



Development of Three-Dimensional DRAGON Grid Technology

Yao Zheng
Taitech, Inc., Cleveland, Ohio

Meng-Sing Liou and Kestutis C. Civinskas
Glenn Research Center, Cleveland, Ohio

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at (301) 621-0134
- Telephone the NASA Access Help Desk at (301) 621-0390
- Write to:
NASA Access Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076



Development of Three-Dimensional DRAGON Grid Technology

Yao Zheng
Taitech, Inc., Cleveland, Ohio

Meng-Sing Liou and Kestutis C. Civinskas
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

Acknowledgments

We thank Dr. James D. Heidmann for kindly providing us the geometry of the film-cooled turbine vane.
We also thank Dr. Kumud Ajmani for generating the Chimera grids for the compressor drum cavity
and the turbine branch-duct.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A03

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A03

Development of Three-Dimensional DRAGON Grid Technology

Yao Zheng

*Taitech, Inc.
Cleveland, Ohio*

Meng-Sing Liou and Kestutis C. Civinskas

*National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio*

Abstract

For a typical three dimensional flow in a practical engineering device, the time spent in grid generation can take 70% of the total analysis effort, resulting in a serious bottleneck in the design/analysis cycle. The present research attempts to develop a procedure that can considerably reduce the grid generation effort.

The DRAGON grid, as a hybrid grid, is created by means of a Direct Replacement of Arbitrary Grid Overlapping by Nonstructured grid. The DRAGON grid scheme is an adaptation to the Chimera thinking. The Chimera grid is a composite structured grid, composing a set of overlapped structured grids, which are independently generated and body-fitted. The grid is of high quality and amenable for efficient solution schemes. However, the interpolation used in the overlapped region between grids introduces error, especially when a sharp-gradient region is encountered. The DRAGON grid scheme is capable of completely eliminating the interpolation and preserving the conservation property. It maximizes the advantages of the Chimera scheme and adapts the strengths of the unstructured grid while at the same time keeping its weaknesses minimal.

In the present paper, we describe the progress towards extending the DRAGON grid technology into three dimensions. Essential and programming aspects of the extension, and new challenges for the three-dimensional cases, are addressed.

1 Introduction

During the past decades, both structured and unstructured grid techniques have been developed and applied to solution of various computational fluid dynamics (CFD) problems.

To deal with situations in which complex geometry imposes great constraints and difficulties in generating grids, composite structured grid schemes and unstructured grid schemes currently are the two mainstream approaches.

The Chimera grid scheme[2, 8, 33] and similar scheme [1, 7, 9] use overset grids to resolve complex geometries or flow features, and are generally classified into the composite structured grid category. Overset grids allow structured grids to be used without excessive distortion or inefficient control of grid spacing. It has been used to compute inviscid and high-Reynolds flow about complex configuration [8, 33], and even been demonstrated for unsteady three-dimensional viscous flow problems [10, 23] in which one object moves with respect to the other. Overset grids have also been used as a solution adaptation procedure [4, 17, 22].

The Chimera method is an outgrowth of the attempt to generalize a powerful solution approach (the structured and body-conforming grid method) to more complex situations. It is recognized, however, that there are weaknesses that may be of concern. There are two main criticisms leveled against the current implementations of the Chimera method: (1) the complexity of the interconnectivity is often difficult, requiring extensive experience to avoid orphan points and interpolation stencils of bad quality, and (2) nonconservative interpolations to update interface boundaries are used in practical cases. The fact that interpolation is used to connect grids with no regard to governing conservation equations in question implies that conservation is not strictly enforced.

Our experiences have indicated that the interpolation error can become globally significant if numerical fluxes are not fully conserved, in particular when a discontinuity runs through the interpolated region. Also, this error can be strongly affected by the underlying flux schemes, such as central vs. upwind schemes. It must be noted that conservative interpolation schemes in two dimensions [24, 37] have been shown to be relatively easy to implement. Extension to three dimensions for irregular polyhedra, however, is not all that straightforward. Even if the three-dimensional conservative interpolation proves to be feasible, a fundamental difficulty exists because the distribution of the coarse-grid data to the fine grid is not unique.

On the other hand, the unstructured grid method is found to be very flexible to generate grids around complex geometries [5, 15, 29]. In particular, the solution adaptivity is perhaps its greatest strength[21, 28]. However, the unstructured grid method has been shown to be extremely memory and computation intensive[12]. Also, choices of efficient flow solvers are limited, thus further affecting computation efficiency of the method. In practice, it seems less amenable than the structured grid to implement a scheme higher than second order accurate in space.

In general, unstructured grid methods are considered to be more versatile and easier to adapt to complex geometries while composite structured grid methods are considered to be more flexible to use efficient numerical algorithms and require less computer memory. Clearly, both methods complement each other on strengths and weaknesses.

Hence, a method that properly employs a hybrid of structured and unstructured grids may prove to be fruitful. In fact some hybrid schemes have already appeared[14, 25, 31,

Table 1: Comparison of gridding methodologies.

Gridding methods	Features
Chimera grid	+ Ease, versatility and flexibility for complex shapes + Efficient solution algorithms + Less memory requirement – Complexity of interconnectivity – Nonconservative interpolation
Unstructured grid	+ Flexibility for complex shapes + Solution adaptivity – Less choices of efficient solvers – Memory and computation intensiveness
DRAGON grid	+ Ease, versatility and flexibility for complex shapes + Efficient solution algorithms + Less memory requirement + Grid communication in a conservative and consistent manner

38] and have shown promising features. Interestingly enough, researchers from each camp have infused ideas from the other to approach a hybrid grid: a structured grid is embedded underneath an otherwise unstructured grid in order to better resolve the viscous region[14, 31], or an unstructured grid is used to enhance the flexibility of structured grid scheme for handling complex geometries[25, 38]. Nevertheless, there is an important difference between the above two approaches toward hybrid grids. On the one hand, a majority of region is filled with unstructured grids, while on the other the region is mostly structured. One can readily conclude that the latter has more efficient flow solvers and utilizes less memory, thereby resulting in fast turnaround.

The DRAGON grid [16, 20], as a hybrid grid, is created by means of a Direct Replacement of Arbitrary Grid Overlapping by Nonstructured grid. The DRAGON grid scheme is an adaptation to the Chimera thinking. It has three important advantages: (1) preserving strengths of the Chimera grid, (2) eliminating difficulties sometimes encountered in the Chimera scheme, and (3) enabling grid communication in a fully conservative and consistent manner insofar as the governing equations are concerned. Table 1 summarizes the features of Chimera grid, unstructured grid, and DRAGON grid schemes.

In the present work, we have attempted to develop an extension of the DRAGON grid technology into three-dimensional space. We are interested in simulating flows over three-dimensional complex objects with complicated physical behavior. A propulsion system is an example of such geometries, and generally involves many separate geometrical entities (multi-connected domain) with odd shapes and sharp turns. This topology creates difficulties and challenges to grid generation, especially for viscous flow calculations. For a typical three-dimensional flow calculation, the time spent in generating a grid takes about two thirds of the total analytical effort, representing a serious bottleneck in the entire analysis cycle, see for example [35]. This suggests that grid generation is the area where significant payoff can

be realized. Furthermore, high quality grids for encompassing viscous regions will lead to accurate and efficient solution. These are the basic requirements that the DRAGON grid technology can achieve. The DRAGON gridding scheme maximizes the advantages of the Chimera scheme, and adopts the strengths of the unstructured grid while minimizing its weakness [16, 20].

In the following text, we will first outline the Chimera grid methodology [32], then address the essentials of the DRAGON grid methodology [20]. Finally, we will concentrate on the extension of the DRAGON grid technology into three-dimensional space. Essential and programming aspects of the extension, and new challenges for the three-dimensional cases, are to be presented.

2 Outline of Chimera Grid Methodology

The Chimera and other like methods have two principal elements: (1) decomposition of a chosen computation domain into subdomains, and (2) communications of solution data among these subdomains through an interpolation procedure. Software is needed to automatically interconnect grids of subdomains, define the holed region, and supply pointers to facilitate communication among grids during the solution process. Two major codes in this area are PEGSUS[2, 34] and CMPGRD[7, 9]. We have used the PEGSUS code[2, 34] to perform the above task. The CMPGRD code has similar capabilities, but uses different and more involved interpolation strategies. Hereafter, we will specifically discuss the Chimera grid within the capability of the PEGSUS codes. PEGSUS 4.0[34] executes four basic tasks: (1) process the grids and user inputs for all subdomains, (2) identify the hole and interpolation boundary points, (3) determine the interpolation stencil and interpolation coefficients for each interpolation boundary point, and (4) supply diagnostic information on the execution and output the results for input to the flow solver.

2.1 Generation of Body-Fitted Grids

The Chimera grid method makes use of body-fitted grids, which we emphasize, are an essential asset known to give viscous solutions accurately and economically, see [14, 31] for example. In the gridding process, the complete geometrical model is divided up into subdomains, which in general are associated with components of a configuration. Each subdomain is gridded independently and is required to have overlapped regions between subdomains. The grid boundaries are not required to join in any special way. A number of grids can be introduced to focus on interesting geometrical or physical features. Figure 1 illustrates the Chimera grid for the complex geometry of the integrated space shuttle vehicle [23], in which we show three grids conforming respectively the orbiter, external tank and solid rocket booster together with the associated hole boundaries, representing an impressive capability of the method. A common or overlapped region is always required to provide the means of matching solutions across boundary interfaces.

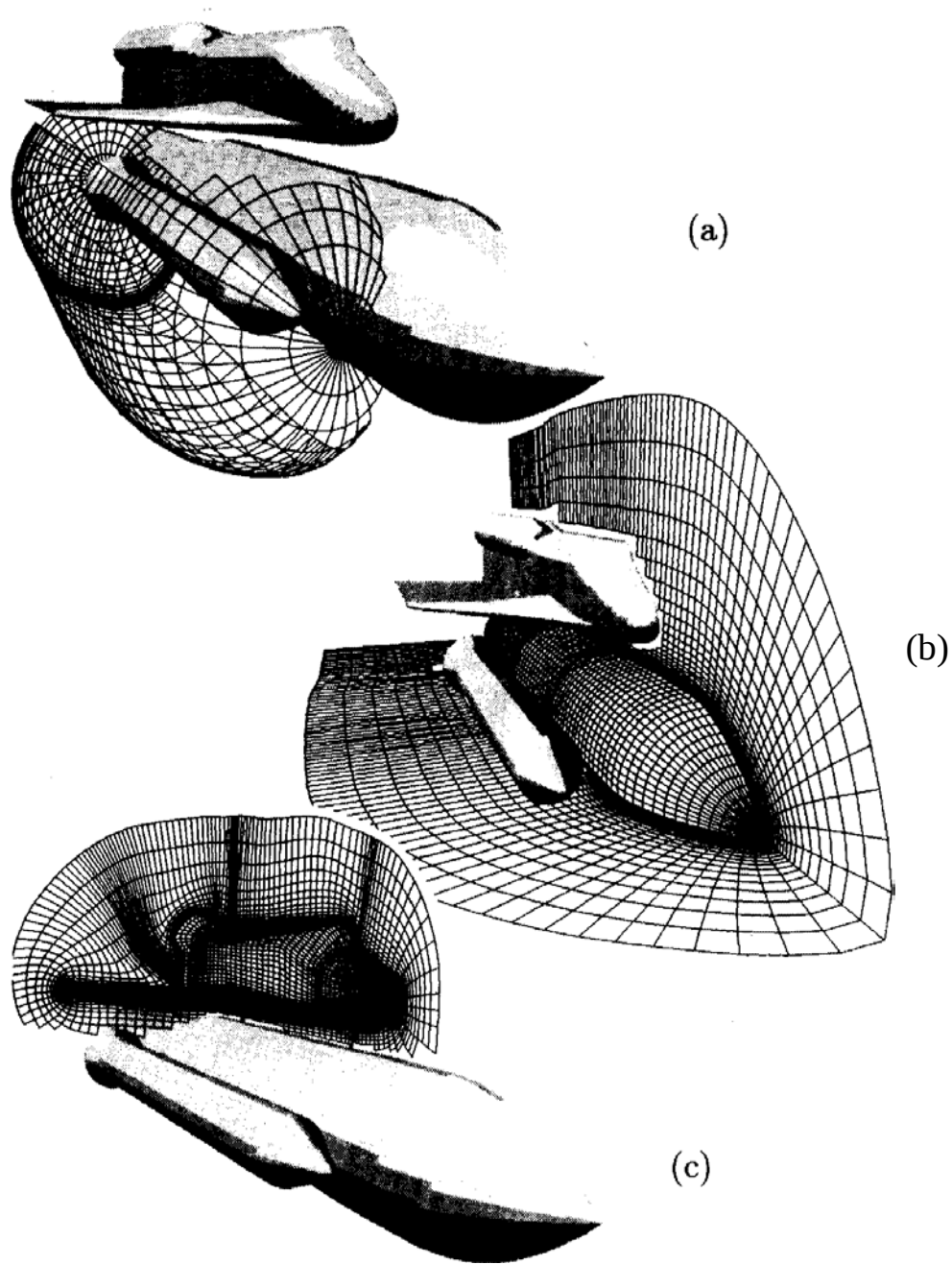


Figure 1: Chimera grid used for integrated space shuttle geometry consisting of orbiter, external tank (ET), and solid rocket booster (SRB): (a) SRB grid with ET and orbiter holes. (b) ET grid with SRB and orbiter holes, and (c) orbiter grid with ET and SRB holes.

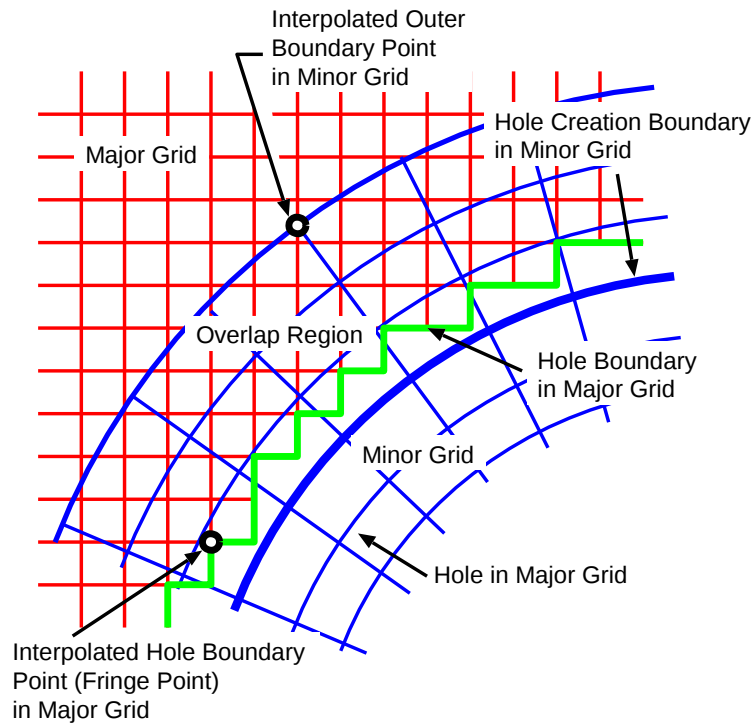


Figure 2: Construct of the Chimera grid.

Hole boundaries and outer boundaries are the two ways through which information is communicated from one grid to another. A novel approach used in the Chimera method to distinguish a hole point and an outer boundary point from a field point is to flag the IBLANK array, which is dimensioned identically to the number of points in each grid, to either 1 for a field point or 0 otherwise. The boundary points with IBLANK = 0 are to be updated by interpolation, while points with IBLANK = 1 are updated as usual for the interior points.

We summarize the grid-related portion of the Chimera method in Figure 2 by displaying: (1) the overlapping of two grids, respectively designated as major and minor grids, with the latter being thought of as conforming to a component, (2) the hole creation boundary specified as a level surface in the minor grid and the fringe-point boundary in the major grid, and (3) the outer boundary in the minor grid.

2.2 Data Communication

Since the separate grids are to be solved independently, boundary conditions for each grid must be made available. Boundary conditions on the interpolated hole boundary and interpolated outer boundary are supplied from the grid in which the boundaries are contained. There are currently several approaches (e.g., [2, 3, 7, 24, 30, 37]) to obtain data for these conditions, but all involve some form of interpolation of data in a grid. Generally they

can be grouped into two categories: nonconservative and conservative interpolations, which are discussed in the following.

2.2.1 Nonconservative Interpolation

Once the interpolation stencils are searched and identified, PEGSUS procedure employs a nonconservative trilinear interpolation scheme[2]. It is unclear to what extent the nonconservative interpolation will affect the solution, locally or globally, especially when a strong-gradient region intersects the interpolated region. Study on this subject has been scarce in the literature. Our experience, such as [20], indicates that significant error can appear. For steady problems, a shock may be placed at an incorrect location and noticeable spurious waves can emanate from the boundary of the interpolated region. For unsteady problems[20], the shock strength and speed is changed as the shock goes through the region of interpolation. In Reference [9], the order of interpolation has been studied in relation to the order of the partial differential equations (PDE), the order of the discretization, and the width of interpolation stencils for a model boundary value problem. For a second order differential equation discretized with a second order formula, it was found necessary to use an interpolant at least of third order, as the overlap width is on the order of the grid size. A critical assessment of their proposal's validity for a range of problems is warranted because interpolating has the advantage of being relatively simple matter to perform.

2.2.2 Conservative Interpolation

It has been asserted that the conservation property needs to be enforced for cases in which shock waves or other high-gradient regions intersect the region of interpolation. Several conservative interpolation schemes have been proposed for patched interfaces[4, 30] and arbitrarily overlapped regions[24]. These schemes are relatively easy in a two-dimensional domain, but still substantially more complex than the nonconservative schemes. Their extensions to three dimensions are extremely difficult and still remain to be done. Simplification is possible[18, 36] for patched grids in three dimensions, but they come with more restrictions on grid generation, making them less attractive for practical use.

A fundamental deficiency in all these approaches is that a choice has to be made concerning the distribution of fluxes from one grid to another, even though the sum of fluxes can be made conservative. Since the overlapped region is necessarily arbitrary, there will be great disparity in grid spacing and orientation along the overlapped region (or hole boundary). The choice of weighting formulas is not clear and certainly not unique.

3 Essentials of DRAGON Grid Methodology

We conclude from the previous section that (1) the property of maintaining grid flexibility and the quality of the Chimera method are definitely to be preserved and even max-

imized (improved), and (2) focusing on improving (choosing) interpolation schemes perhaps only leads to more complication and it does not seem to be a fruitful way to follow.

An alternative method which avoids interpolation altogether and strictly enforces flux conservation for both steady and unsteady problems is to **solve the region in question on the same basis as the rest of the domain**.

Since the overlapped region (thus, the hole boundary) is necessarily irregular in shape, the unstructured grid method is most suitable for gridding up this region. Furthermore, this region is in general away from the body where the viscous effect is less important than the inviscid effect, and coarse grid would suffice as far as solution accuracy is concerned. This situation would be amenable to using the unstructured grid, thus minimizing its penalty associated with memory requirements. The combination of both types of grids results in a hybrid grid. In this approach, we in effect Directly Replace Arbitrary Grid Overlapping by a Non-structured grid [20]. The resulting grid is thus termed the DRAGON grid. Major differences of our approach from other hybrid methods are: (1) We heavily utilize the proven Chimera method and the powerful and versatile automatic code PEGSUS, thus retaining attractive features already described in the Introduction. (2) We use unstructured grids only in limited regions which are mostly located away from viscous-dominant regions, thus minimizing disadvantages of unstructured grids insofar as memory and efficiency are concerned.

In what follows we will separately describe the structured and unstructured grid features of the DRAGON grid method. In this section, two-dimensional topology is assumed in order to simplify the description of this methodology, but without loss of generality.

3.1 Structured Grid Region

The PEGSUS code now is used and modified to provide information necessary for the DRAGON grid. The algorithmic steps are enumerated as follows.

1. As in the Chimera grid, the entire computation domain is divided up into subdomains. We often designate a major (or background) grid enclosing the complete computational domain and the component grids as minor grids.
2. Hole regions are created, referring to Figure 3(a), and the IBLANK array is generated, in which the elements associated with grid points inside the hole boundary are set to 0 (Default value is 1). For example, the outer boundary of a minor grid may be used as the hole creation boundary.
3. Both fringe points (hole boundary points) and interpolation boundary points are no longer treated as blanked points. Instead, they are now represented as interior points, and their IBLANK values are set to 1 (rather than 0) in the PEGSUS code.
4. The fringe points as well as the interpolation boundary points are now forming the new boundaries for the unstructured grid region.

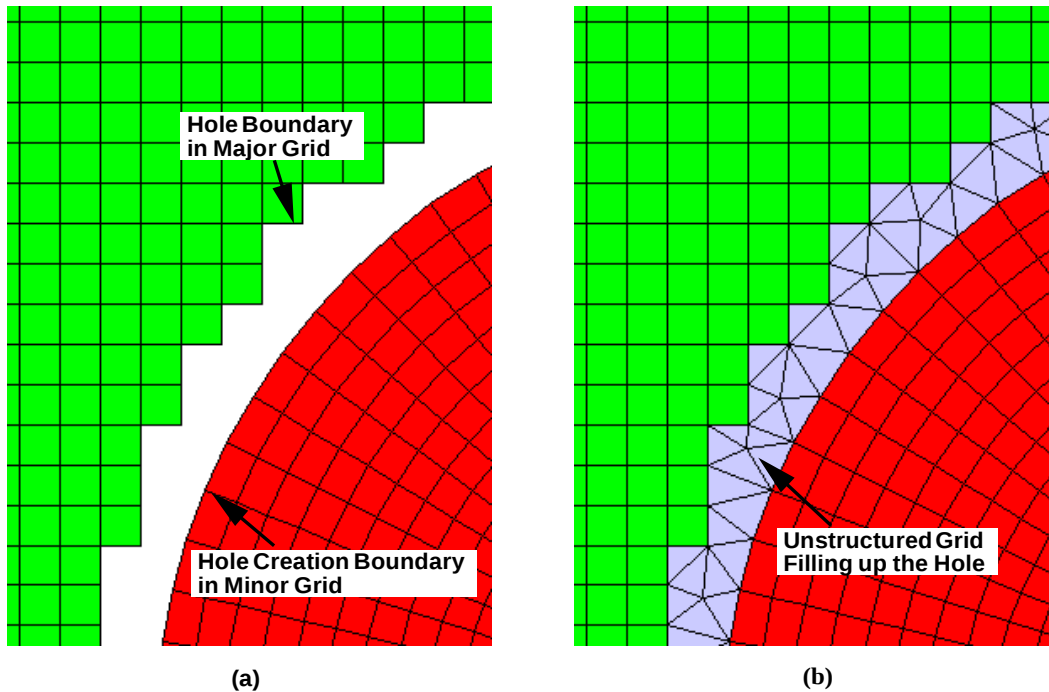


Figure 3: Direct Replacement of Arbitrary Grid Overlapping by Non-structured grid: (a) Chimera grid, (b) DRAGON grid.

It is noted that there is no reason that grids are to be overlapped under the DRAGON grid framework. If grids are not overlapped, the void space is likewise filled with an unstructured grid as in the hole region. Thus this in a sense results in a more robust and flexible procedure than the Chimera grid method. We remark that since the interpolation process is no longer performed between structured grid blocks, output files providing interpolation information are deleted from the PEGSUS code. This completes the portion of structured grid in the DRAGON grid.

3.2 Non-Structured Grid Region

The gap region created by arbitrarily overset grids is inevitably of irregular shape. Unstructured grid can lend its strength readily to handle this irregularly shaped space. Triangular cells, especially, can provide a good deal of flexibility to adapt to the odd shape. Recall that one important feature in the DRAGON grid is to eliminate any cumbersome interpolation which causes nonconservation of fluxes. Unstructured grids alone are not sufficient to do the task. An additional constraint to the grid generation is imposed to require that the boundary nodes of the structured grid coincide with vertices of boundary triangular cells. In other words, there might be more than one triangle connected to one structured grid cell, but not vice versa (Refer to Figures 3 and 6). Fortunately, this constraint fits well in unstructured grid generation. The Delaunay triangulation scheme [6, 39, 40] is applied to generate an unstructured grid in the present paper. Figure 3(b) depicts the unstructured grid

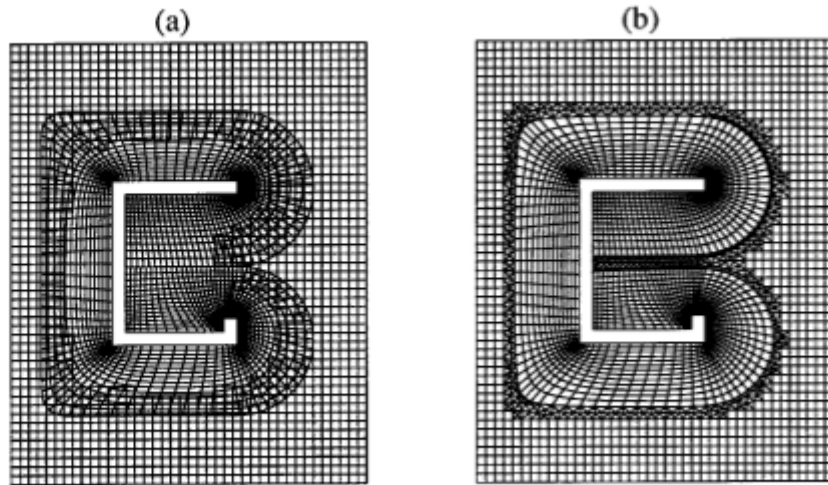


Figure 4: Comparison of the (a) Chimera and (b) DRAGON grids for the letter C in a channel.

filling up the hole created in the structured grid by the Chimera method. The steps adopting the unstructured cells in the framework of the Chimera grid scheme are summarized below.

1. Boundary nodes provided by the PEGSUS code are reordered according to their geometric coordinates.
2. Delaunay triangulation method is then performed to connect these boundary nodes, for example, based on the Bowyer algorithm[6].
3. In the unstructured grid, since there is no logical ordering of the cells and their neighbors, connectivity matrices containing the cell-based as well as edge-based information are introduced. Also the present approach requires additional matrices to describe the connection between the structured and unstructured grids.

This step completes the specification of the initial grid. Figure 4 now compares the DRAGON grid and the Chimera grid for the letter C in a channel. The Chimera method wraps around the C with a body-conforming (minor) grid and oversets it on the background (major) H-type grid. Note that we intentionally make an asymmetric C to emphasize grid flexibility.

3.3 Data Communication through Grid Interfaces

In the Chimera method, this communication is made through the hole boundary or the outer boundary. Since the interface treatment methods do not necessarily satisfy any form of conservative constraints, the solutions on overlaid grids are often mismatched with each

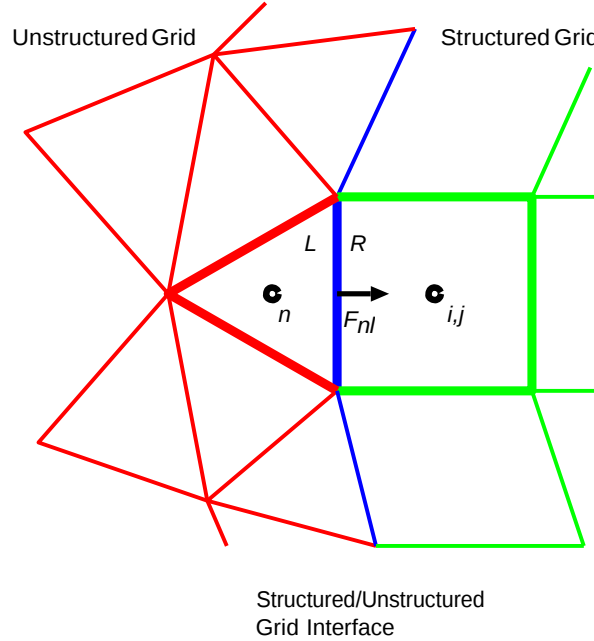


Figure 5: Fluxes at the cell face connecting the structured and unstructured grids.

other. More seriously, this may ultimately lead to spurious or incorrect solutions, especially when a shock wave or high-gradient region passes through boundaries of overlaid grids.

For the DRAGON grid, conservation laws are solved on the same basis on both the structured and unstructured grids. The cell center scheme is used for storage of variables as well as flux evaluations, in which the quadrilateral and triangular cells are used respectively. Figure 5 shows the interfaces connecting both structured and unstructured grids. As described earlier, the numerical fluxes, evaluated at the cell interface, are based on the conditions of neighboring cells (denoted as L and R cells, respectively). For the unstructured grid, the interface flux F_{nl} , will be evaluated using the structured-cell value as the right (R) state and the unstructured-cell value as the left (L) state. Consequently, the interface fluxes which have been evaluated in the unstructured process can now be directly applied in computing the cell volume residuals for the structured grid.

Thus, the communication in the DRAGON grid is considered seamless in the sense that no manipulation of data, other than satisfying of the conservation laws, is required. The solution is obtained on the same basis whether it is in the structured or unstructured grid region. Consequently this strictly enforces conservation property locally and globally.

4 Three-Dimensional DRAGON Grid Technology

The extension of the concept of DRAGON grid technology in two dimensions, to that in three-dimensional space, is straightforward. However, more difficulties are anticipated in

an actual implementation. Unique challenges for the case of three dimensions exist in various aspects, such as algorithms and code implementation. Research and development has been conducted to establish the three-dimensional DRAGON gridding methodology, in light of previous experiences on two-dimensional DRAGON [16, 20], three-dimensional Chimera and unstructured gridding [19, 39, 40].

With reference to Figure 3, it is obvious that boundary edges of the structured grid coincide with edges of boundary triangular cells in two dimensions, whereas faces of the structured grid do not match the faces of the unstructured grid in the same location for three-dimensional cases as illustrated in Figures 6 and 7. In Figure 7, Points 1, 2, 3, 4, 5 and 6 on the structured grid are coincident with the corresponding ones on the unstructured grid. Therefore, a scheme can be easily developed to meet flux conservation at common faces without resorting to interpolation of fluxes. Occasionally, there might be areas of different grid spacings in an interfacial region between the structured and unstructured grids, as shown in Figure 6(b). This is due to the requirements in the grid generation stage, and/or those to reflect the associated physical phenomena.

As an example, Figures 8 and 9 depict a DRAGON grid for the compressor drum cavity problem. This grid is created from a base structured grid, where there are both overlapped domains and voids. The structured grid is body-fitted, and of high quality for viscous flow simulation. The easily generated unstructured grid fills the overlapped domains and voids. Figure 8 provide a sideview of the DRAGON grid, where the outer rim barely reveals the third dimension. Meanwhile Figure 9 presents the split view of the DRAGON grid.

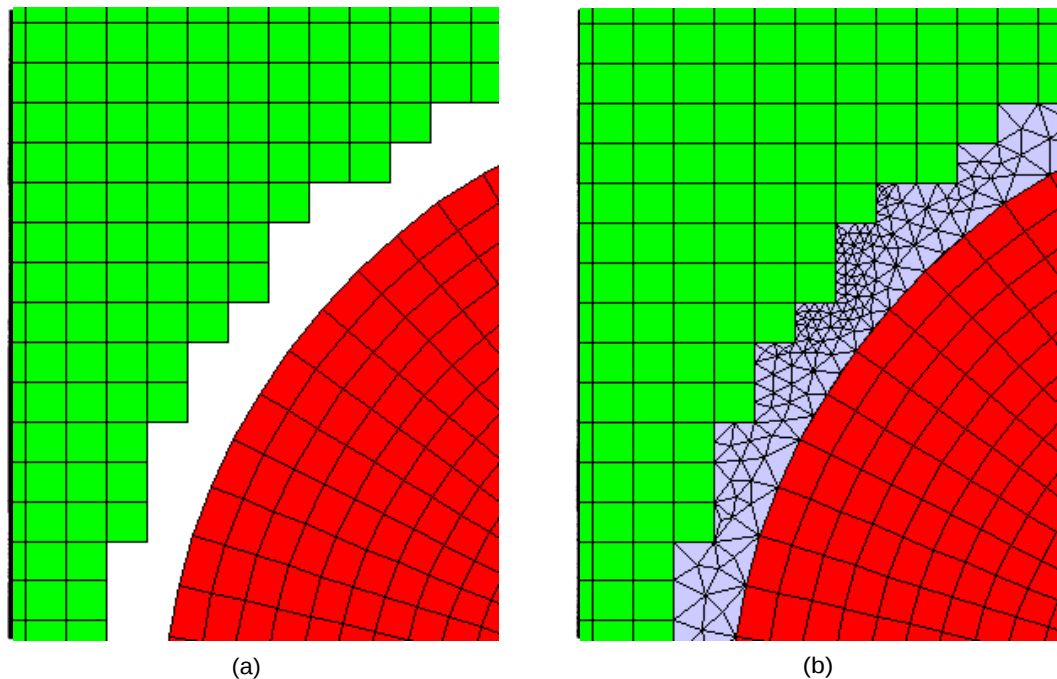


Figure 6: A sideview of a three-dimensional DRAGON grid.

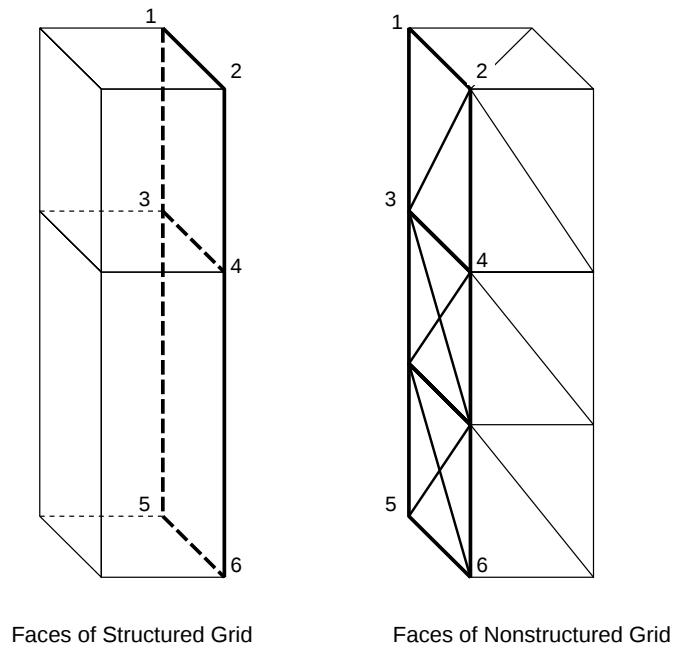


Figure 7: Interface between structured and nonstructured grids in a three-dimensional DRAGON grid.

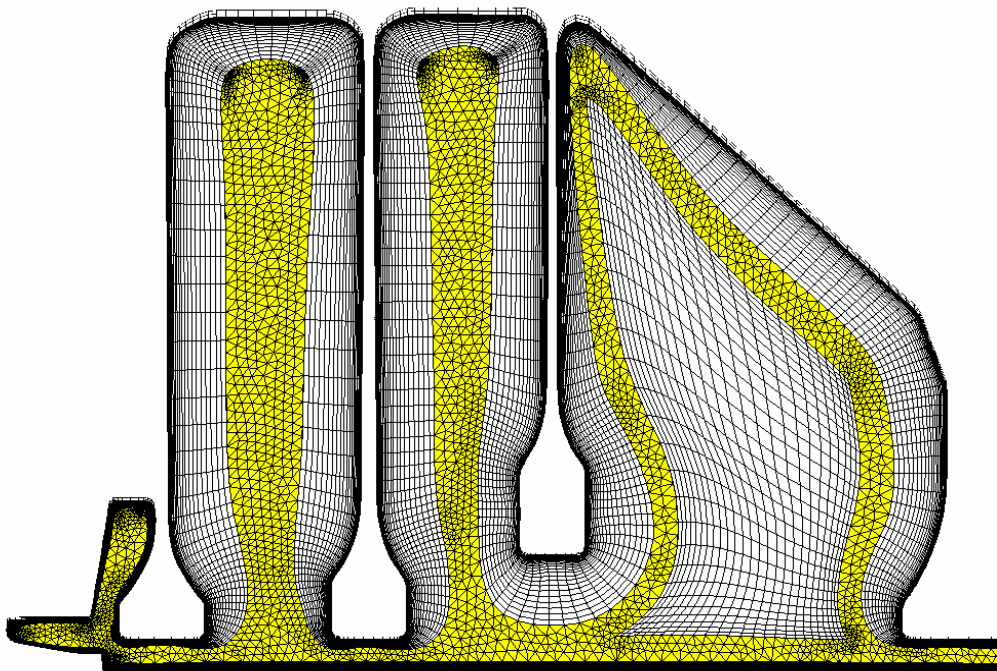


Figure 8: DRAGON grid for the compressor drum cavity problem (The outer rim barely reveals the third dimension).

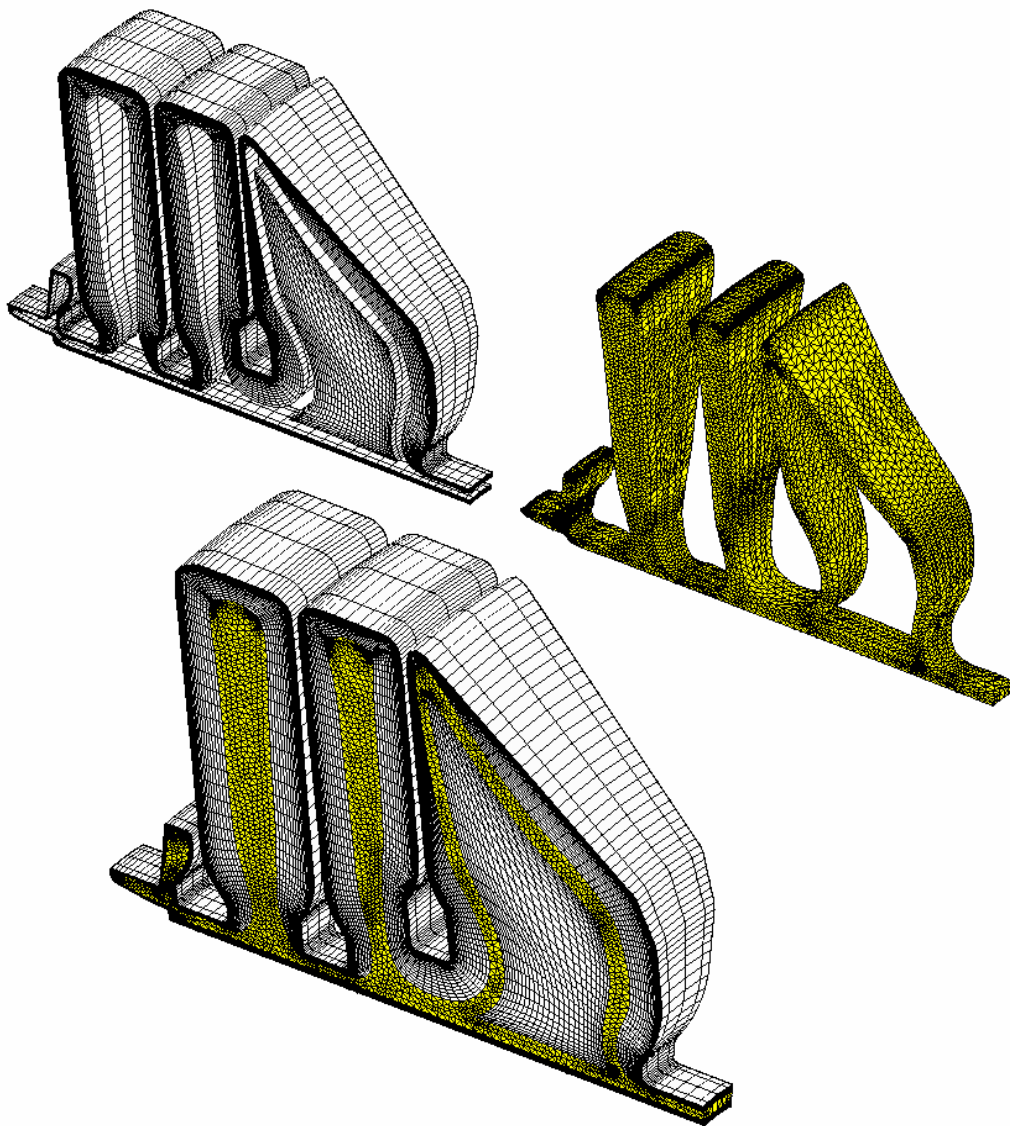


Figure 9: Split view of the DRAGON grid for the compressor drum cavity problem.

4.1 Programming Aspects

Apart from the PEGSUS code[2, 34], there are two main modules involved in the three-dimensional DRAGON grid generation, namely DRAGONFACE and MGEN3D. Figure 10 shows the flow chart in the three-dimensional DRAGON grid generation. DRAGONFACE is a FORTRAN code, which reads in composite structured grids with IBLANK values assigned by the PEGSUS code; and writes out topology information of the faces as a surface description for the unstructured grid to be created. It also provides a new set of IBLANK values, as some IBLANK values have been updated to ensure that a closed surface is available. Primary steps involved in the DRAGONFACE procedure are: identification of new faces, creation of isolated edges of the surface, and filling openings of the surface to ensure the closedness of the surface.

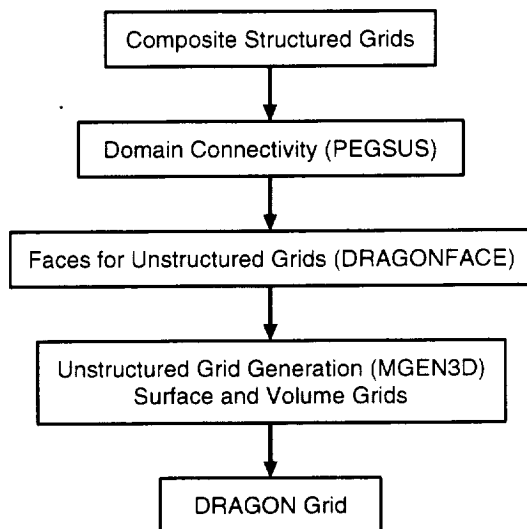


Figure 10: Program modules involved in the three-dimensional DRAGON grid generation.

MGEN3D is a C code, which generates volumetric unstructured grids, and topology information of the corresponding surface grids, which is essential to the data communication near the interface between structured and unstructured grids. The input data is the surface description from the DRAGONFACE procedure. Main steps involved in the MGEN3D procedure are: spacing source creation, three-dimensional surface triangulation, face reorientation, domain classification, and generation of volumetric grids.

4.2 Surface Generation for Nonstructured Grids

4.2.1 Basics of Resurfacing Module

In the DRAGON gridding procedure, a void domain between structured grids is to be filled with an unstructured grid. Meanwhile a domain occupied by overlapped structured

grids is also to be replaced with an unstructured grid. Before the filling and replacement, quadrilateral (and then triangular) surface elements are constructed to describe the boundary of the volume, where a great deal of element/element classification for the volumetric structured grids is involved. DRAGONFACE, a resurfacing module, has been developed for this purpose, in conjunction with employing the existing PEGSUS code, which cuts the overlapped regions and provides the associated blanking information.

A surface element consists of several edges. An edge is referred to as an isolated edge, if it belongs to only one element of the surface. For a general surface, there are many edges, which belong to two neighboring elements, and clearly these edges are not isolated edges. Furthermore, for a closed surface, there are no isolated edges; for an open surface, the corresponding isolated edges are bounding the openings of the surface. Therefore, in order to close the surface, it is natural to establish the isolated edges first, and then to identify facets which fill the openings.

In the following discussion, a special three-dimensional structured grid, namely 2.5-dimensional grid, is to be mentioned. A 2.5-dimensional grid is a two-dimensional grid, except for being topologically transformed (such as swept, rotated, and swept/rotated). Although the corresponding geometry is in three dimensions, it can be obtained by means of extrusion from a two-dimensional geometry, and the grid can be built based on a two-dimensional base grid *via* extrusion.

Referring to Figure 11, a flow chart of the DRAGONFACE module, we list the primary steps as follows.

1. Read in composite structured grids and the associated IBLANK values assigned by the PEGSUS code.
2. Identify new faces based on connectivity information of the volumetric grid elements, taking into account the known IBLANK values.
3. Assign coordinates and set flags for the points associated with the newly defined faces. Update IBLANK values if necessary to ensure that a closed surface is available.
4. Calculate default tolerance for determining coincident points.
5. Flag edge entries based on the surface information. With the help of a sorting scheme, establish isolated edges.
6. If the grid is 2.5-dimensional in the sense of topology, and the L direction is assumed to be the one which the third dimension lies in without loss of generality, then perform Operations `addwal1` and `addwal2`. Otherwise perform Operation `addwal3` for general cases. In summary, we can list:
`addwal1` – Fill gaps in the J and K directions;
`addwal2` – Identify openings in the L planes; and
`addwal3` – Identify openings for general cases.

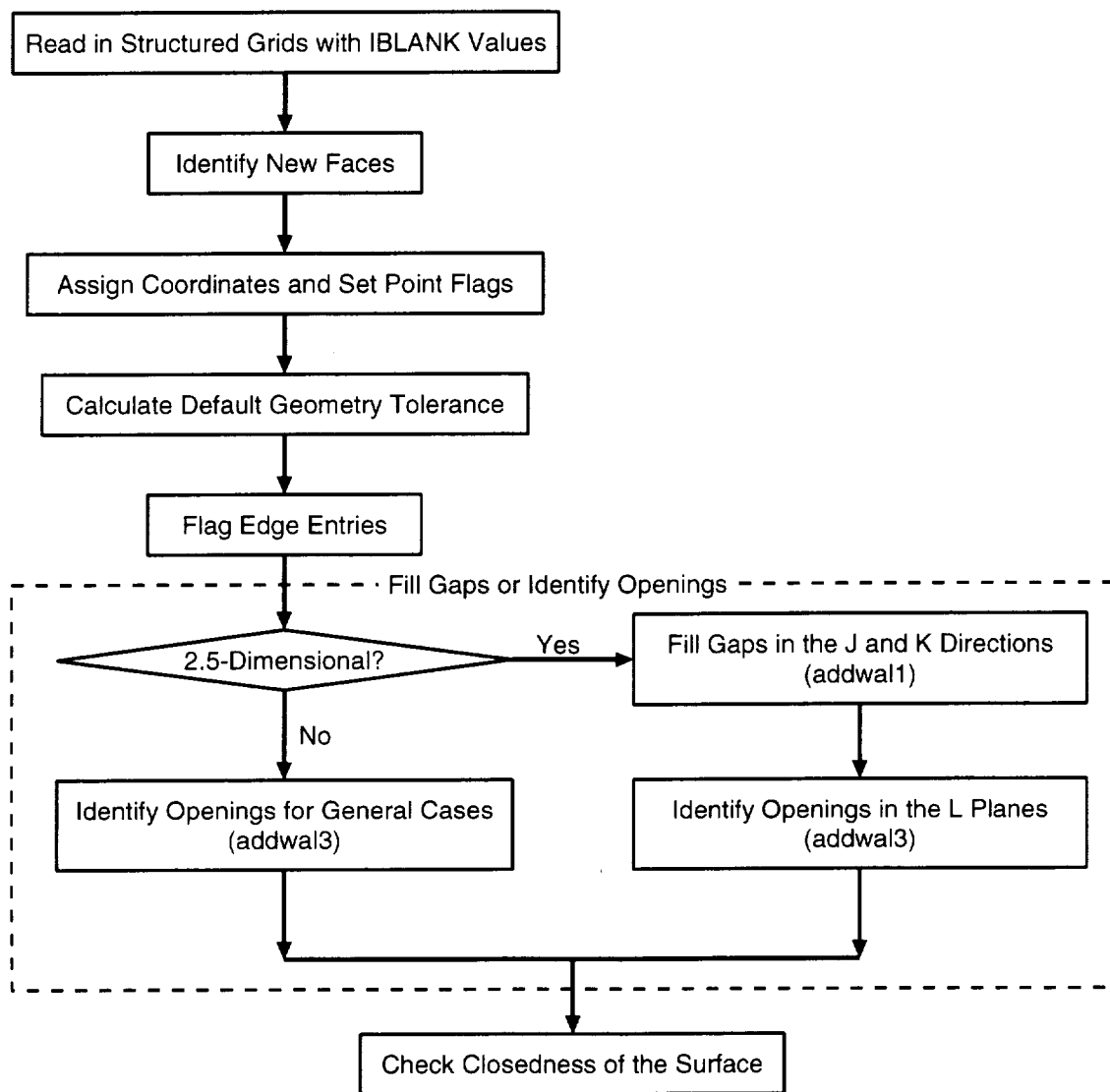


Figure 11: Flow chart of the DRAGONFACE procedure.

4.2.2 Defining Closed Surfaces

Here we will dissect schemes of filling gaps and identifying openings in the DRAGON-FACE procedure (Refer to Figure 11), which are the master part of the procedure. Operation `addwall` is designed to fill gaps in the J and K directions for 2.5-dimensional cases, and its detailed steps are itemized as follows.

1. Establish segments along the isolated edges of the faces defined so far.
2. Identify turning points on the segments. A point is referred to as a turning point if there exist two segments through the point, where one segment is on a J - K plane, but the other is not. Refer to Figure 12. Points A , B , C , and D are turning points.
3. If Points A and B , and Points C and D are pairs of coincident points, then zip the seam denoted by segments AC and BD .
4. If Points A and B , and Points C and D are not pairs of coincident points, then fill the gap $ABDC$ with quadrilaterals, referring to Figure 12.

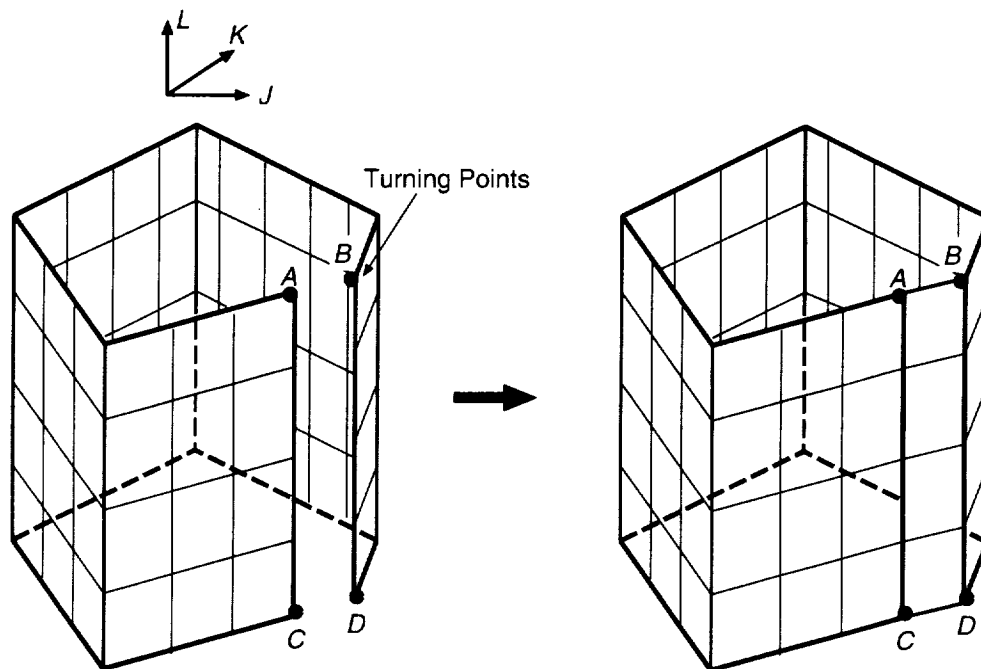


Figure 12: Illustration of filling a gap in the J and K directions and definition of turning points.

Operation `addwall2` is designed to identify openings in the L planes for the 2.5-dimensional case and the steps involved are given as follows.

1. Establish segments along the isolated edges of the faces defined so far.

2. Identify turning points on the segments if they exist.
3. Based on the segments, identify openings in the L planes. This will result in a set of closed loops of segments (Refer to Figure 13).
4. Determine an approximate fitting plane for the loops to be projected to.
5. According to the projection, determine location relationship of the loops (insidedness) by means of area method, rebuild loops if necessary and then create the surface description. For the example shown in Figure 13, the corresponding surface region is a region (*ie*, an opening to be filled) with two isolated islands inside it, where Loops L_2 and L_3 are considered to be inside Loop L_1 . These loops together define a surface region, a shaded area in Figure 13.

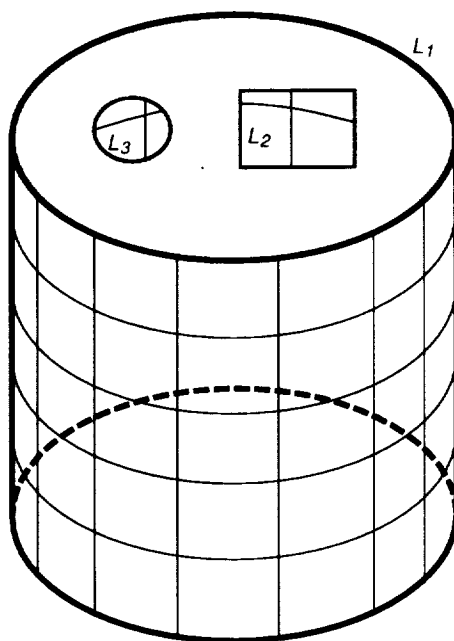


Figure 13: Illustration of identifying an opening in the L planes.

Figure 14 shows a surface definition for unstructured grid generation for a turbine branch-duct problem. This definition is the result from the DRAGONFACE procedure before the `addwal2` operation has been performed. The corresponding faces have not yet been split into triangles, and are still in the quadrilateral form. Figure 15 depicts the unstructured grid generated to fill the gap regions between structured grids.

For general cases, Operation `addwal3` has been designed to identify openings towards establishing the surface definition. The corresponding steps involved are as below.

1. Establish segments along the isolated edges of the faces defined so far.
2. Construct a set of segment loops from the segments created above.

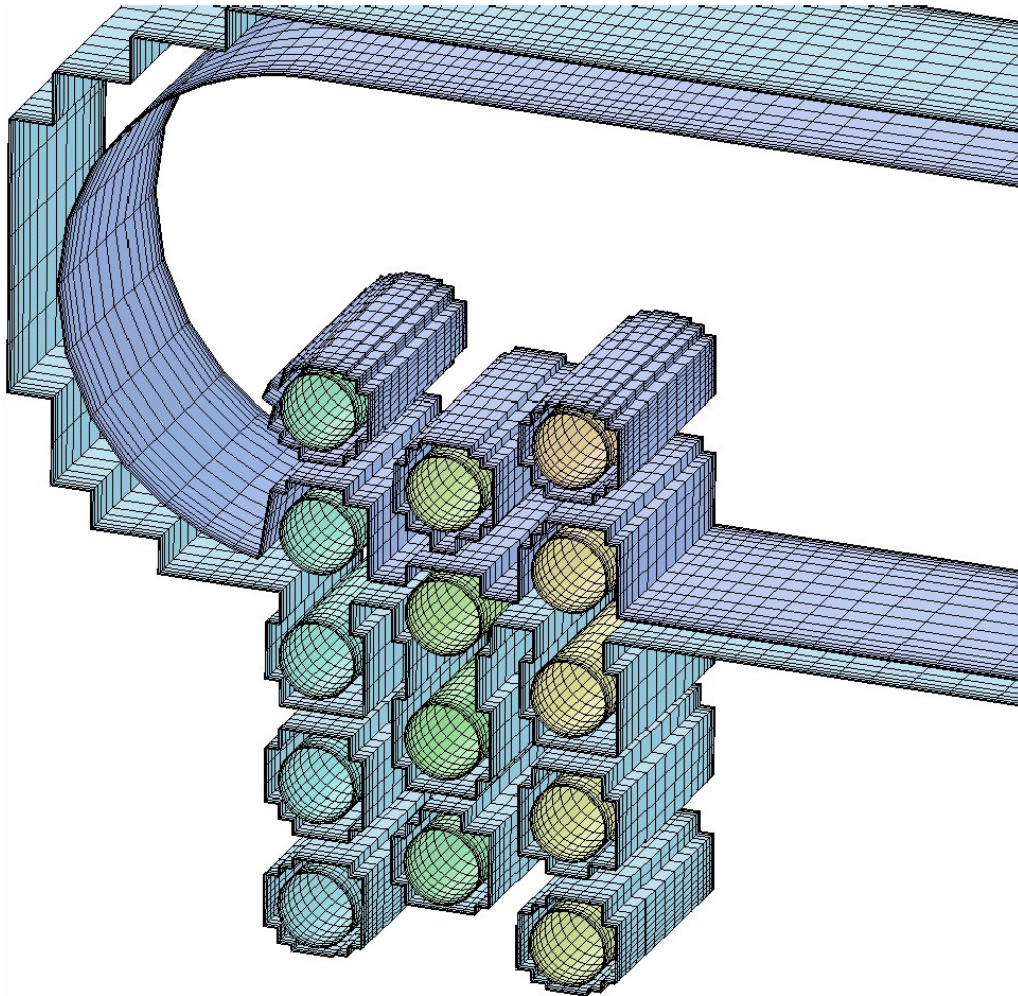


Figure 14: Surface description for unstructured grid generation for the turbine branch-duct problem.

3. Determine an approximate fitting plane for the loops to be projected to.
4. According to the projection, determine location relationship of the loops (insidedness) by means of area method, rebuild loops if necessary, and then create the surface description.

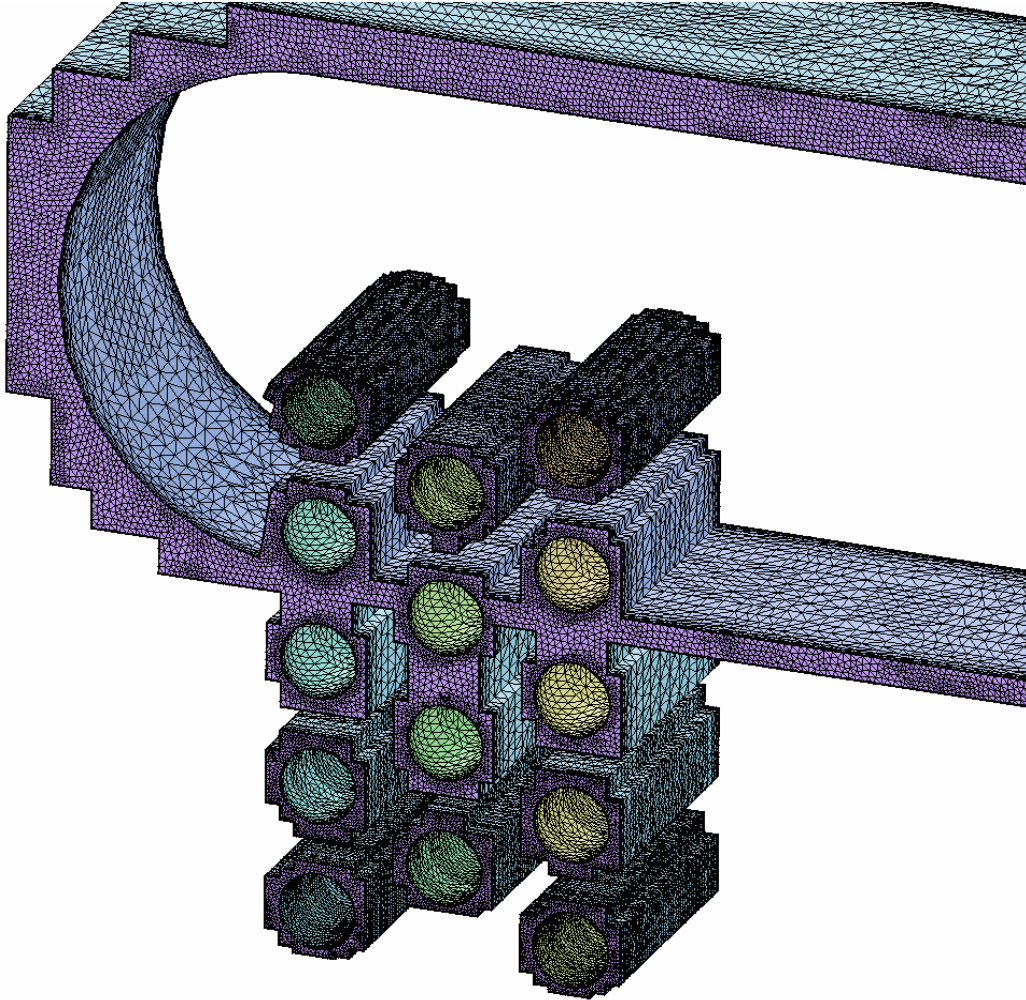


Figure 15: Unstructured grid domains of the DRAGON grid for the turbine branch-duct problem.

4.3 Generation of Nonstructured Grids

4.3.1 Basics of the Generation Method

The generation of nonstructured grids is essential to the three-dimensional DRAGON gridding. The specific task is the implementation of a tetrahedral grid generation kernel. This kernel is the key to the success of DRAGON gridding, and serves as an engine to create tetrahedral grids for particular domains. Currently there are two basic types of tetrahedral grid generation schemes: advancing front approach [27] and Delaunay triangulation [19]. Also a coupled approach [11] has been proposed to take the advantages of both schemes. After an investigation into these schemes, Delaunay triangulation has been chosen, partly taking into account the time frame required in the development.

The algorithm of the MGEN3D code evolves from the framework of previous work [19, 39, 40], and is a Delaunay-type method. The Steiner point[26] creation algorithm has been incorporated into this grid generator. Surface grids are created based on two-dimensional geometries by means of coordinate mapping. Volume grids are generated through three-dimensional triangulation and interior point creation based on the surface grids. Boundary surface conformity is gained *via* edge swapping, boundary edge and surface recovery. In order to control the quality of the resulting grids, several parameters have been introduced, such as the baseline grid spacing and the grid regularity parameter.

Algorithm involved in the MGEN3D procedure are summarized as follows.

1. Surface description generated *via* the DRAGONFACE procedure is read in, while spacing source creation is performed automatically (Refer to Section 4.3.2 below).
2. Surface triangulation is carried out taking into account the baseline grid spacing, point and line source distribution.
3. From this surface triangulation, the face orientation may not be consistent over the whole surface. Face reorientation is necessary and essential, which ensures that all the normals of the triangular faces are consistently outwards or inwards.
4. Domain classification is performed to provide domain information of the geometry specified by the above surface definition.
5. Volumetric grids are generated for the domains one by one until the end. Data of the assembly of individual volumetric grids, including nodal coordinates and connectivities, are finally created.

4.3.2 Automatic Spacing Source Creation

A conversion from the quadrilateral grid-based surface into the triangular grid-based surface is necessary, in order to generate tetrahedral grids. From time to time, there might be

sharp changes in the sizes of the quadrilaterals. In order to obtain a surface grid, and then a volume grid, with smooth variation of point spacing, certain controls need to be introduced. Sometimes certain quadrilaterals are stretched too much, then another type of control may be employed, in order to get a grid with limited stretch ratio of its elements.

A point source is assigned with two radii and a spacing value. Within the first radius the elements will be generated which have a length scale that is similar to the spacing value specified. The element length scale then exponentially increases to a global spacing value outside the first radius. The incremental rate of expansion of point spacing is defined by the relationship shown in Figure 16. This expression prescribes the spacing at the outer radius to be twice that at the inner radius. Figure 16 illustrates the influence of a point source on the point spacing variation. In light of the concept of point source, the concept of line source can be introduced. A line source has a cylindrical range of influence. The point and line source distribution control surface grid generation, and further volumetric grid generation as well.

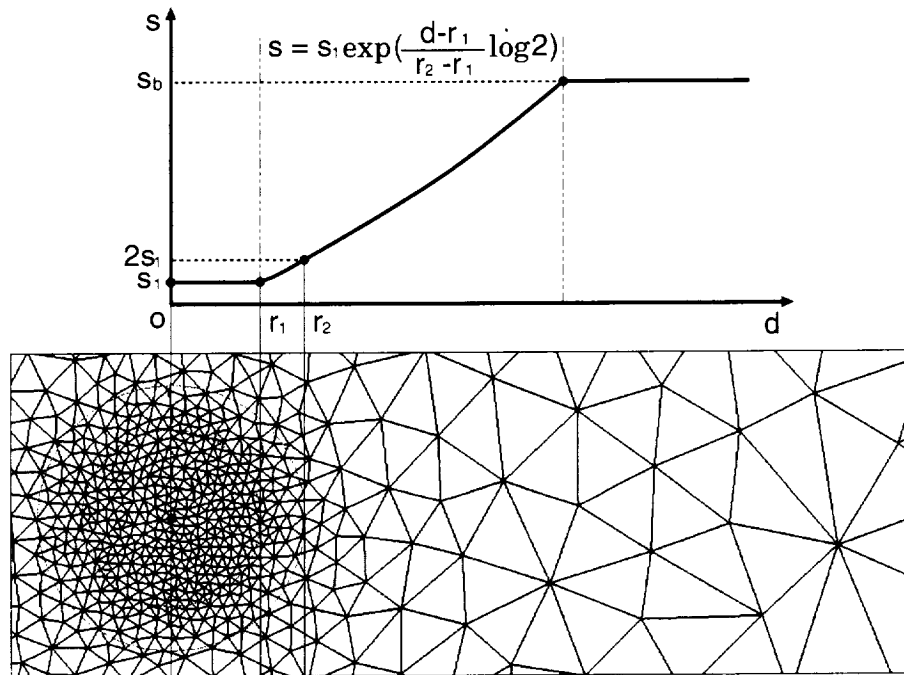


Figure 16: Illustration of the influence of a point grid source on the point spacing variation.

The procedure to create point and line sources to control the spacing of the new surface grids has been automated during the DRAGONFACE process. Figure 17 depicts point sources, which reflect the spacing variation of the structured grid, and line sources, which are introduced in the neighborhoods of stretched elements. A coloring scheme is used to display the intensity variations of various sources.

A baseline grid spacing can take a value, which is of the same order as the average grid size of the quadrilaterals. Then roughly a quadrilateral, with an edge length larger than the baseline grid spacing, will be divided into more than two triangles. This baseline grid

spacing value is only a reference value for the grid generation. And the point and line sources provide a further and local control on the overall grid spacing variation.

If the maximal edge length of a quadrilateral is larger than a given grid spacing threshold, then a point source will be created to influence the neighborhood of the quadrilateral. If the stretch ratio of a quadrilateral is larger than a given stretch threshold, then a line source will be generated with an influence on the neighborhood of the quadrilateral.

4.4 Industrial Examples

Three examples often seen in propulsion systems have been introduced to validate the three-dimensional DRAGON gridding methodology. These include a compressor drum cavity, turbine branch-duct – a portion of an internal coolant passage, and a film-cooled turbine vane. We have seen in Figures 8 and 9 the three-dimensional DRAGON grid for the compressor drum cavity.

Figures 18 and 19 are for the turbine branch-duct. They show rows of pins placed in a turbine coolant passage used to promote mixing. Once a cylindrical grid is generated for a pin, a duplication of it can be added and moved to any place to assess their effects on aerodynamic performance and heat transfer. For the generation of the overlaid structured grids, it is just as easy to add/delete components to/from the configuration. This flexibility clearly has great potential for design purposes.

By examining Figure 19, we find that the unstructured grid is much finer than the structured grid nearby. The small grid spacing is due to the fact that the spacing of the structured grids vary in the spanwise direction, with the smallest spacing values at the top and bottom of the geometries in that direction to account for the viscous effects near the side walls (Refer to Figure 15). When an original quadrilateral element is greatly stretched, a line source will be automatically created within the framework of the standard Delaunay-type gridding algorithm, which usually tends to yield a finer grid locally. The standard Delaunay-type grid generation normally leads to isotropic grids. If an extension to the standard method were made, in order to create stretched grids, then the unstructured grid shown in Figure 19 could be coarser, and some memory would be saved.

Lastly, we consider a typical film-cooled turbine vane, whose schematic is shown in Figure 20. This geometry includes the vane, coolant plena, and 33 holes inside of the vane. Figure 21 depicts the whole resulted DRAGON grid, where the connecting regions between the 33 holes and the flow domain are filled with unstructured grids. Figure 22 is a close-up view of this DRAGON grid in the leading edge region of the film-cooled turbine vane. Furthermore, Figure 23 followed by Figure 24 provide a deeper insight into this DRAGON grid by means of cutting through the grid, where attention has only been paid to the leading edge region.

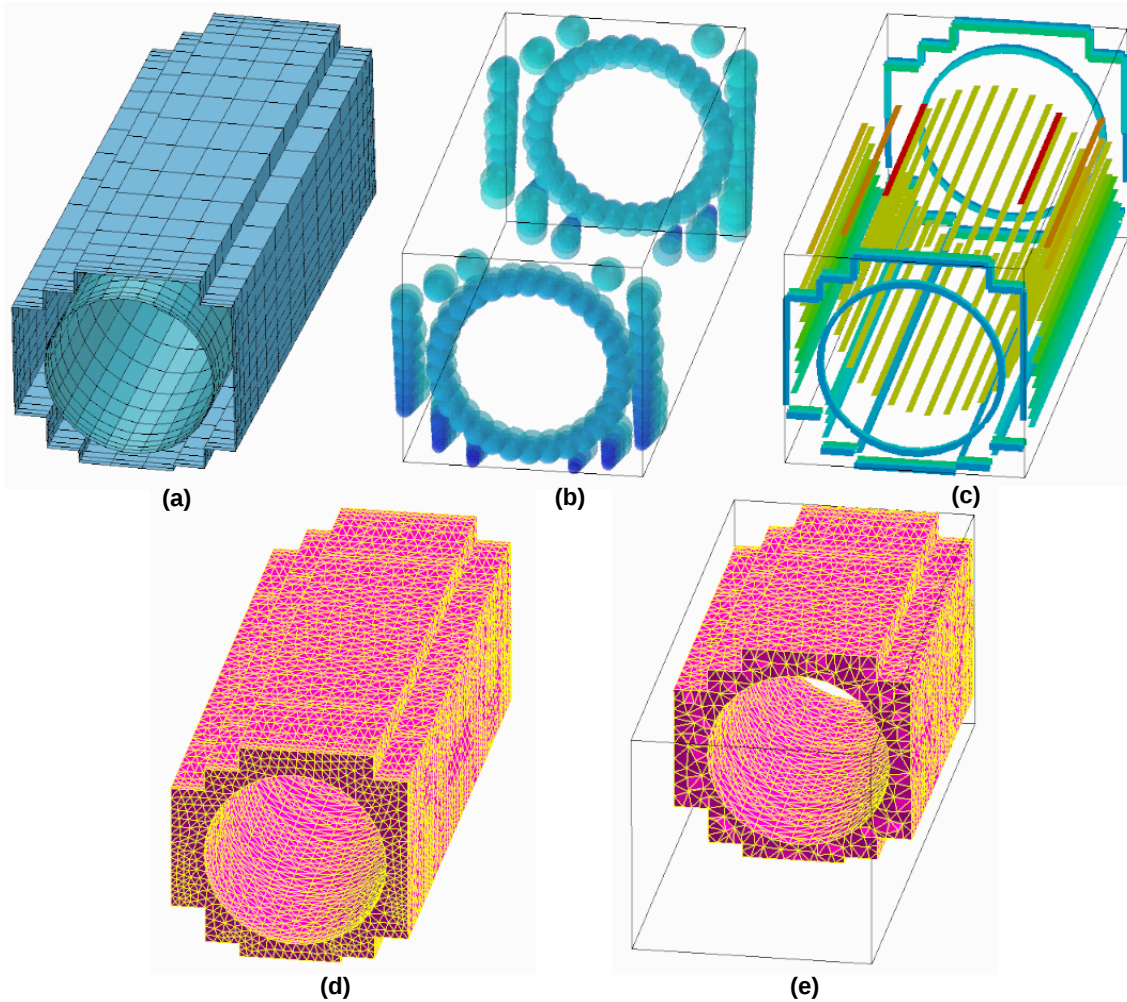


Figure 17: Illustration of automatic spacing source creation during unstructured grid generation: (a) Surface description; (b) Point sources created to reflect the spacing variation of the structured grid; (c) Line sources introduced in the area of stretched elements; (d) Volumetric grid; and (e) Volumetric grid with a part cut.

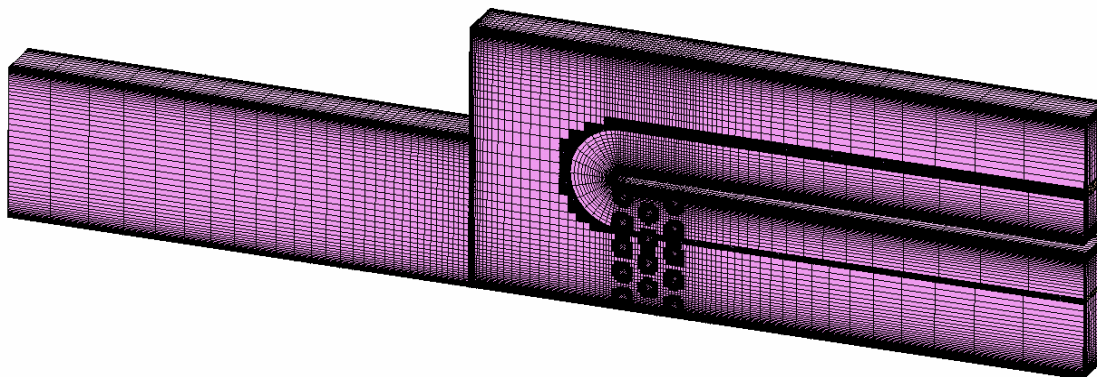


Figure 18: DRAGON grid for the turbine branch-duct problem.

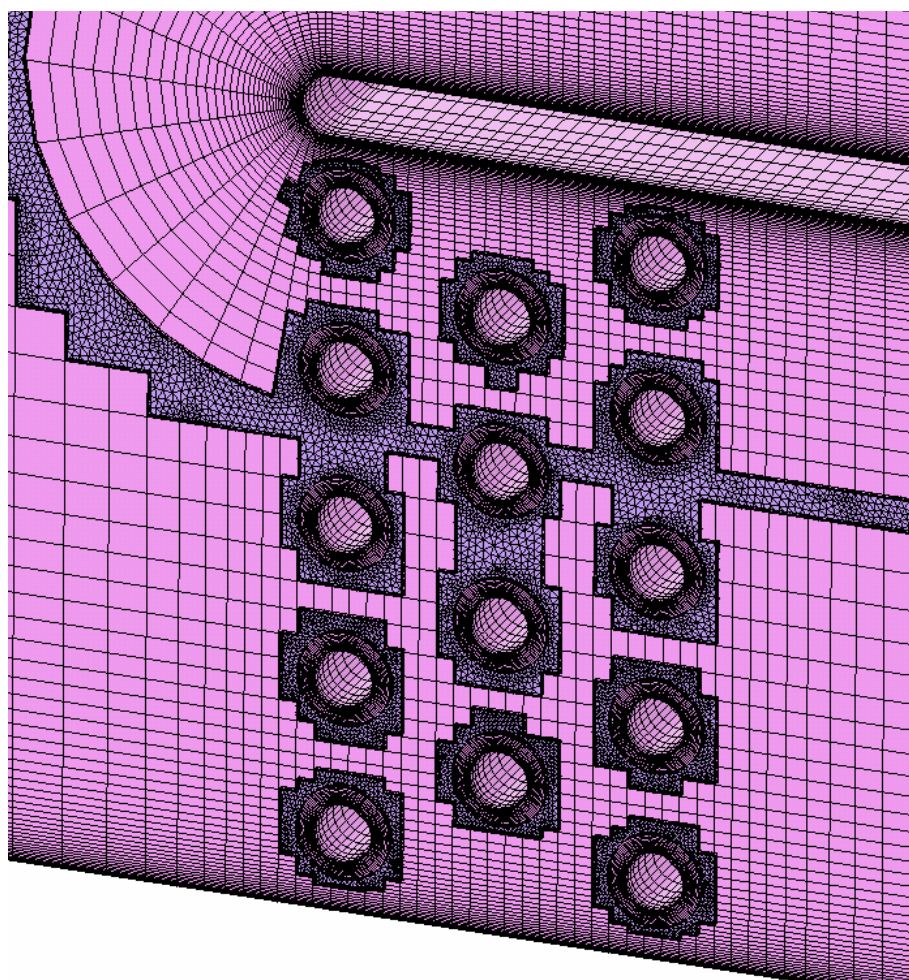


Figure 19: Close-up view of the DRAGON grid for the turbine branch-duct problem.

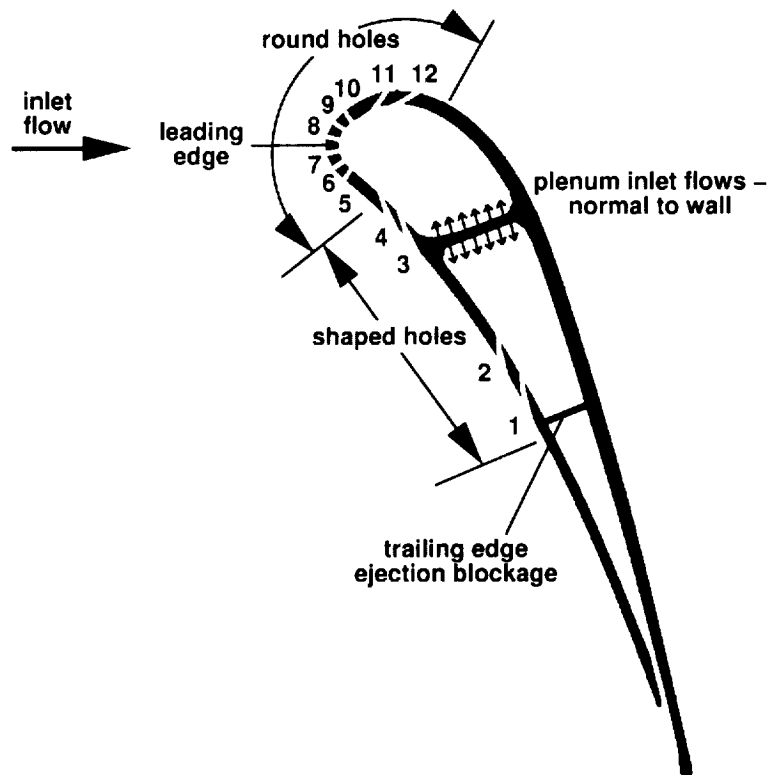


Figure 20: Schematic of the vane cross-section and coolant holes.

During the DRAGON grid generation, the 33 individual structured grids for the holes have been created without trying to topologically join them to the grids representing the external flow domain, which would be required in the multiblock grid generation [13]. From this point of view, the DRAGON grid scheme could be considered as a more flexible and easier approach.

However, the user may prefer to have structured grids presented somewhere, due to the physical and numerical characteristic. For instance in this example, the user might like to use structured grids to fill the connecting regions between the holes and the flow domain. Because the DRAGON grid is actually created by using a base composite structured grid, one can make structured grids presented in a particular region, as long as the base composite structured grid is generated with such an intention.

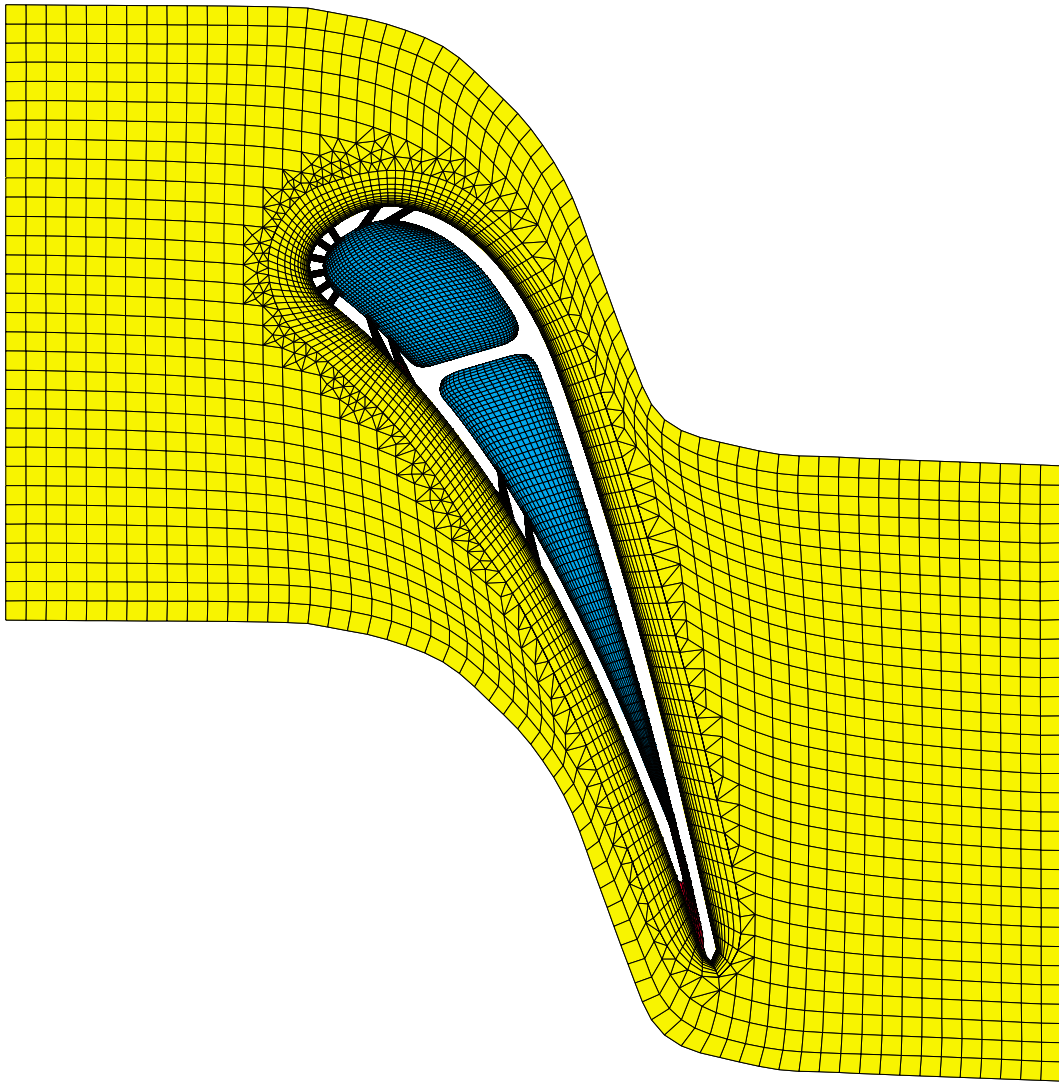


Figure 21: DRAGON grid for the film-cooled turbine vane.

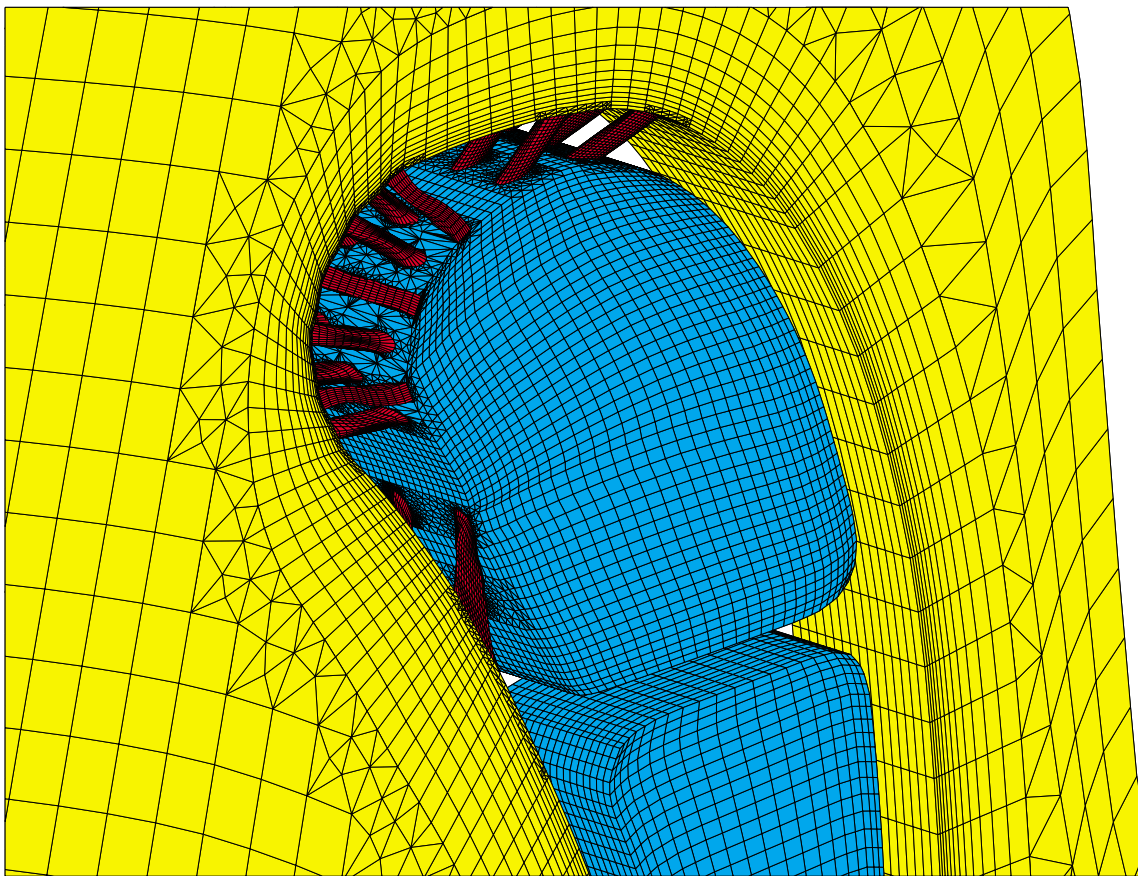


Figure 22: Close-up view of DRAGON grid in the leading edge region of the film-cooled turbine vane.

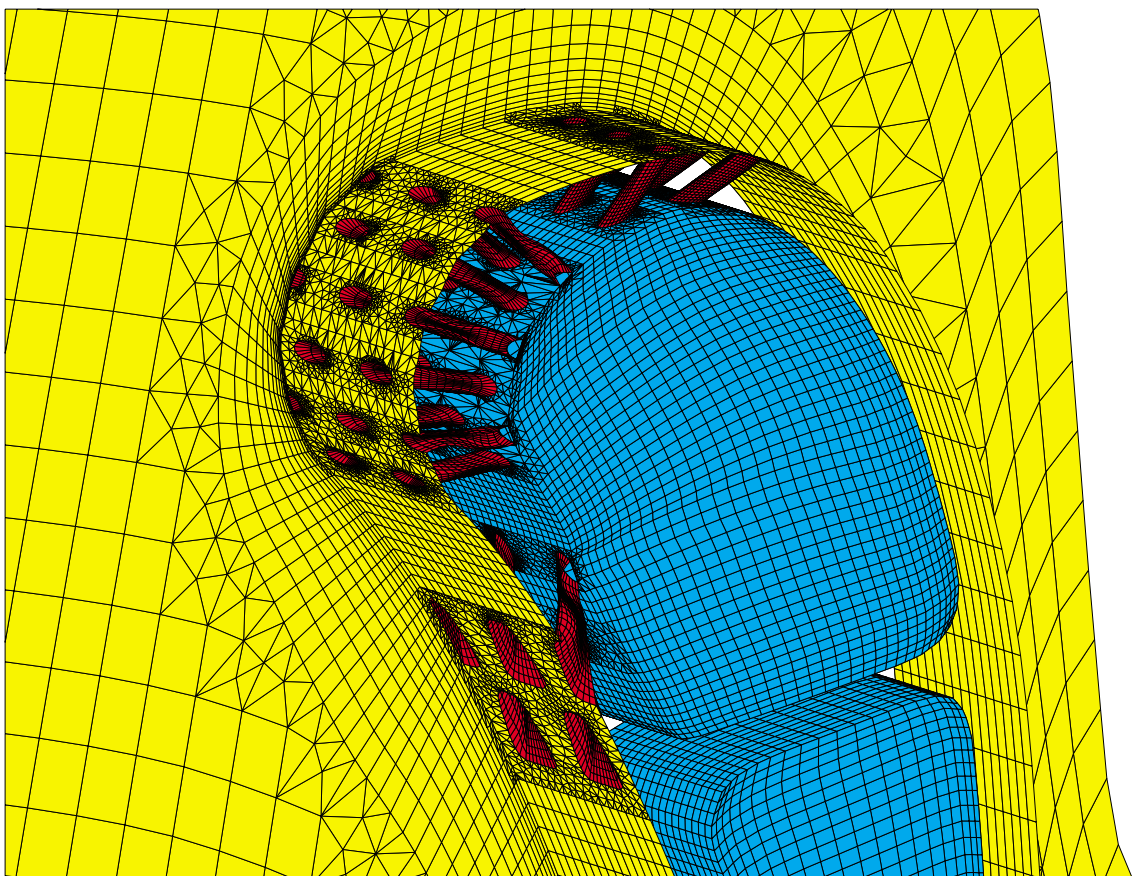


Figure 23: Cut-through view of the DRAGON grid in the leading edge region of the film-cooled turbine vane.

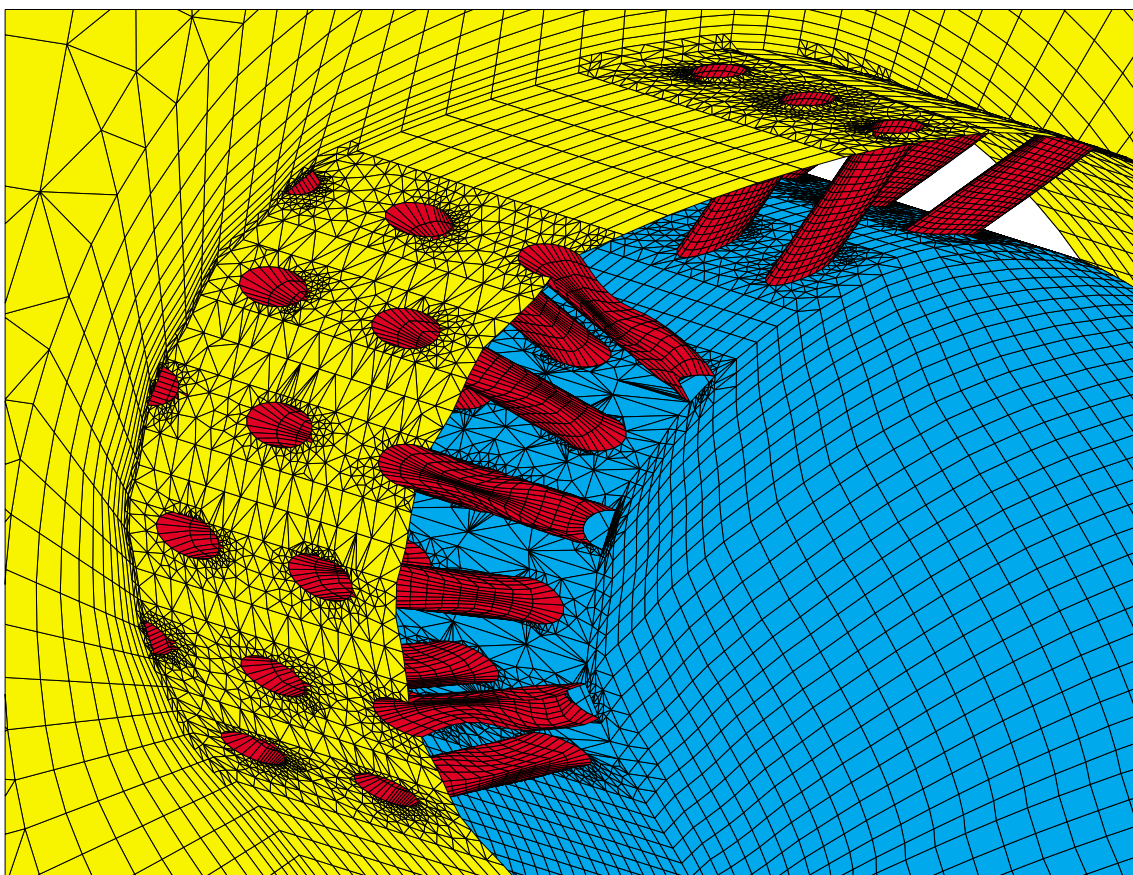


Figure 24: Close-up view of the DRAGON grid shown in Figure 23.

5 Conclusions

In the present work, we have presented the essentials of a new three-dimensional hybrid grid, termed DRAGON grid, methodology. This methodology attempts to blend the advantageous features of both the structured and unstructured grids, while eliminating or minimizing their respective shortcomings. As a result, the method is very amenable to **quickly** creating **quality** viscous grids for various individual components with **complex** shapes found in an engineering system. Computationally, the resulting grid drastically reduces memory requirements used in the unstructured grid and makes more efficient solvers accessible because the major portion of the DRAGON grid is in the form of the structured grid. Moreover, high quality viscous meshes are inherited in the process, unlike in the unstructured grid generation where structured-grid-like meshes (such as prismatic layers) need to be added, but still based on the unstructured-grid data structure.

The DRAGON grid uses the Chimera grid methodology [32] as a foundation to generate the bulk of grids and then applies the strength of unstructured grid methodology to make up the remaining grids. As a central point of this work, we have concentrated on the extension of the DRAGON grid technology into three-dimensional space. Various aspects of the extension, and new challenges for the three-dimensional cases, have been investigated. As a result of this study, we concluded that the applications of three-dimensional DRAGON grid technology are very encouraging because the objectives of quickly generating quality grids for complex geometries have been demonstrated.

Acknowledgments

We thank Dr. James D. Heidmann for kindly providing us the geometry of the film-cooled turbine vane. We also thank Dr. Kumud Ajmani for generating the Chimera grids for the compressor drum cavity and the turbine branch-duct.

References

- [1] E. H. Atta. Component-adaptive grid interfacing. AIAA Paper 81-0382, AIAA 19th Aerospace Sciences Meeting, St. Louis, MO, 1981.
- [2] J. A. Benek, J. L. Steger, F. C. Dougherty, and P. G. Buning. Chimera: A grid-embedding technique. Report AEDC-TR-85-64, Arnold Engineering Development Center, April 1986.
- [3] M. J. Berger. On conservation at grid interfaces. NASA ICASE Report 84-43, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, September 1984.

- [4] M. J. Berger and J. Oliger. Adaptive mesh refinements for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512, 1984.
- [5] K. R. Blake and G. S. Spragle. Unstructured 3D Delaunay mesh generation applied to planes, trains and automobiles. AIAA Paper 93-0673, AIAA 31th Aerospace Sciences Meeting & Exhibit, Reno, NV, January 1993.
- [6] A. Bowyer. Computing Dirichlet tessellations. *The Computing Journal*, 24(2):162–166, 1981.
- [7] D. L. Brown, G. Chesshire, and W. D. Henshaw. Getting started with CMPGRD, Introductory user's guide and reference manual. LA-UR-89-1294, Los Alamos National Laboratory, 1989.
- [8] P. G. Buning, I. T. Chiu, F. W. Martin, Jr., R. L. Meakin, S. Obayashi, Y. M. Rizk, J. L. Steger, and M. Yarrow. Flowfield simulation of the space shuttle vehicle in ascent. In *Proc. the Fourth International Conference on Supercomputing*, pages 20–28, Santa Clara, CA, April 1989.
- [9] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90:1–64, 1990.
- [10] F. C. Dougherty, J. A. Benek, and J. L. Steger. On application of Chimera grid schemes to store separation. NASA TM 88193, NASA, October 1985.
- [11] P. J. Frey, H. Borouchaki, and P.-L. George. 3D Delaunay mesh generation coupled with an advancing-front approach. *Computer Methods in Applied Mechanics and Engineering*, 157:115–131, 1998.
- [12] F. Ghaffari. On the vortical-flow prediction capability of an unstructured-grid Euler solver. AIAA Paper 94-0163, AIAA 32nd Aerospace Sciences Meeting & Exhibit, Reno, NV, January 1994.
- [13] J. D. Heidmann, D. Rigby, and A. A. Ameri. A three-dimensional coupled internal/external simulation of a film-cooled turbine vane. Paper 99-GT-186, International Gas Turbine & Aeroengine Congress & Exhibition, Indianapolis, IN, 1999.
- [14] D. G. Holmes and S. D. Connell. Solution of the 2D Navier-Stokes equations on unstructured adaptive grids. AIAA Paper 89-1932, AIAA 9th CFD Conference, Buffalo, NY, June 1989.
- [15] A. Jameson, T. J. Baker, and N. P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103, AIAA 24th Aerospace Sciences Meeting, Reno, NV, January 1986.
- [16] K.-H. Kao and M.-S. Liou. Advance in overset grid schemes: From Chimera to DRAGON grids. *AIAA Journal*, 33(10):1809–1815, 1995.
- [17] K.-H. Kao, M.-S. Liou, and C. Y. Chow. Grid adaptation using Chimera composite overlapping meshes. *AIAA Journal*, 32(5):942–949, 1994.

- [18] G. H. Klopfer and G. A. Molvik. Conservative multi-zonal interface algorithm for the 3D Navier-Stokes equations. AIAA Paper 91-1601-CP, AIAA 10th CFD Conference, Honolulu, Hawaii, June 1991.
- [19] R. W. Lewis, Y. Zheng, and D. T. Gethin. Three-dimensional unstructured mesh generation: Part 3. Volume meshes. *Computer Methods in Applied Mechanics and Engineering*, 134(3/4):285–310, 1996.
- [20] M.-S. Liou and K.-H. Kao. Progress in grid generation: From Chimera to DRAGON grids. NASA TM 106709, Lewis Research Center, Cleveland, Ohio, August 1994. Also Chapter 21, in *Frontiers of Computational Fluid Dynamics 1994*, ed. D. A. Caughey and M. M. Hafez, John Wiley & Sons, November 1994.
- [21] R. Löhner. An adaptive finite element scheme for transient problems in CFD. *Computer Methods in Applied Mechanics and Engineering*, 61(3):323–338, 1987.
- [22] R. L. Meakin. On adaptive refinement and overset structured grids. AIAA Paper 97-1858. AIAA 13th CFD Conference, Snowmass, CO, June 1997.
- [23] R. L. Meakin and N. Suhs. Unsteady aerodynamic simulation of multiple bodies in relative motion. AIAA Paper 89-1996-CP, AIAA 9th CFD Conference, Buffalo, NY, June 1989.
- [24] Y. J. Moon and M.-S. Liou. Conservative treatment of boundary interfaces for overlaid grids and multi-level grid adaptations. AIAA Paper 89-1980-CP, AIAA 9th CFD Conference, Buffalo, NY, June 1989.
- [25] K. Nakahashi and S. Obayashi. FDM-FEM zonal approach for viscous flow computations over multiple-bodies. AIAA Paper 87-0604, AIAA 25th Aerospace Sciences Meeting, Reno, NV, January 1987.
- [26] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Chichester, 1992.
- [27] J. Peraire, J. Peiro, and K. Morgan. Adaptive remeshing for three-dimensional compressible flow computations. *Journal of Computational Physics*, 103:269–285, 1992.
- [28] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72:449–466, 1987.
- [29] M. Posenau. *Unstructured Grid Generation Techniques and Software, NASA CP-10119*. NASA, 1993.
- [30] M. M. Rai. A conservative treatment of zonal boundaries for Euler equation calculations. *Journal of Computational Physics*, 62:472–503, 1986.
- [31] M. Soetrismo, S. T. Imlay, and D. W. Roberts. A zonal implicit procedure for hybrid structured-unstructured grids. AIAA Paper 94-0645, AIAA 32nd Aerospace Sciences Meeting & Exhibit, Reno, NV, 1994.

- [32] J. L. Steger. Thoughts on the Chimera method of simulation of three-dimensional viscous flow. In M.-S. Liou and L. A. Povinelli, editors, *Proc. Computational Fluid dynamics Symposium on Aeropropulsion*, NASA CP-3078, pages 1–10, 1990.
- [33] J. L. Steger and J. A. Benek. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 64(1/3):301–320, 1987.
- [34] N. E. Suhs and R. W. Tramel. *PEGSUS 4.0 User's Manual*. AEDC-TR-91-8. Calspan Corporation/AEDC Operations, Arnold AFB, TN, November 1991.
- [35] R. Taghavi. Automatic, parallel and fault tolerant mesh generation from CAD on Cray research supercomputers. Technical report, Cray User Group Conference, Tours, France, 1994.
- [36] J. L. Thomas, R. W. Walters, T. Reu, F. Ghaffari, R. Weston, and J. M. Luckring. Patched-grid algorithm for complex configurations directed towards the F/A-18 aircraft. AIAA Paper 89-0121, AIAA 27th Aerospace Sciences Meeting, Reno, NV, January 1989.
- [37] Z. J. Wang and H. Q. Yang. A unified conservative zonal interface treatment for arbitrarily patched and overlapped grids. AIAA Paper 94-0320, AIAA 32nd Aerospace Sciences Meeting & Exhibit, Reno, NV, January 1994.
- [38] N. P. Weatherill. On the combination of structured-unstructured meshes. In S. Sengupta, J. Häuser, P. R. Eiseman, and J. F. Thompson, editors, *Numerical Grid Generation in Computational Fluid Mechanics*, pages 729–739, Swansea, UK, 1988. Pineridge Press.
- [39] Y. Zheng, R. W. Lewis, and D. T. Gethin. Three-dimensional unstructured mesh generation: Part 1. Fundamental aspects of triangulation and point creation. *Computer Methods in Applied Mechanics and Engineering*, 134(3/4):249–268, 1996.
- [40] Y. Zheng, R. W. Lewis, and D. T. Gethin. Three-dimensional unstructured mesh generation: Part 2. Surface meshes. *Computer Methods in Applied Mechanics and Engineering*, 134(3/4):269–284, 1996.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 1999		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE Development of Three-Dimensional DRAGON Grid Technology			5. FUNDING NUMBERS WU-523-26-13-00	
6. AUTHOR(S) Yao Zheng, Meng-Sing Liou, and Kestutis C. Civinskas				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-11957	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-1999-209458	
11. SUPPLEMENTARY NOTES Yao Zheng, Taitech, Inc., Cleveland, Ohio, 44135; Meng-Sing Liou and Kestutis C. Civinskas, NASA Glenn Research Center. Responsible person, Meng-Sing Liou, organization code 5880, (216) 433-5855.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Categories: 02, 07, 34, and 61 This publication is available from the NASA Center for AeroSpace Information. (301) 621-0390.			12b. DISTRIBUTION CODE Distribution: Standard	
13. ABSTRACT (Maximum 200 words) For a typical three dimensional flow in a practical engineering device, the time spent in grid generation can take 70 percent of the total analysis effort, resulting in a serious bottleneck in the design/analysis cycle. The present research attempts to develop a procedure that can considerably reduce the grid generation effort. The DRAGON grid, as a hybrid grid, is created by means of a Direct Replacement of Arbitrary Grid Overlapping by Nonstructured grid. The DRAGON grid scheme is an adaptation to the Chimera thinking. The Chimera grid is a composite structured grid, composing a set of overlapped structured grids, which are independently generated and body-fitted. The grid is of high quality and amenable for efficient solution schemes. However, the interpolation used in the overlapped region between grids introduces error, especially when a sharp-gradient region is encountered. The DRAGON grid scheme is capable of completely eliminating the interpolation and preserving the conservation property. It maximizes the advantages of the Chimera scheme and adapts the strengths of the unstructured grid while at the same time keeping its weaknesses minimal. In the present paper, we describe the progress towards extending the DRAGON grid technology into three dimensions. Essential and programming aspects of the extension, and new challenges for the three-dimensional cases, are addressed.				
14. SUBJECT TERMS Computational fluid dynamics; Grid generation			15. NUMBER OF PAGES 41	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

