Line Generation for General Unstructured Grids

I. Background

A line generation algorithm for general unstructured grids has been developed and implemented within USM3D. An overview and terminology of the line generation algorithm is provided in the main body of the article. In this supplement, sections II and III below describe the two main steps of the line generation process. Section IV assesses the line generation method for three unstructured grids composed of different types of cells.

II. Pairing between current- and next-layer faces

USM3D generates lines using a recursive advancing face-layer algorithm. The core procedure of the algorithm is to pair faces that are located at the current and next face layers. The procedure inputs the faces of the current face layer and outputs the faces of the next face layer. A next-layer face must satisfy three conditions, (1) the face is paired with a current-layer face, (2) the face belongs to a cell that has at least one node on the current-layer face it is paired with, and (3) the face has no nodes on either the current layer of nodes or any layer of nodes preceding the current layer. After pairing the layer faces, the cells between the paired layer faces are added to the corresponding lines, and the next-layer faces are designated as the current-layer faces for the advancement of lines. The next face layer may have fewer faces than the current face layer. If a current-layer face cannot be paired with any next-layer face, it becomes the terminal face of the line.

At the beginning of the core procedure, the following preliminary steps are performed:

Step 1. Using the current-layer faces, all the current-layer nodes are defined.

Step 2. A pool of potential next-layer cells is established by identifying all cells attached to a current-layer node. The pool does not accept any cell that has nodes on any layer of nodes preceding the current layer. This condition helps exclude the cells that have already been assigned to a line.

Step 3. A pool of potential next-layer faces is established by examining the faces of the potential next-layer cells.

Step 4. All nodes of the potential next-layer faces make up a pool of prospective next-layer nodes.

Step 5. A face-neighbor data structure is defined that identifies for each potential next-layer face all the faces in the pool of potential next-layer faces that share an edge with the face under consideration. A similar face-neighbor data structure for the current-layer faces is preexisting due to the recursive nature of the algorithm.

In the core procedure, nonpaired current-layer faces are visited one by one in an attempt to find a unique pair with a nonpaired face in the pool of potential next-layer faces defined in Step 3 above. An advancing-front algorithm defines the specific order in which nonpaired current-layer faces are visited. The details of the advancing-front algorithm are presented in Section III below.

The face-pairing procedure for a given nonpaired current-layer face, f_c is described next. First, the unique potential next-layer cell that is attached to f_c is identified. This cell is a member of the pool of cells defined in Step 2 above.

All potential next-layer nodes of this cell are identified. These nodes form a subset of the pool of nodes defined in Step 4. The nonpaired potential next-layer faces attached to the subset of nodes are designated as candidate faces, i.e., these potential next-layer faces are the candidates to pair with f_c. The faces that do not have the same number of nodes as that of f_c are removed from the set of candidate faces, i.e., pairing between triangular and quadrilateral faces is not allowed. Next, candidate faces are evaluated one by one. For each candidate face, f_n, an ordered sequence of potential next-layer cells (e.g., a single hexahedral or prismatic cell, two side prisms, or three tetrahedral cells) is sought. The sequence has to satisfy the following four conditions:

a. The first cell in the sequence is attached to f_c.
b. The last cell in the sequence is attached to f_n.
c. If the sequence has two cells, then the first cell will be a face neighbor to the last cell.
d. If the sequence has three cells, then the middle cell in the sequence is a face neighbor to the first cell and to the last cell. Note that on tetrahedral grids, up to five cells can have all their nodes on f_c and f_n, but only three cells can be included in a sequence.

Only those candidate faces for which such a sequence can be found are retained as candidate faces. If after evaluating all candidate faces, the set of candidate faces is empty, f_c is declared as the terminal face for the corresponding line. If there is exactly one candidate face, f_n, then the face is paired with f_c. Several different sequences satisfying the four conditions may exist and be associated with one or more candidate faces. If sequences include a different number of cells, then the sequence containing the fewest cells is chosen. If several candidate faces are left that have sequences with identical number of cells, then the following rules govern the selection of the candidate face to be chosen for pairing with f_c. The rules are applied in the order listed next. (1) If f_c touches physical boundary patches, then the preference is given to a candidate face that touches the same physical boundary patches. If no candidate face touches all the physical boundary patches associated with f_c, then the candidate that touches the most physical boundary patches touched by f_c is chosen. (2) If f_c has a paired current-layer neighbor, f_{cp}, then the preference is given to the candidate face, f_n, that neighbors f_{np}, the next-layer face that has been paired with f_{cp}. (3) If preferences (1) and (2) cannot uniquely resolve pairing for f_c, then a face, f_n, is randomly chosen from the remaining candidate faces to pair with the face f_c. When f_c is successfully paired with f_n, the sequence cells between f_c and f_n are added to the cells of the line in the order in which they connect f_c and f_n. Only one set of sequence cells that connect f_c and f_n is added. No cells are added to the line, if f_c is designated as a terminal face.

The pairing process can be illustrated by the following two examples based on Fig. 4(b) in the main body of the article. In the first example, we consider that the advancing layer algorithm is moving from the second global layer of faces that includes faces (5, 6), (6, 7), and (7, 8) to the next layer of faces. The faces (9, 10), (10, 11) and (11, 12) are identified as potential next-layer faces. The nodes 9, 10, 11, 12 form a pool of potential next-layer nodes. We assume that faces (5, 6) and (9, 10) have already been paired and the current-layer face f_c considered for pairing is face (6, 7). Then, the potential next-layer cell attached to f_c is (6, 7, 11). This cell has the only potential next-layer node 11. The candidate nonpaired next-layer faces associated with node 11 are (10, 11) and (11, 12). Both the candidate faces have a two-cell sequence connecting them to f_c and equally satisfy the four selection conditions

(conditions a.–d. listed above). The current-layer face f_c does not touch a physical boundary. Based on the previous pairing, the candidate face (10, 11) is preferred as it neighbors the face (9, 10) which has been paired with the current-layer face (5, 6). Thus, faces (6, 7) and (10, 11) are paired and cells (6, 7, 11) and (6, 11, 10) are added to the line.

In the second example, we consider pairing of face (16, 13). The only cell attached to this face that is not already in a line is (16, 13, 11, 17). This cell cannot be a next-layer cell as it has node 11 which belongs to a preceding layer. Face (16, 13) becomes terminal for its line.

III. Sequencing of current-layer faces for pairing

An advancing front algorithm is used to assign the order in which current-layer faces are considered for pairing. For this purpose, a two-dimensional (2D) manifold concept is introduced. All current-layer faces that have been neither paired nor defined as terminal form a 2D manifold. The 2D manifold can be simply-connected or composed of disjoint segments. The advancing front is composed of so-called perimeter faces, which are the nonpaired faces attached to the perimeter of the 2D manifold. That is, the front includes those current-layer nonpaired faces that neighbor either a paired or terminal current-layer face or have an edge on the perimeter of the 2D manifold. The face-neighbor data structure defined earlier (Step 5 in section II) is used for the identification of perimeter faces. Formally, the perimeter faces include nonpaired current-layer triangular faces with fewer than three nonpaired current-layer face neighbors and nonpaired current-layer quadrilateral faces with fewer than four nonpaired current-layer face neighbors.

To minimize random-choice pairing, the faces with the fewest nonpaired neighbors are considered for pairing first. The perimeter faces are placed in the front in an ascending order, as per the number of nonpaired face-neighbors. An additional advantage of such an ordering is that it allows face-pairing in one pass through the front. If the front is not empty, then the first face in the front is chosen for pairing. The front is continuously updated. After the first front face has been resolved (that is, either paired or defined as terminal), it is removed from the front, and all its nonpaired current-layer face neighbors are promoted in the front as they have one less nonpaired current-layer neighbors than before.

The formation of a global advancing layer of faces is a fundamental step of the present line-generation algorithm. The lines are typically generated from a set of faces on no-slip aerodynamic surface. In parallel computations, each grid partition may contain only a part of a no-slip aerodynamic surface and it may lack the information about the grid connectivity in the other partitions. Therefore, identification of global layers is a formidable task in any parallel framework. In the present work, the line generation is accomplished in a sequential preprocessing step. For parallel computations, the grid partitioning procedure is enhanced such that an entire grid line is fully contained within a partition.

IV. Performance of line-generation algorithm

To demonstrate the performance of the USM3D line-generation algorithm, three grids around the hemisphere-cylinder (HC) and a NASA Common Research Model (CRM) configuration have been considered. Table A shows the grid statistics. Table B shows grid-line statistics and the computational time required to generate these lines. Both grids around the HC configuration are entirely composed of advancing layers. The line generation algorithm follows the advancing layers and generates lines that include all cells in both the HC grids. The number of lines equals the number of faces on the surface of the hemisphere cylinder. All lines have the same number of cells. The line generation procedure takes about 68 seconds on the tetrahedral HC grid and about 650 seconds on the mixed-element HC grid. The NASA CRM mixed-element grid is more representative of practical grids. Line generation on this grid takes about 24 seconds. Approximately 88% of the cells are included in lines. The lines include prismatic and tetrahedral cells. Lines vary significantly in length; the shortest line includes 13 cells and the longest line includes 98 cells.

Table A. Grid statistics.

Configuration	Tetrahedra	Pyramids	Prisms	Total cells	Nodes
HC, tetrahedral	6,635,520	0	0	6,635,520	1,143,081
HC, mixed	34,753,536	0	6,110,208	40,863,744	8,995,153
NASA CRM, mixed	271,331	26,728	1,226,266	1,524,325	685,941

Table B. Line statistics.

Configuration	Lines	Line cells	Cells per line		CPU time,
			min	max	sec
HC, tetrahedral	6,912	6,635,520	960	960	67.66
HC, mixed	27,648	40,863,744	1,478	1,478	650.45
NASA CRM, mixed	38,979	1,345,695	13	98	24.48