

Development of Three-dimensional Grid-free Solver and its Applications to Multi-body Aerospace Vehicles

K. Anandhanarayanan

Defence Research and Development Laboratory, Hyderabad-500 058

Email: kanand_cfd@yahoo.com

ABSTRACT

Grid-free solver has the ability to solve complex multi-body industrial problems with minimal effort. Grid-free Euler solver has been applied to number of multi-body aerospace vehicles using Chimera clouds of points including flight vehicle with fin deflection, nose fairing separation of hypersonic launch vehicle. A preprocessor has been developed to generate connectivity for multi-bodies using overlapped grids. Surface transpiration boundary condition has been implemented to model aerodynamic damping and to impose the relative velocity of moving components. Dynamic derivatives are estimated with reasonable accuracy and less effort using the grid-free Euler solver with the transpiration boundary condition. Further, the grid-free Euler solver has been integrated with six-degrees of freedom (6-DOF) equations of motion to form store separation dynamics suite which has been applied to obtain the trajectory of a rail launch air-to-air-missile from a complex fighter aircraft.

Keywords: Grid-free method, chimera clouds, high speed flows, moving multi-body, complex multi-body, computational fluid dynamics

1. INTRODUCTION

Grid generation is one of the major challenges in solving governing equations of fluid dynamics around complex flight vehicle configurations. The complexity increases further in generating suitable grids around relatively moving multi-bodies. The grid-free methods which operate on distribution of points, reduce the difficulty of grid generation to a greater extent. Further, the grid-free methods are amenable for parallelisation due to uniform and simple data structure for even complex multi-bodies and hence it is easy to handle relatively moving multi-bodies in a parallel environment. Some of the notable grid-free methods that are being used in computational fluid dynamics (CFD) for compressible fluids are Deshpande's least squares kinetic upwind method (LSKUM)¹⁻³ and its extension to higher-order accuracy through entropy variables(q) called q -LSKUM^{4,5}, meshless method of Batina⁶, gridless method of Morinishi⁷, finite point method (FPM)⁸, least squares finite difference-upwind (LSFD-U) method of Sridhar and Balakrishnan⁹ and kinetic meshless method (KMM) of Praveen¹⁰.

The above methods basically use least squares method for estimating spatial derivatives of fluxes. Among those methods, LSKUM and q -LSKUM have been applied on various distributions of points¹¹⁻¹⁴ and have solved number of complex fluid problems¹⁵⁻¹⁸. A 3-D grid-free Euler code based on q -LSKUM has been developed¹⁹; lower-upper symmetric gauss Seidel²⁰ has been implemented in q -LSKUM code²¹ to get faster convergence and the code has been

parallelised using message passing interface (MPI) to run on cluster/parallel computers²². The code has been thoroughly verified and validated for various complex flight vehicle configurations, from subsonic to hypersonic flows¹⁷. The code has been integrated with a 6-DOF solver for equations of motion and preprocessor to form a store separation dynamics suite²³. The suite is being used in quasi-steady mode to solve the store separation problems. The suite has been validated for a generic store separating from a wing-pylon configuration. The code is persistently being enhanced to meet the requirements of the current need and future requirements. The paper presents recent modifications to the q -LSKUM code, store separation dynamics suite and their applications to complex multi-body flight vehicle configurations.

2. LEAST SQUARES KINETIC UPWIND METHOD

A detailed description of the method is presented²⁴ and only a brief description is given here for completeness. Entropy (q) variables-based least squares kinetic upwind method (q -LSKUM) is based on the kinetic flux vector splitting (KFVS)²⁵ 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 Ψ -moments of the Boltzmann equation with Maxwellian as velocity distribution function. In q -LSKUM, the spatial derivatives of the Boltzmann equation are discretised using weighted least squares method and

the upwinding is enforced by choosing split sub-stencils from the connectivity based on sign of the molecular velocity to evaluate the spatial derivatives. Finally, taking Ψ -moments lead to q -LSKUM numerical scheme. Entropy variables, also called q -variables, are used in the defect-correction step to achieve higher-order accuracy in space at all points including boundary points. The q -LSKUM operates on a distribution of points in the computational domain and does not require complex grid generation effort to solve the governing equations of fluid dynamics. Therefore, it considerably reduces the grid generation time and also makes it possible to obtain solutions for the geometrically complex configurations. The performance of the solver crucially depends upon the quality of the connectivity (set of neighbours) to estimate the spatial derivatives of flux vectors using least squares method.

3. CONNECTIVITY

The grid-free solver requires just a distribution of points and a set of neighbours, called connectivity, around each point. The distribution of points can be obtained by two methods, namely simple cloud method and chimera cloud method. In the 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 as the present method is different from the chimera grid method²⁶ as it is more efficient than chimera cloud method which is being discussed along with the procedures of generating connectivity.

An efficient preprocessor¹⁹ 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. Due to 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 called 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. 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. 1. The first type of cells are those cells that lie completely inside the solid body. The second type of cells, are those that are 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 thin surface, but all the nodes

are outside the solid body, i.e., the cells pierce through the body. The fourth type of cells are fully outside of the cutting-surface. Handling of first and fourth types of cells in our procedure is the same as that of the chimera grid method.

The nodes of cells of first type are removed from the computation and these are called as blanked nodes. The nodes of cells of fourth type are available for computation; these are called as field nodes and the cells are called as field cells. In the chimera grid method, the second and third types of cells are also blanked and the field cells adjacent to the blanked cells are flagged as fringe cells. The flow field of these fringe cells is not computed, but interpolated from the field cells of the grid whose surface is used for hole-cutting. This condition implies that some field cells should be available between the two hole-cutting surfaces in both the grids for interpolation of flow field 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 deflected fins.

In our approach, the nodes of cells of second type, that are outside, are also used for the computation and those nodes are called as fringe nodes whose connectivity should include neighbours from the other grids. The nodes inside the solid body are removed from the neighbour list of the fringe nodes. The nodes of cells of third type are also called as fringe nodes, but the nodes on the same side of the surface are only included in their respective connectivity lists. 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: (i) the same upwind discretised equations, which satisfy local characteristics of the flow, are solved at these points, (ii) there is no time lag in the unsteady computations between various clouds, (iii) uniform treatment for all the points, which simplifies parallelisation and implementation of convergence acceleration methods, and (iv) importantly, very thin gap can be modelled 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; neighbours for field nodes are obtained from the cell connectivity information, and for the fringe nodes, neighbours from the same side of the hole-cutting surface of the same grid and some nodes from the overlapping grids are also included. The neighbour nodes in the overlapping grids are the vertices of the cell in the overlapping grid that contains the fringe node. Overall, connectivity generation process involves identification of cut-cell, classification of cut-cells and solid cells, classification of solid-nodes, field nodes, and fringe nodes, and connectivity generation for the above nodes.

3.1 Identification of Cut-cell

Each component is associated with a bounding box as shown in Fig. 2 (for clarity 2-D example is shown) and

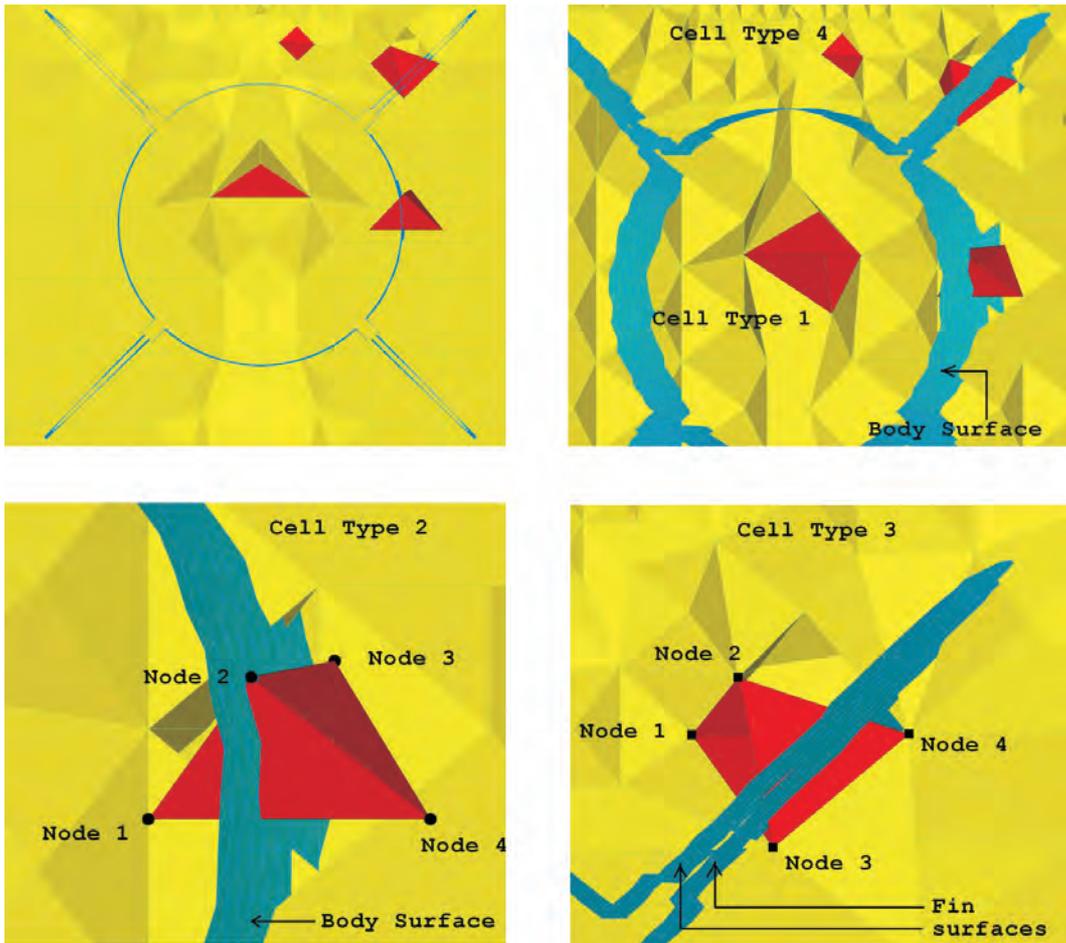


Figure 1. Four possible types of cells in the overlapped grids.

the bounding box is divided into small uniform bins as shown in Fig. 3. 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 number of bins along each co-ordinate direction (I, J) . For example,

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

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

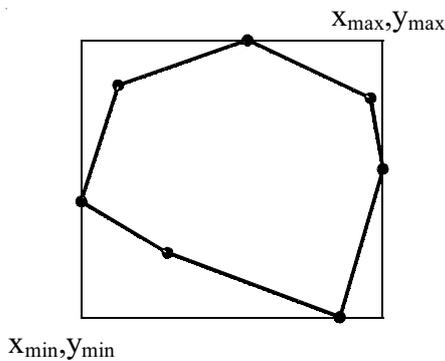


Figure 2. Bounding box of a component.

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 shown in Fig. 3.

In the cut-cell 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

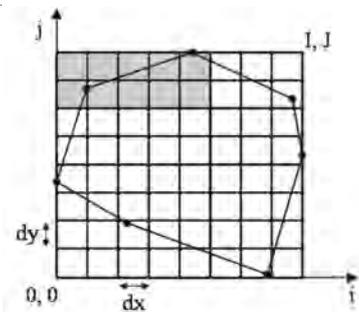


Figure 3. Intersecting bins of a component edge.

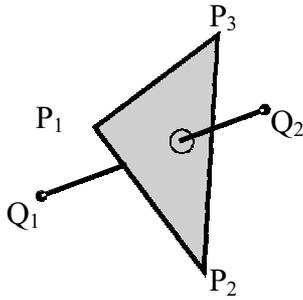


Figure 4. Intersection of an edge with a triangle.

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. 4. The parametric representation of the edge and triangle are

$$Q = Q_1 + (1-s)Q_2$$

$$P = P_1 + u(P_2 - P_1) + v(P_3 - P_1)$$

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 cut-cell and each vertex of the cut-cell is checked to classify the vertex as solid node or fringe node using surface normal test which will be discussed later.

3.2 Classification of Active Cells, Solid Cells, and Blanked Cells

The active cells are cells that are outside all the components. The active cells are identified using 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 the cell as active cell. Starting from this active cell, each of its neighbours is checked. The un-flagged neighbour cells (neither active cell nor cut-cell) are flagged as active cells. In turn, their neighbours are checked and flagged as active cells, if these are also un-flagged. This procedure is repeated recursively till all the active cells are flagged. Finally, the cells that are neither active nor cut-cell are flagged as blanked cells.

3.3 Classification of Solid Nodes and Fringe Nodes

The vertices of blanked cells are straightaway flagged as solid nodes. The vertices of cut-cells, which lie inside other component, are flagged as solid nodes and 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. 5. A point Q

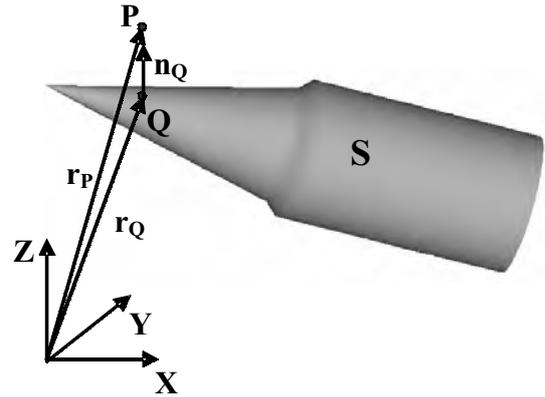


Figure 5. Surface Normal Test.

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.

3.4 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. 6. Vertices of all the cells connected

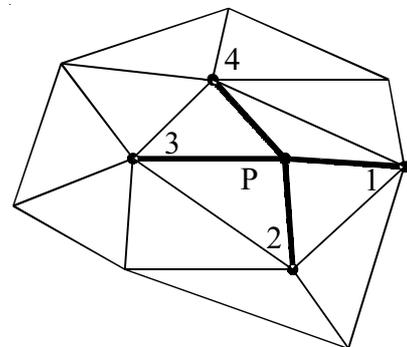


Figure 6. Connectivity for an active node.

to the node P are considered as its neighbours.

For fringe nodes, similar to active node, first connectivity is 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. 7. The connectivity includes nodes a and b of the 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 donor cell, in the overlapping grid G_2 that contains the point Q as shown in Fig. 7. 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 cut-cell, then each vertex is passed through surface normal

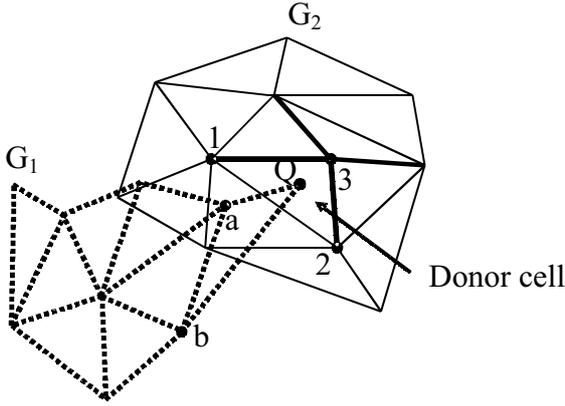


Figure 7. Connectivity for a fringe node.

test before it is added to the connectivity of Q .

A gradient search method²⁷ is used to obtain the donor cell. In the gradient search method, physical coordinates $x(x, y, z)$ of each cell are mapped to a reference frame of coordinates $s(\xi, \eta, \zeta)$ of uniform size as shown in Fig. 8. A trilinear function is used for mapping and is given by

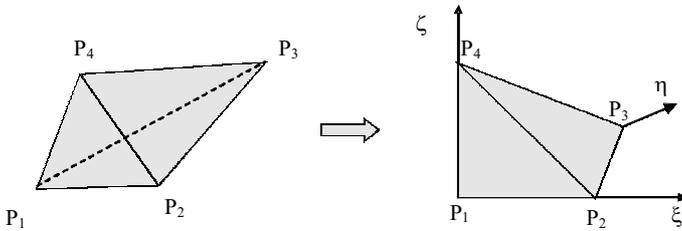


Figure 8. Mapping of a tetrahedral cell.

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

Let $x(s_p)$ 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 Eqn (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 neighbours to reach the donor cell shortly as shown in Fig. 9. 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.

4. MODELLING OF AERODYNAMIC DAMPING

Dynamic derivatives or aerodynamic damping derivatives are very important for slender-finned vehicles and play a vital role in neutrally stable air-to-air-missiles separating from fighter aircraft. During store separation studies, these derivatives are naturally included in the unsteady simulations, but in the quasi-steady simulations, the dynamic derivatives are generally ignored or added explicitly in the aerodynamic moments. These damping terms are obtained either empirically

or experimentally for isolated flight vehicle not in the presence of aircraft flow field. In the grid-free code, the aerodynamic damping is estimated using surface transpiration boundary condition²⁹. The boundary condition includes damping due to linear and angular rates. During simulation of store separation, the relative velocity of the store is imposed as solid wall boundary condition on the surface of the store. This boundary condition models both relative velocity of the store as well as the aerodynamic damping to the store. The transpiration boundary condition models the relative movement by imposing a flow into or out of each boundary node on the surface of the store to match the linear and angular velocity of that point on the store relative to the parent aircraft. The transpiration velocity, \vec{v}^{Trans} is prescribed at each node on the surface of the store, such that,

$$\vec{v}^{trans} = \vec{v}^{Store} + \vec{\omega}^{Store} \times (\vec{r} - \vec{r}_{CG}) \quad (3)$$

where \vec{r}_{CG} is the position vector of the CG of the store and \vec{r} is the position vector of the node on the store at which transpiration velocity is applied. The translational velocity \vec{v}^{Store} and rotational rates $\vec{\omega}^{Store}$ predicted by the 6-DOFs trajectory solver are used to find the transpiration velocity at each store's surface nodes and the transpiration velocity is used in the grid-free solver to apply the surface boundary condition. Since the motion of the store relative to its parent aircraft is modelled by this technique, the effects of the induced incidence and damping due to rotary motion are also modelled.

5. APPLICATIONS

The grid-free Euler solver has been thoroughly validated for various complex flight vehicle configurations and is being routinely used for the flight vehicle design and analysis. Some of the important recent applications of the grid-free solver are presented.

5.1 Control Characteristics of Finned Missiles

The prediction of aerodynamic loads and hinge moments of a missile with all movable fins are crucial in the design of control systems. Further, minimising the hinge moment is necessary to meet the control actuator power. Estimation of aerodynamic loads of missiles with deflected fins involves

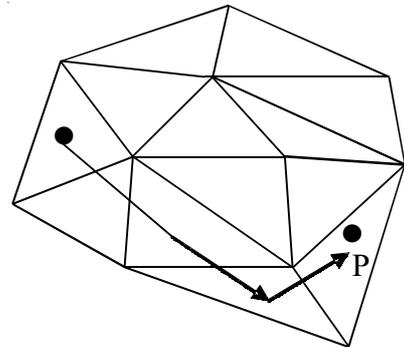


Figure 9. Gradient search method.

exact modelling of the gap between the missile and the fins during the grid generation and flow simulation. In our approach, clouds of points are generated around the body and fins separately. The cloud of points around the fins are rotated for the desired deflection angles and overlapped with the cloud of points around the body. The solid points are blanked and connectivity is generated for each point in the overlapped clouds. This approach automatically models the thin gap irrespective of complexity of the shape. The various fin deflection angles can be simulated efficiently by rotating the cloud around the fins and overlapping with cloud of points around the body. This approach does not require regeneration of grids for every deflection angle. The grid-free solver has been applied to a flight vehicle configuration with various deflection angles²⁸. The point distribution is obtained using overlapped structured grids as shown in Figs 10 and 11. The number of points around body, wings, and fins are 1.7, 1.0, and 0.8 million, respectively. The q -LSKUM solver is applied on the cloud of points to obtain the flow fields around the vehicle. The surface Mach contours for a flow condition are shown in Fig. 12. The flow deflection near leading edge of deflected fins is clearly seen. The predicted aerodynamic loads are compared with the experimental results in Table 1 which compare well.

5.2 Fairing Separation of Hypersonic Launch Vehicle

A hypersonic cruise vehicle is placed in the nose portion of a launch vehicle and covered with nose fairings (Fig. 13). The nose fairings are separated before launching of the cruise vehicle. The fairing is opened in two portions with the help of aerodynamic loads and is separated from the vehicle after opening to an optimum angle. The design of fairing, separation mechanism, and the time sequences of separations require aerodynamic characteristics of the vehicle with fairings at different positions. In the present work, 3-D unstructured grids are generated around the launch vehicle with cruise vehicle and two nose fairings separately. The surface grids near the fairings are shown in Fig. 14. The grids are overlapped to get the distribution of points within the domain. Total number of points in the domain is 1.3 million. The grid blocks around the nose fairings are rotated about their respective hinge lines during the opening of fairings and as well moved during the separation of fairings to get the distribution of points at different geometrical orientations of the panels. The preprocessor is then applied on the overlapped grids to generate data structure. The overlapped unstructured grids before and after the blanking of solid nodes are shown in Figs 15 and 16 respectively. This application is complex and would have been a great challenge for even the chimera grid-based approach.

At smaller opening angles, the vehicle grid and the panel grids cut each other and no field cell is available in both the grids near the hinge line because of small gaps. Some of the fringe points of one grid has only wall points of the other grid as neighbours. The q -LSKUM

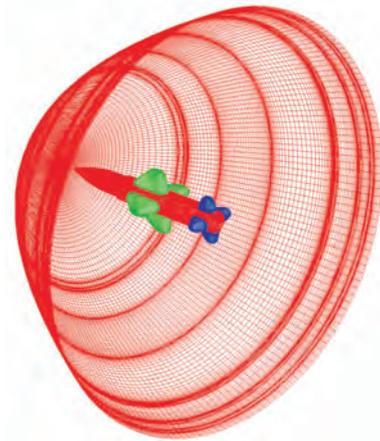


Figure 10. Overlapped grids around the flight vehicle.

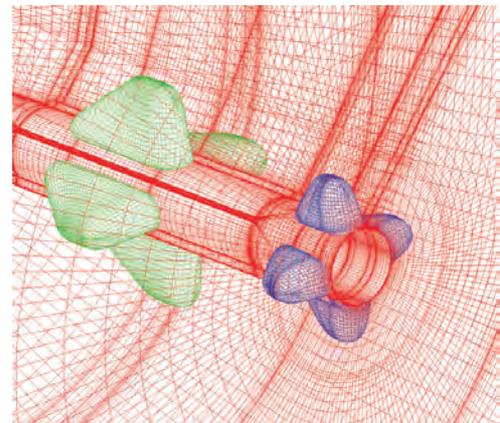


Figure 11. Zoomed view of overlapped grids showing grid blocks around wings and fins.

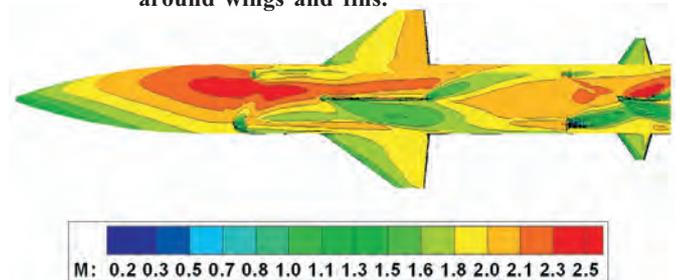


Figure 12. Surface Mach contours on aerospace vehicle ($M_\infty = 2.0$, $\alpha = 14^\circ$, $\delta = 5^\circ$).

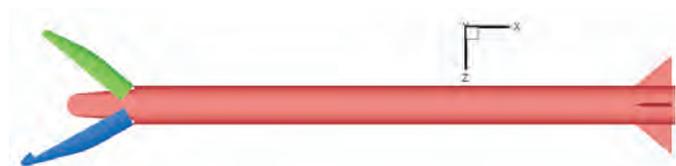


Figure 13. Launch vehicle with cruise vehicle during opening of fairings in the yaw plane.

solver is applied on the cloud of points for flow simulations to get aerodynamic loads on the vehicle and the fairings. The Mach contours for a typical orientation during opening and separation are shown in Figs 17 and 18, respectively.

The flow field consists of shocks and expansions

Table 1. Comparison of aerodynamic force and moment coefficients with the experimental data.

Aerodynamic coefficients	M_∞	0.6		3	
	α	2.5°	5.0°	2.5°	5.0°
Normal force, C_N	Present	0.494	0.984	0.335	0.705
	Expt.	0.490	1.046	0.375	0.793
Pitching moment, C_m	Present	-2.308	-4.723	-1.475	-3.175
	Expt	-2.170	-4.712	-1.653	-3.535

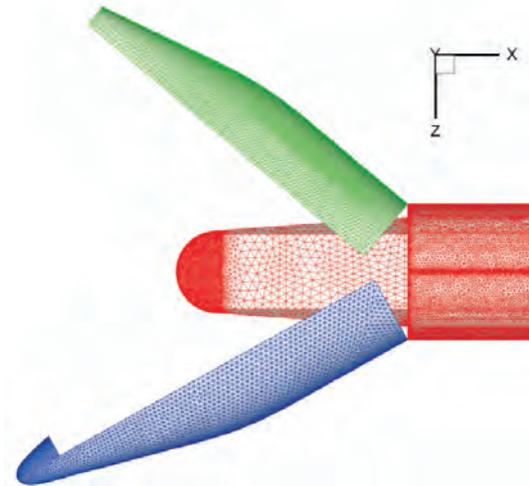


Figure 14. Zoomed view of surface grids.

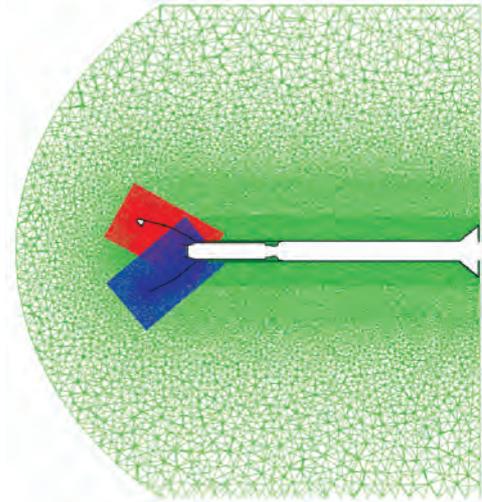


Figure 16. Overlapped grids after blanking of solid nodes.

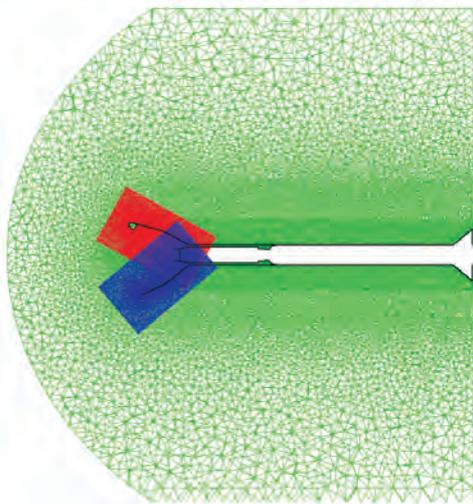


Figure 15. Overlapped grids before blanking of solid nodes.

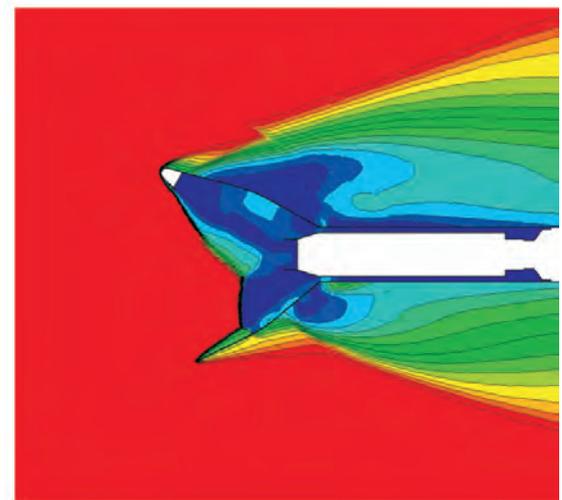


Figure 17. Mach contours in yaw plane of the launch vehicle during opening of fairings.

waves which cross different domain boundaries. Since, in the present approach, the governing equations are solved at all the points (no interpolation as present in the chimera grid approach), the discontinuities are captured crisply. The figures indicate that there is a shock-shock interaction, a triple-point shock occurs near lower side panel and there are expansions and recompression shocks at aft of the panels. The flow field is complex which includes large subsonic pockets, and therefore, number of iterations required

for convergence for each case is order of few ten thousands. The flow field of the previous case is used as initial condition for the computation for the next orientation of fairings and the better initial conditions reduce the computation time by one order. The aerodynamic data estimated at various panel orientations are used to arrive at the scheme of panel separation and time sequence of separations of panels. The trajectory of fairings during opening and separation are shown in Fig. 19.

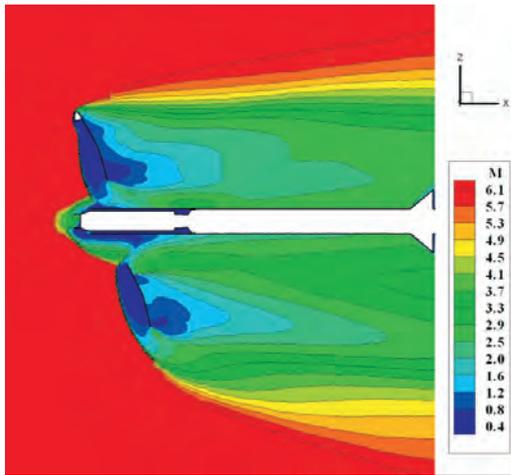


Figure 18. Mach contours in yaw plane of the launch vehicle during separation of fairings.

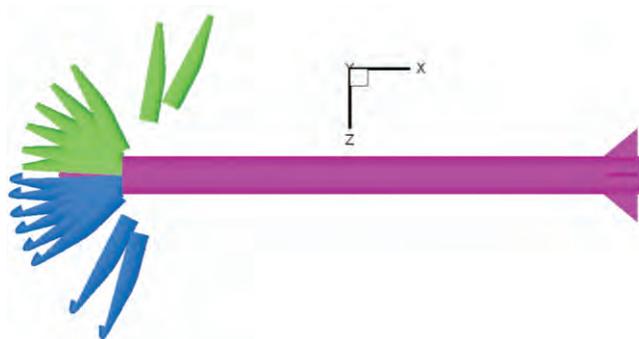


Figure 19. Trajectory of fairings during opening and separation.

5.3 Aerodynamic Roll Damping of a Finner Configuration

The grid-free solver is applied to the finner configuration at free stream Mach number (M_∞) = 2.5 and angle of attack (α) = 10° with zero and 0.01 non-dimensional roll rate ($\bar{p} = pD_{ref}/2U_\infty$) to compute roll damping coefficient. The roll rate (p) is non-dimensionalised with body diameter D_{ref} and free-stream flow velocity U_∞ . The surface Mach contours with and without roll rate are shown in Figs 20 and 21, respectively. The increase in local Mach number due to roll rate can be seen on the surface of the fins.

The aerodynamic coefficients with and without roll rate are given in Table 2. The aerodynamic normal force and pitching moment coefficients do not vary with roll rate and compare very well with experimental results. The solver has predicted the coefficient of roll damping as -18.0 compared to the experimental value³⁰ of -20.5 and the results prove the validity of the present approach for estimating forces and moments due to aerodynamic damping.

Table 2. Aerodynamic force and moment coefficients for finner configuration

α (Deg.)	Roll Rate $\bar{p} = \frac{pD}{2U_\infty}$	Normal Force C_N		Pitching moment C_m		Rolling moment C_l	
		Present	Expt.	Present	Expt.	Present	Expt.
10	0	1.677	1.765	-1.642	-1.588	0.001	---
10	0.01	1.681	---	-1.653	---	-0.178	-0.205

5.4 Missile Powered Separation from Fighter Aircraft

Separation dynamics of a rail-launched air-to-air missile from an aircraft (Fig. 22) has been carried out using the grid-free Euler solver. In the present study, the simulations are carried out including the missile motion in rail, tip-off motion at the end of rail motion, and free flight in the presence of aircraft flow field.

The constraint forces of launch shoes are derived and implemented in the six degrees of freedom (6-DOFs) trajectory solver. The present study includes the launch shoes of missile and rail launcher of the aircraft in the geometry definition. There is a small gap of 1 mm present between the launch shoes and rail launcher. As stated earlier, unstructured grids are generated around missiles and aircraft separately and overlapped to get the distribution of points. The surface grids on aircraft and missiles are shown in Fig. 23.

The number of points around aircraft is 2 million and about 1 million points are around each missile configuration. As mentioned earlier, the gaps are modelled automatically and no special care is required to treat the gaps. This is

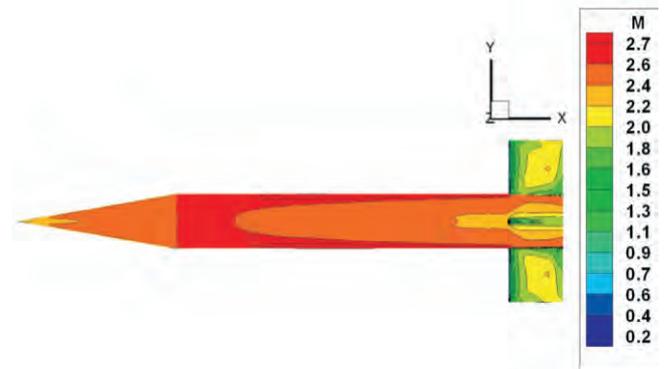


Figure 20. Surface Mach contours in yaw plane without roll rate.

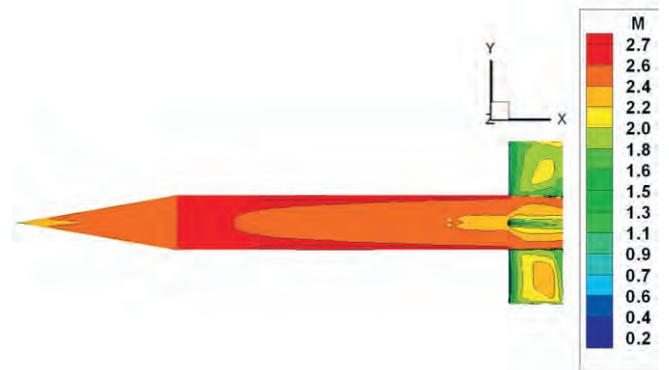


Figure 21. Surface Mach contours in yaw plane with roll rate.



Figure 22. Aircraft with four missiles attached with the help of launcher and launch shoes.

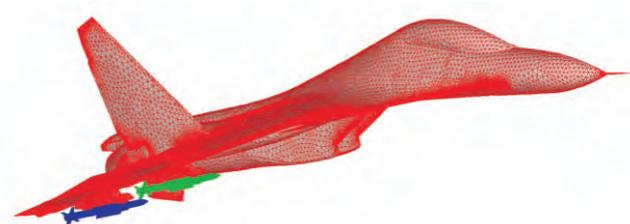


Figure 23. Surface grids of aircraft and missiles.

another application where grid-free method stands out in handling multi-body geometry. The grid-free Euler solver is applied on overlapped clouds of points to obtain aerodynamic coefficients which are used to integrate equations of motion using 6-DOF trajectory solver to get new position and velocity of missile. The cloud of points around the missile is moved to new location and the connectivity is regenerated. The missile velocity obtained using 6-DOFs solver is used to treat the moving wall points with surface transpiration boundary condition which models the relative velocity of missile with respect to aircraft as well as aerodynamic damping. The procedure is repeated till the missile reaches safe distance. The overlapped grids and corresponding Mach contours at different instant of time are shown in Figs 24 and 25, respectively.

6. CONCLUSIONS

The grid-free Euler solver has been applied to a number of multi-body aerospace vehicles using chimera clouds of points. A preprocessor has been developed to generate connectivity using chimera clouds of points. Surface

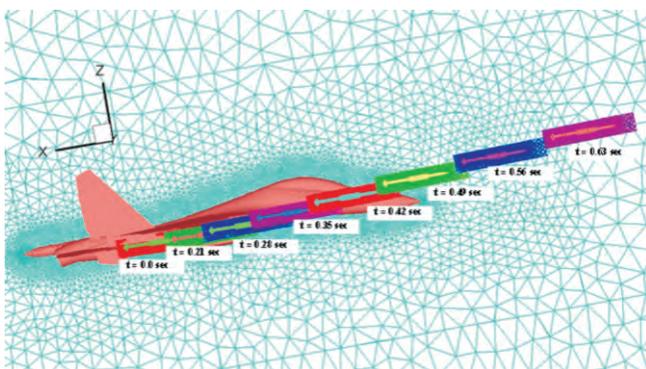


Figure 24. Cloud of points at different instant of time.

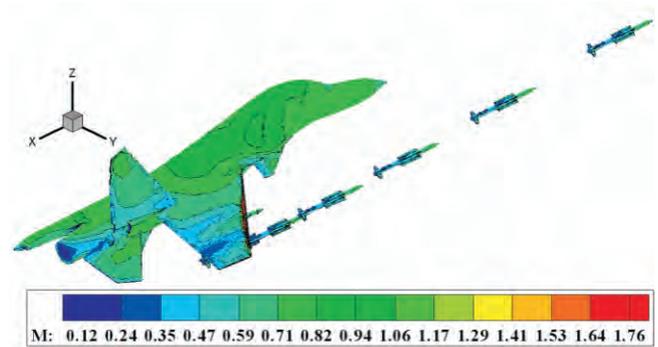


Figure 25. Surface Mach contours at different instant of time.

transpiration boundary condition is implemented to model aerodynamic damping and to impose relative velocity of moving components. Dynamic derivatives are estimated with reasonable accuracy and less effort using the grid-free Euler solver with transpiration boundary condition. Further, the grid-free Euler solver along with transpiration boundary condition has been integrated with 6-DOFs equations of motion to solve powered separation of air-to-air-missile from a complex fighter aircraft.

ACKNOWLEDGEMENTS

Author expresses his sincere gratitude to Prof SM Deshpande, JNCASR, Bangaluru for his guidance during the development of the solver, Shri P Venugopalan, Director, DRDL; Dr Debasis Chakraborty, Technology Director, DOCD; and Dr R Krishnamurthy, Head, CFD Division; for their support and encouragement during the application of solver to complex flight vehicle configurations. The author thanks his co-workers Rohit R More, Harish, Pavanakumar, Mondal, Vaibhav Shah and Konark Arora whose support made it possible to reach this level of maturity in the application of grid-free solver.

REFERENCES

1. Deshpande, S.M.; Ghosh, A.K. & Mandal, J.C. Least squares weak upwind methods for Euler equations. *In* Department of Aerospace Engineering, Indian Institute of Science, Bangaluru, Fluid Mechanics Report No. FM4, 1989.
2. Ghosh, A.K. & Deshpande, S.M. Least squares kinetic method for inviscid compressible flows. *In* AIAA 12th CFD Conference, San Diego, California, June 1995. Paper No. AIAA-95-1735.
3. Deshpande, S.M. Meshless method, accuracy symmetry breaking, upwinding and LSKUM. Department of Aerospace Engineering, Indian Institute of Science, Bangaluru, Fluid Mechanics Report No. FM 1, 2003.
4. Dauhoo, M.Z.; Ghosh, A.K.; Ramesh, V. & Deshpande, S.M. *q*-LSKUM—A new higher-order kinetic upwind method for Euler equations using entropy variables. *Comput. Fluid Dyn. J.*, 2000, **9**(1), 272-77.
5. Deshpande, S.M.; Anandhanarayanan, K.; Praveen, C. & Ramesh, V. Theory and applications of 3-D LSKUM based on entropy variables. *Int. J. Num. Meth. Fluids*,

- 2002, **40**, 47-62.
6. Batina, J.T. A grid-free Euler/Navier-Stokes solution algorithm for complex aircraft applications. AIAA Paper-1993-0333, 1993.
 7. Morinishi, K. An implicit gridless type solver for the Navier-Stokes equations. *CFD Journal*, 2000, **9**(1).
 8. Lohner, R.; Sacco, C.; Onate, E. & Idelsohn, S. A finite point method for compressible flows. *Int. J. Num. Meth. Engg.*, 2002, **53**(8), 1765-779.
 9. Sridar, D. & Balakrishnan, N. An upwind finite difference scheme for meshless solvers. *J. Comp. Phy.*, 2003, **189**, 1-29.
 10. Praveen C. & Deshpande, S.M. Kinetic meshless method. Department of Aerospace Engineering, Indian Institute of Science, Bangalore, IISc Fluid Mechanics Report No. FM-10-2003.
 11. Anandhanarayanan, K.; Dhananjay, B.; Dhokrikar; Ramesh, V. & Deshpande, S.M. Grid-free method for 2-D Euler computations using least squares kinetic upwind method. In Proceedings of Third Asian Computational Fluid Dynamics Conference, Bangalore, 1998. Vol. **2**, pp. 390-95.
 12. Ramesh, V. & Deshpande, S.M. Euler computations on arbitrary grids using LSKUM. In Proceedings of first International Conference on CFD, July 2000, Kyoto. pp. 783-84
 13. Praveen, C.; Ghosh, A.K. & Deshpande, S.M. Positivity preservation, stencil selection and applications of LSKUM to 3-D inviscid flows. *Computers Fluids*, 2009, **38** (8), 1481-494.
 14. Varma, U. Mohan; Rao, S.V. Raghurama & Deshpande, S.M. Point generation using quadtree data structure for meshless solvers. Department of Aerospace Engineering, Indian Institute of Science, Bangalore, Fluid Mechanics Report No. FM8. 2003.
 15. Ramesh, V. & Deshpande, S.M. Euler computations of flow past a missile configuration using LSKUM. NAL Internal Document No. NAL PD CF-0112, 2001.
 16. Ramesh, V. Least squares grid-free kinetic upwind method. Indian Institute of Science, Bangalore, 2001. PhD Thesis.
 17. Anandhanarayanan, K. Development and applications of a grid-free kinetic upwind solver to multi-body configurations. Department of Aerospace Engineering, Indian Institute of Science, Bangalore, 2003. PhD Thesis.
 18. Praveen, C. Development and application of kinetic meshless methods for Euler equations. Department of Aerospace Engineering, Indian Institute of Science, Bangalore, 2004. PhD Thesis.
 19. Anandhanarayanan, K.; Nagarathinam, M. & Deshpande, S.M. Development and applications of a grid-free kinetic upwind solver to multi-body configurations. AIAA Paper No. 2005-4628, 2005.
 20. Jameson, A. & Yoon, S. Lower-upper implicit schemes with multiple grids for the euler equations. *AIAA Journal*, 1987, **25**(7), 929-35.
 21. Anandhanarayanan, K.; Nagarathinam, M. & Deshpande, S.M. An entropy variable-based grid-free Euler solver with LU-SGS accelerator. In Proceedings of 24th ICAS Congress, Yokohama, 2004. Paper No. ICAS 2004-2.4.3.
 22. Anandhanarayanan, K.; Nagarathinam, M. & Deshpande, S.M. Parallelisation of a grid-free kinetic upwind solver. AIAA Paper No. 2005-4846, 2005.
 23. Harish, G.; Pavanakumar, M. & Anandhanarayanan, K. Store separation dynamics using grid-free Euler solver. AIAA Paper No. 2006-3650, 2006.
 24. Deshpande, S.M.; Ramesh, V.; Malagai, Keshav & Arora, Konark. Least squares kinetic upwind mesh-free (LSKUM) method. invited review article in Defence Science Journal, 2010
 25. Mandal, J.C. & Deshpande, S.M. Kinetic flux vector splitting for Euler equations. *Compu. Fluids J.*, 1994, **23**, 447-78.
 26. Steger, J.L. & Benek, J.A. On the use of composite grid schemes in computational aerodynamics. *Comp. Meth. Appl. Mech. Engg.*, 1987, **64**, 301-20.
 27. Meakin, R.L. Composite overset structured grids. In Handbook of grid generation, edited by J.F. Thompson, N.P. Weatherill, and B. Soni, CRC Press, New York, 1999. pp. 11.1-11.20.
 28. Anandhanarayanan, K.; Shah, Vaibhav; Krishnamurthy, R. & Chakraborty, Debasis. Numerical exploration of an aerospace vehicle with deflected fins using a grid-free solver. In Proceedings of SAROD-07, Trivananthapuram, November 2007. pp. 536-44.
 29. Stokes, S.; Chappell, J.A. & Leatham, M. Efficient numerical store trajectory prediction for complex aircraft/store configurations. AIAA Paper No. 1999-3712, 1999.
 30. Jenke, L.M. Experimental roll-damping, magnus and static stability characteristics of two slender missile configurations at high angles of attack (0 to 90 Deg.) and Mach numbers 0.2 through 2.5. Arnold Engineering Development Center, Arnold Air force base, Tennessee, Report No. AEDC-TR-76-58. July 1976.

Contributors



Dr K. Anandhanarayanan, received his BTech (Aeronautical Engineering) from MIT, Chennai, in 1991, ME (Guided Missiles) from IAT, Pune, in 1993 and PhD in 2004 from Dept of Aerospace Engineering, Indian Institute of Science, Bangalore. He is working at the Defence Research and Development Laboratory, Hyderabad since 1993. His areas of interests are algorithm development for CFD applications, Grid-free Euler and Navier-Stokes solvers, direct simulation Monte-Carlo-based Euler-Boltzmann coupled solver for nonequilibrium transitional flows and distributed computing. He is a member of various bodies like the AeSI and ASI. He has about 40 publications in several conferences and journals to his credit.