensional aerodynamic side force and normal force distribution along the length of the flight vehicles from their nose during the pushover maneuver of the aircraft at transonic Mach number ( $M_\infty = 1.2$ ,  $\alpha = -12.6^\circ$ ,  $\beta = 5.2^\circ$ ) are given in Figs. 15a and 15b respectively. The maximum loading occurs on the wing, the

side force is acting toward the aircraft, and the normal force is acting downward due to negative angle of attack. The tail surfaces experience considerable side force that can be seen in Fig. 15a. The captive aerodynamic load corresponding to a critical case has been used for the structural analysis of the launcher.

## VI. Conclusions

The captive loads at various Mach numbers, angles of attack, and sideslip angles have been estimated for a flight vehicle mounted on the fighter aircraft, using a grid-free Euler solver. Forces and moments acting on the fighter aircraft with launchers and flight vehicles have been estimated at critical flow conditions, and load distributions have been obtained for those flow conditions. The load distribution on the flight vehicle along with the inertia loads will be used for verifying the load-carrying capability of launch shoes and rail.

## Acknowledgments

The authors thank the Director of Defence Research and Development Laboratory and the Project Director for their support and encouragement during the work. They also express their sincere gratitude to the Deputy Project Director and his team for technical discussions, and also for providing the CAD model of the flight vehicle and fighter aircraft.

## References

- [1] MIL-A-8591H, "Airborne Stores, Suspension Equipment and Aircraft Interface (Carriage phase); General Design Criteria for Military Specifications," March 1990.
- [2] Jeter, E. L., "Application of Modern Multidisciplinary Approaches to the Integration of Weapons on Aircraft," *RTO SCI Symposium on Aircraft Weapon System Compatibility and Interaction*, NATO/RTO, pp. 20-1–20-13.
- [3] Anandhanarayanan, K., Nagarathinam, M., and Deshpande, S. M., "Development and Applications of a Grid-Free Kinetic Upwind Solver to Multi-Body Configurations," AIAA Paper 2005-4628, June 2005.
- [4] More, R. R., Srinivasa Raju, S., Anandhanarayanan, K., and Krishnamurthy, R., "Nose Panel Opening Study of a Hypersonic Launch Vehicle," *Proceedings of the First International Conference on High Speed Transatmospheric Air and Space Transportation*, Aeronautical Society of India, Hyderabad, India, June 2007, pp. 171–177.
- [5] Steger, J. L., and Benek, J. A., "On the Use of Composite Grid Schemes in Computational Aerodynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, 1987, pp. 301–320. doi:10.1016/0045-7825(87)90045-4
- [6] Meakin, R. L., "Composite Overset Structured Grids," *Handbook of Grid Generation*, edited by J. F. Thompson, N. P. Weatherill, and B. Soni, CRC Press, Boca Raton, FL, 1999, pp. 11.1–11.20.
- [7] Ghosh, A. K., and Deshpande, S. M., "Least Squares Kinetic Method for Inviscid compressible flows," *AIAA 12th CFD Conference*, AIAA Paper 95-1735, June 1995.
- [8] Mandal, J. C., and Deshpande, S. M., "Kinetic Flux Vector Splitting for Euler Equations," *Computers and Fluids*, Vol. 23, No. 2, 1994, pp. 447–478. doi:10.1016/0045-7930(94)90050-7
- [9] Deshpande, S. M., "Meshless Method, Accuracy Symmetry Breaking, Upwinding and Lskum," Indian Inst. of Science, Department of Aerospace Engineering Rept. No. 2003 FM 1, 2003.
- [10] Deshpande, S. M., "On the Maxwellian Distribution, Symmetric Form and Entropy Conservation for the Euler Equations," NASA Technical Paper 2583, 1986.
- [11] Deshpande, S. M., Anandhanarayanan, K., Praveen, C., and Ramesh, V., "Theory and Applications of 3-D LSKUM Based on Entropy Variables," *International Journal for Numerical Methods in Fluids*, Vol. 40, Nos. 1–2, 2002, pp. 47–62. doi:10.1002/flid.266
- [12] Ramesh, V., "Least Squares Grid-Free Kinetic Upwind Method," Ph.D. Thesis, Indian Inst. of Science, Bangalore, India, 2001.
- [13] Anandhanarayanan, K., Nagarathinam, M., and Deshpande, S. M., "Parallelization of a Grid-free Kinetic Upwind Solver," AIAA Paper No. 2005-4846, 2005.
- [14] Anandhanarayanan, K., "Development and Applications of a gridfree Kinetic Upwind Solver to Multibody Configurations," Ph.D. Thesis, Indian Inst. of Science, Bangalore, India, 2003.
- [15] FLUENT® User Reference Manual.