Application of LSKUM to 3-D flows using FAME mesh

Praveen C, S M Deshpande
ARDB CFD Centre
Dept. of Aerospace Engg.
IISc, Bangalore

A K Ghosh
Aeronautical Development Agency
Vimanapura, Bangalore

1 Introduction

LSKUM [1] is a *kinetic meshless* method for the numerical solution of compressible flows governed by Euler and Navier-Stokes equations. LSKUM requires only a distribution of points in the computational domain and does not demand that the points should be capable of being joined to form a grid. In this sense it is said to be a meshless method. The input to the solver consists of local *connectivity* information. Spatial derivatives are approximated using a *least squares* technique on the local connectivity. The advantage of meshless lies in the fact that it is more easier to obtain a point distribution than a grid especially for a complex 3-D configuration. A very fruitful strategy that has been followed by Ramesh et al. [2] and Anandhanarayanan et al. [3] is to generate body-fitted grids around geometrically simple parts; the point distribution is obtained by taking the union of points from all the grids. This is similar to the *chimera* approach but with two important differences: (1) the grids need not be of good quality, and, (2) there is no need for interpolation in the overlapping regions. A similar but more sophisticated method of grid generation is followed in the FAME (Feature Associated Mesh Embedding) [4] technique developed at QinetiQ, UK (formerly DERA). Here, apart from individual grids around geometrically simple parts of the configuration there is a background Cartesian mesh, which is adaptively refined to blend with the body-fitted meshes as seen in figure (1). Only a few layers (4 to 8) of body-fitted mesh normal to the body surface are required in this case since the rest of the computational domain is covered by Cartesian mesh. This makes it more easy to generate the body-fitted meshes. Another distinctive feature of FAME is the presence of local meshes to resolve sharp geometric features like wing-fuselage intersection, trailing-edge line, wing and fuselage tips, etc. Such an approach is ideally suited for meshless methods since all we need is the total distribution of points obtained from all the meshes.

In this paper we describe the application of LSKUM solution technique to point distributions obtained from FAME mesh. The first step in this process is to generate the point distribution since FAME mesh has a cell-based data structure. The second step is to obtain the set of neighbours which form the connectivity. The connectivity is then subjected to many quality tests and modified if necessary. This makes the LSKUM solver more robust and capable of operating on quite general point distributions [5]

2 FAME data structure

FAME consists of two types of data structures corresponding to curvilinear and Cartesian meshes. The curvilinear grids are defined by an implicit (i, j, k) ordering of the grid points. The Cartesian mesh has a more complicated block data structure. Each block consists of $4 \times 4 \times 4$ cells (hence $5 \times 5 \times 5$ vertices). The vertices within a block have an implicit (i, j, k) numbering ranging from -2 to +2. A block may be further refined in which case there might be a smaller block in some of the octants.

For each block, the following information is provided: (1) coordinates of the center of the block and the length of the side, (2) neighbouring blocks, (3) embedded blocks in any of the octants, (4) parent block and the octant of the parent block in which the current block is located, (5) status of 4^3 vertices, whether the vertex is an interior node, an overlapping node or a blanked node.

The other important information in FAME mesh is related to the overlapping region. Since the curvilinear mesh extends only a few layers normal to the body, the outer boundary of the curvilinear mesh has to get data from the Cartesian mesh or even from another curvilinear mesh. Hence for each point on the outer boundary the cell number of another mesh in which it lies is given.

The connectivity required for LSKUM solver is generated using the FAME data structure. In the interior of each grid, the connectivity consists of points from that grid alone. At the boundary of a grid the connectivity will include points from another grid in that region; this connectivity is obtained using the over-lapping information provided with the FAME mesh. We will see that this strategy does not lead to good connectivity and there is a need to completely merge the nodes from all the grids and then choose the connectivity using some search algorithm like octree.

3 Connectivity generation

The data structure of FAME mesh is given in a block or cell format while the LSKUM solver requires a node-based data structure. For point P_o we have to find a set of neighbouring nodes $\{P_j\}$ such that it can be partitioned into six stencils $C_1, C_2, \ldots, C_6$, each of which satisfies the fitness tests outlined in the next section.

The FAME mesh is a composite mesh consisting of several body-fitted grids which overlap with a back-ground Cartesian mesh. The mesh can be divided into three zones for the purpose of connectivity generation: (Z_1) consists of interior nodes of body-fitted mesh, (Z_2) consists of interior nodes of Cartesian mesh, and (Z_3) consists of nodes on outer boundary of a mesh which overlaps with another mesh.

The first step is to generate the node information from the block information. The node information consists of an indexed array of nodes without duplication together with their Cartesian coordinates. Some nodes which fall inside a solid body are removed based on the `iow` flag provided in `m165.iow` file. For body-fitted grids, generating the node information is straight-forward since their data structure is already node-based. For Cartesian grid, the node generation step is slightly involved and the full details are given elsewhere. Two important points that should be kept in mind while generating the nodes for Cartesian mesh are to (1) take account of the staggered storage in the FAME mesh, and (2) take account of hanging nodes. While staggered data structure has been taken into account, the hanging nodes have been removed from the list of nodes due to insufficient information. Once all the nodes are generated and stored in a single array, the stencil selection procedure is applied. The procedure depends on the zone in which the points are located.

Z_1: The points have an implicit (i, j, k) ordering and the neighbours are selected by moving a unit index in each direction, ie., the neighbours for $P_o = (i, j, k)$ are $(i \pm 1, j, k)$, $(i, j \pm 1, k)$, $(i, j, k \pm 1)$, $(i \pm 1, j \pm 1, k)$, $(i \pm 1, j, k \pm 1)$, $(i, j \pm 1, k \pm 1)$, $(i \pm 1, j \pm 1, k \pm 1)$ which makes a total of 26 neighbours.

Z_2: Each point belongs to a Cartesian block of size $4 \times 4 \times 4$ and has its own (i, j, k) indexing. Neighbouring points are picked up by moving a unit index in all three directions as in the body-fitted case. If the point falls on the boundary of the Cartesian block, then points from the neighbouring blocks are picked. The neighbouring blocks are given in the `m165.iow` file. If two neighbours are collinear then the closest neighbour is retained and the other is removed from the connectivity. Also if the total number of neighbours exceeds a certain specified number then some of them are removed from the connectivity based on a distance criterion.

Z_3: These points fall on the boundary of body-fitted or Cartesian grids and their connectivity must be built up with points from other grids present in that region. The overlap information given in the `m165.mesh` file is used for this purpose. For every point P_o in this zone the `m165.mesh` file gives the cell number of another grid in which this node is located. All the eight points P_j which form the vertices of this cell are added to the connectivity of P_o.

4 Tests for fitness of connectivity

The selection of connectivity is the most crucial aspect of meshless methods. The least squares problem will not be solvable if all points in the connectivity lie on a disc. There will be stability problems even if all the points are almost on a disc For LSKUM we require connectivity which is split in all the three coordinate directions and denoted as $C_1, C_2, \ldots, C_6$. The properties of the least squares matrix $A(C)$ [1] which depends on the connectivity C play an important role.

Minimum stencil size: Each of the six stencils $C_1, C_2, \ldots, C_6$ at each point must contain atleast four points, which is the minimum number required for making the least squares problem over-determined. If this condition is not satisfied then additional points are added to the stencil by searching the secondary neighbours, ie., by adding the neighbours of neighbours.

Degeneracy test: The least squares matrix $A(C)$ must be invertible for each of the split-stencils $C_1, C_2, \ldots, C_6$ at each point. This means that the determinant must be non-zero. Degeneracy takes place when all points lie on a disc (straight line in 2-D).

SVD test: The main purpose of SVD (Singular Value Decomposition) test is to detect ill-conditioning of the least squares problem. Ill-conditioning usually occurs when the connectivity is almost disc-type. The determinant is not a good indicator because it can be made arbitrarily small/large by scaling the coordinates. The singular values of $X(C)$ where

$$X(C) = \begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \vdots & \vdots & \vdots \\ \Delta x_N & \Delta y_N & \Delta z_N \end{bmatrix} \tag{1}$$

are also the square root of eigenvalues of $A(C) = X^\top X$ and can be used to calculate the condition number. A large condition number indicates that the system is ill-conditioned. Let $\lambda_{\min}$ and $\lambda_{\max}$ be the smallest and largest singular values of X. If $\lambda_{\min}/\lambda_{\max} < \texttt{SVD\_TOL}$ then we consider the least squares problem to be ill-conditioned. If $\lambda_{\min} = 0$ then the stencil will be flat and the problem is again ill-conditioned. Whenever a particular stencil is ill-conditioned we flag the stencil to full and the corresponding flux derivative is calculated with full stencil. This leads to a loss of upwinding but we expect that stability will not be lost since the other flux derivatives introduce enough numerical dissipation. This is of course borne out by numerical experience. If more than two out of six stencils are switched to full, then we add extra neighbours and set the derivatives back to half stencil since otherwise stability is indeed lost (This was experienced at the fuselage tips where the stencils are very flat).

Polynomial derivative test: The quality of a connectivity is ultimately judged by the accuracy with which derivatives can be computed. If the least squares matrix is invertible then for a linear function we get exact (to machine precision) derivatives. Hence we must use atleast a quadratic polynomial to test the quality of the connectivity. We use a symmetric second order polynomial

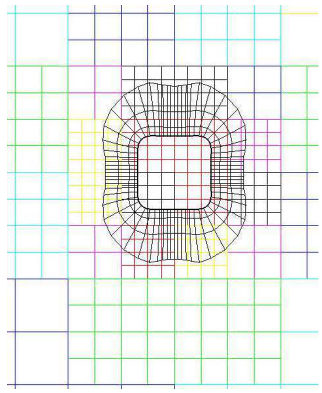
$$p(x, y, z) = x^2 + y^2 + z^2 + xy + yz + zx + x + y + z \tag{2}$$

There are six flux derivatives in the 3-D conservation law and each one is evaluated using a different one-sided connectivity. The least squares formula is applied on the polynomial using each one of these six stencils and the corresponding errors in the derivatives are calculated. Since we are using a first order formula, the error should be order h where h is a measure of the length scale of the connectivity, i.e, error $= Kh$. The quantity K should be small or $O(1)$. If K is greater than some tolerance value we add more points to the corresponding stencil.

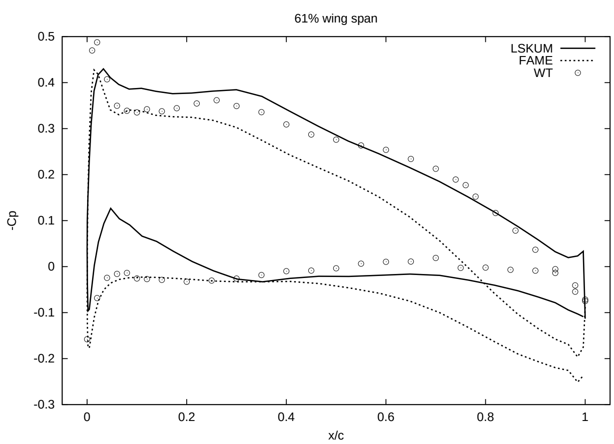
Removal of duplicate nodes: The connectivity of any node cannot contain duplicate nodes, i.e., neighbouring nodes cannot be included more than once. Hence the connectivity is checked for duplicate nodes and corrected if necessary.

5 Results and Conclusions

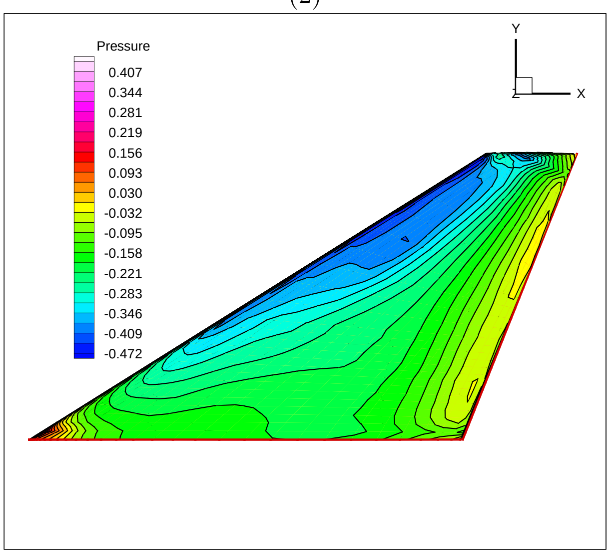
The connectivity generation and LSKUM flow solver are applied to the M165 configuration for which FAME mesh is available. The configuration consists of a fuselage, two canards and two wings. There are six body-fitted meshes and a background Cartesian mesh. The point distribution and connectivity are first generated as described before. The connectivity is passed through all the quality checks. If any test indicates that the connectivity is bad then corrective action like enhancing the connectivity or using full stencil are taken. With these modifications the LSKUM solver is able to operate on the connectivity and gives converged solution. Without these corrective actions LSKUM fails due to poor connectivity leading to negative density and pressure.



(1)



(2)



(3)

Transonic flow over M165 configuration at $M_\infty = 0.789$ and angle of attack 4.36 deg is computed using explicit Euler scheme with local time-stepping. The results are compared with numerical solutions using a grid-based scheme and interpolation in over-lapping regions given by QinetiQ, UK, and also with wind-tunnel data. In figure (2) the pressure coefficient at 61% of wing-span is shown which indicates that LSKUM gives better results and compares well with experimental data. The discrepancy in the leading edge region is thought to be due to large support of stencil and further work is required to resolve this issue. In figure (3) the pressure contours on the top surface of the wing are shown.

A pre-processor has been developed to generate point distributions from FAME mesh and obtain connectivity information for these points. Several quality tests for connectivity have been developed which are used to automatically make modifications to the connectivity making the solution process more robust. Application of these techniques to a 3-D test problem has demonstrated good accuracy and robustness of this meshless approach.

Acknowledgments: This work was conducted under the ADA sponsored project on *LSKUM-FAME Integration for Store Trajectory Prediction* which is a collaborative effort between IISc, ADA and QinetiQ. The authors thank QinetiQ for providing the FAME mesh used in these computations.

References

(1) Ghosh A K, PhD thesis, IISc, 1996.
(2) Ramesh V, PhD thesis, IISc, 2003.
(3) Anandhanarayanan, PhD thesis, IISc, 2004.
(4) Albone C M, AIAA paper 92-0662, 1992.
(5) Project report on LSKUM-FAME Integration, 2004-FM-11, IISc.