Simulation of a Single-Element Lean-Direct Injection Combustor Using Arbitrary Polyhedral Mesh

Thomas Wey[1] and Nan-Suey Liu[2]

NASA Glenn Research Center

Cleveland, Ohio 44135 USA

Abstract

This paper summarizes procedures of generating the arbitrary polyhedral mesh as well as presents sample results from its application to the numerical solution of a single-element LDI combustor using a preliminary version of the new OpenNCC.

Introduction

The lean direct injection (LDI) concept has the potential for low emissions under operational (high temperature, high pressure) conditions. In this concept, the liquid fuel is injected from a venture directly into the incoming swirling airflow, and the swirling air stream is used to atomize the injected liquid as well as to promote fuel-air mixing. The flame structure can be very complex and locally range from non-premixed to premixed burning.

Recently, a single-element LDI combustor experiment has been used as a test bed for assessing, further developing and validating the capability of two-phase turbulent combustion modeling and simulation. A series of numerical calculations have been performed by using (1) the time filtered Navier-Stokes (TFNS) methodology and (2) the large eddy simulation (LES) methodology. The sub-grid models employed for turbulent mixing and combustion include the well-mixed model, the linear eddy mixing (LEM) model; the Eulerian filtered mass density function (EUFDF/EUPDF) model, and the flamelet-based model. Results from these methodologies invoking various sub-grid models are summarized in Reference [1], and a more detailed description of the TFNS approach can be found in Reference [2]. It should be pointed out that TFNS is not LES, nor hybrid RANS/LES, nor, in general, unsteady Reynolds-averaged Navier-Stokes (URANS). Like the LES, TFNS is capable of capturing the dynamically important, unsteady turbulent flow structures, even when RANS-grade meshes are used. Unlike the

[1] Research Aerospace Engineer, Combustion Branch, Member AIAA.
[2] Research Aerospace Engineer, Combustion Branch, Associate Fellow AIAA.

LES, the grid resolution and the turbulence model fidelity are not formally linked, therefore, in principle; a grid independent solution can be unambiguously attained in the TFNS approach.

In the present work, a mesh-based enhancement for flux capturing is explored. The idea is that, in addition to numerical scheme and physical model, the shape of the mesh also plays an important role in the accuracy of the calculated solution. It is envisioned that, for the unsteady, swirling and re-circulating flow typically occurring in the combustion chamber, the polyhedral mesh, due to its increased number of the flux-exchange sites between the solution elements, can better capture the temporally and spatially varying fluxes, leading to better resolved large scale mixing. In this report, the practical aspects of the generation of the arbitrary polyhedral mesh are first described, followed by the sample results from its application to a single-element LDI combustor using a preliminary version of the new OpenNCC, which is intended as the self-contained, releasable edition of the National Combustion Code.

Generation of Polyhedral Mesh via Refining Simple Regular Mesh

Arbitrary polyhedrons permit cells of arbitrary shape to be used. Cells can have an arbitrary number of faces. Faces can have an arbitrary number of points. All commonly used meshes, such as tetrahedron/triangle, quadrilateral/hexahedron, prism and pyramid, can be expressed in terms of polyhedral data structure. There are several ways to generate a polyhedral mesh. In the present work, the polyhedral mesh is generated from the regular unstructured mesh through the processes of refinement, reconnection and agglomeration. Since the refinement process is involved, the number of cells is increased instead of decreased. A discussion of this approach to derive arbitrary polyhedron/polygon from a tetrahedral/triangular mesh can be found in Reference [3]. Here, the more general procedures to build the polyhedrons/polygons from all hexahedron/quadrilateral, tetrahedron/triangle, prisms and pyramids; or from their mixes are discussed.

The process starts by applying the 3-h edge refinement to each edge of the mesh, i.e. dividing each edge into three equal-size segments. Newly inserted points are reconnected across each edge to encircle the original nodes of the mesh. Then, agglomeration of sub-grid cells is applied to form a polyhedron centered at the nodes of the original mesh. (Concept of refinement and reconnection is shown for a two-dimensional mixed triangular and quadrilateral mesh in Figure 1.) The truncated cells of the original mesh form another type of polyhedrons. In short, one type of polyhedrons will form around the nodes of the original mesh; another type of polyhedrons will form from the leftover of the cells in the original mesh. The number of the polyhedrons will be equal to the sum of the number of the nodes and the number of the cells of the original mesh.

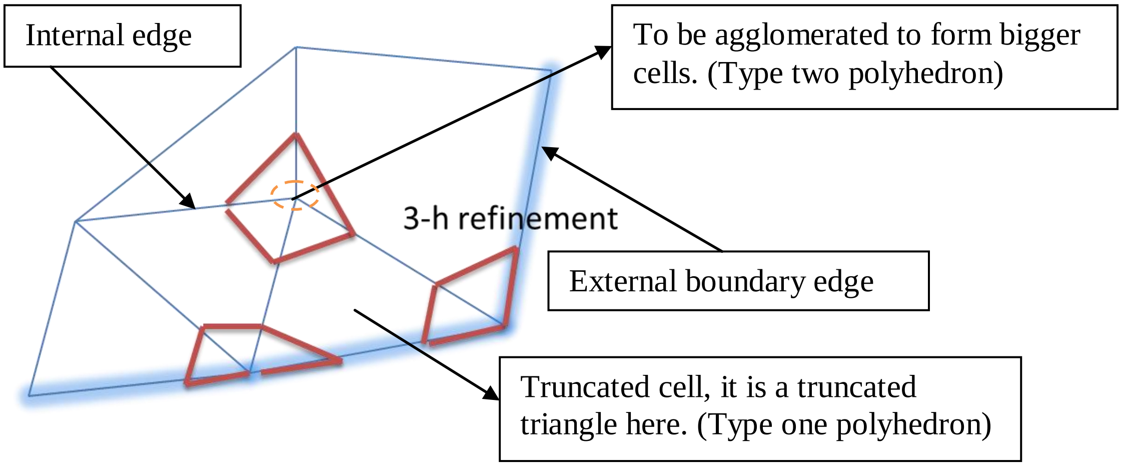
Internal edge
To be agglomerated to form bigger cells. (Type two polyhedron)
3-h refinement
External boundary edge
Truncated cell, it is a truncated triangle here. (Type one polyhedron)

Figure 1 Apply the 3-h refinement to each edge and reconnect the inserted points to encircle the original nodes for type two polyhedron. Leftover of cells becomes type one polyhedron.

In Figure 2, polyhedrons are derived from a set of hexahedrons, while in Figure 3; polyhedrons are derived from a set of tetrahedrons.

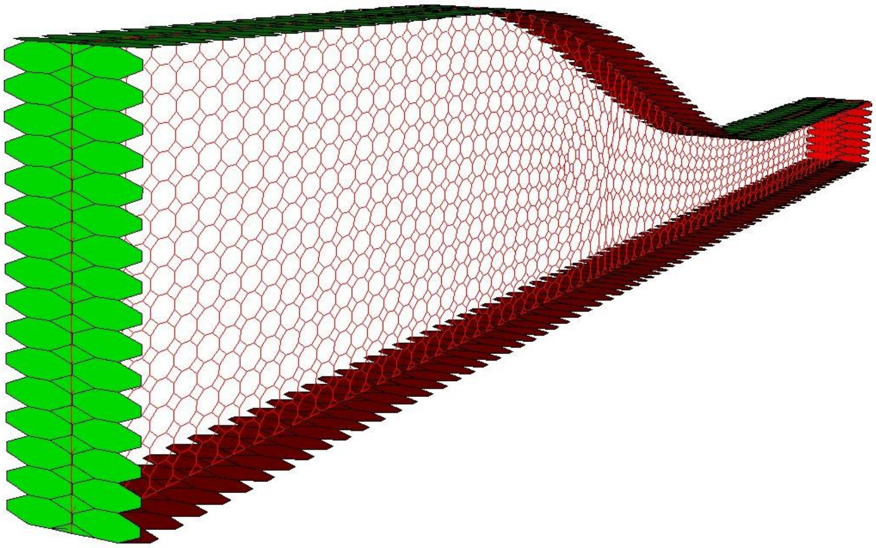

Figure 2 Partial boundary polygons and internal plane cut of polyhedrons generated from a set of all-hexahedron inviscid mesh. (Pattern of staggered volumes/areas is quite noticeable.)

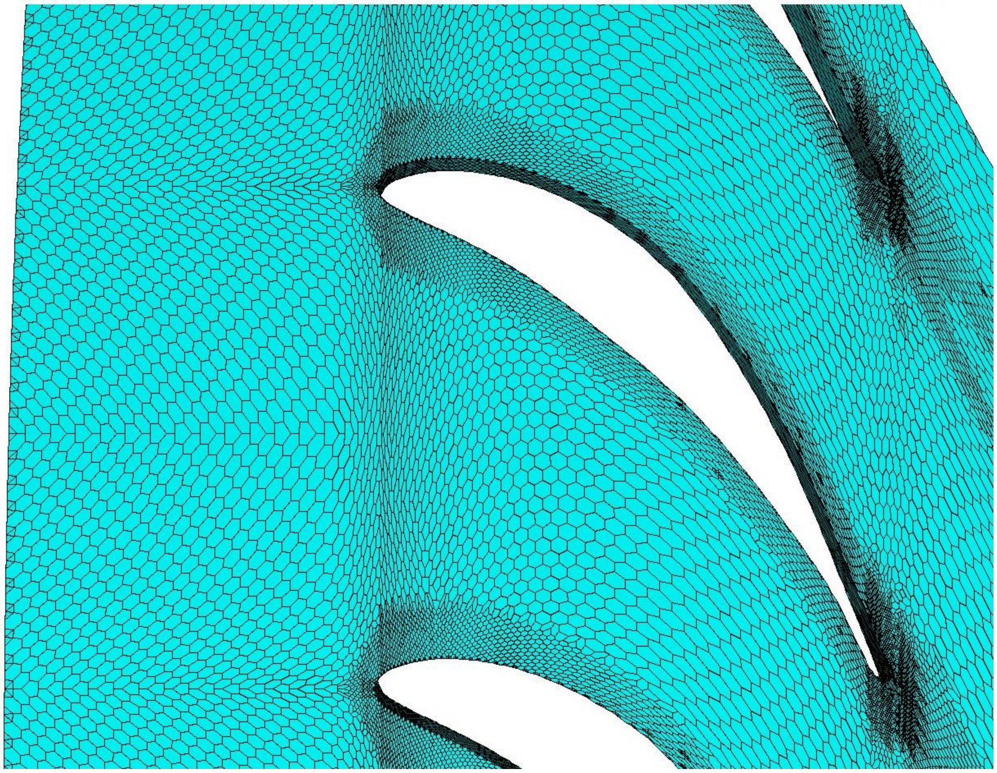

Figure 3 Boundary polygons of a polyhedron generated from a set of all-tetrahedron rotor mesh. (Pattern of staggered volumes/areas is not noticeable.)

For a given triangle mesh; N is the number of nodes, 2N is the number of cells, then the derived poly-mesh will have N+2N cells. The count of cells increases by 50%. For a given tetrahedral mesh, N is the number of nodes, 5~6 N is the number of cells, then the derived poly-mesh will have N+5~6N cells. Its cell count increases by 15~20%. In the case of hexahedron mesh, the cell count of its derived poly-mesh increases by around 100%.

It is noted here that another very popular polyhedron mesh generation method, which uses midpoint-based rule to solely convert tetrahedron/triangle into polyhedral mesh, actually reduces the number of derived cells. That method is equivalent to store unknowns at the nodes of the original tetrahedron/triangle mesh instead of at the centroids. This results in much less number of polyhedrons because the number of nodes is less than that of centroids for a tetrahedral/triangle mesh. On the one hand, less unknowns use less computing resources. On the other hand, less number of solution elements in a given domain may have adverse effects on the calculated drag, temperature and scalar mixing, etc.

The current method is equivalent to store unknowns at the nodes together with the centroids of the original mesh. Undoubtedly, this requires more computing resources, due to the increased number of cells (i.e. unknowns). Furthermore, one numerical challenge associated with the refined cells near the boundary is that the standard explicit finite volume methods take the time step proportional to the size of a grid cell. This would typically require smaller time steps near the boundary as the results of the refinement. At the moment, the projection of the newly inserted grid points are not applied, thus the curvatures of the surfaces are not changed. By comparing the volume distribution of type one and type two polyhedrons, it is assessed that tetrahedron/triangle mesh is a better choice than hexahedron/quadrilateral mesh as the basis for polyhedron generation. For example, the area ratio of type-one polygon to type-two polygon is about one for the polygons generated from a set of unit-length triangles, while the area ratio of type-one polygon to type-two polygon is about 14:4 for the polygons generated from a set of unit-length quadrilaterals. The disparity of this volume ratio is much worse for three-dimension cases. Another advantage of choosing a tetrahedron mesh as the starting mesh for polyhedrons is that the generation of conforming tetrahedral mesh is much more robust and automatic than the generation of conforming hexahedral mesh.

Domain Decomposition of Polyhedral Mesh

Domain decomposition is a very important subject for massively parallel computing, especially when the number of faces in an arbitrary polyhedron mesh is much greater than that in a regular unstructured mesh. In the present work, METIS 4.0.1, the freely available software from University of Minnesota (Reference 4), is used in the partition of the computational domain. Although METIS 4.0.1 currently supports only four basic element types: triangles, quadrilaterals, tetrahedrons, and hexahedrons, i.e., not including the arbitrary polyhedrons, fortunately, PARTDMESH will produce both a list of the elements belonging to each part of the partition and a list of the nodes belonging to each associated part of the pseudo-partition of the nodes. Since current arbitrary polyhedral mesh utilizes both elements and nodes of the original mesh, by concatenating these two lists, the final list of partition information for the arbitrary polyhedral mesh is established.

Spray Droplet Search in Polyhedral Mesh

Some polyhedrons at the corners and ridges of the boundary faces could be concave. Concave polyhedrons will create issues for the routines that are related to the turbulence modelling and the search of the spray droplets. In the present work, the centroid, i.e., the geometric centre of the cell, is used as the solution center for a convex polyhedron. For a concave polyhedron, the centroid of the visible region of the cell is used as the solution center. The visible region of the cell is defined as the union of the points that is visible from any point on the boundary of the cell and vice versa, i.e., it can see any point on the boundary. Based upon this visibility restriction, any ray from the boundary of the cell to the visible solution center will not intersect with any other boundary, and any droplet contained in this concave cell can be detected easily. In addition, the normal distance from the wall surface to the solution centre is always positive, and this is essential for turbulence wall function calculation.

Sample Applications of Polyhedral Meshes

(1) Non-reacting Cases

Although hexahedron mesh is not a good basis for polyhedron mesh generation, an all-hexahedron mesh describing a single-element LDI combustor is selected for the purpose of demonstration, because of the availability of the grid and serving as the worst case scenario. The geometry of the single-element LDI combustor consists of an air swirler, a convergent-divergent venturi, followed by a rectangular combustion chamber. The fuel injector tip is at the throat of the venturi. The original mesh contains 329312 hexahedrons, 346835 nodes and 1005272 faces. The face-to-cell ratio is about 3.05. The derived polyhedral mesh contains 676147 polyhedrons, 2080262 nodes and 3691282 faces. The face-to-cell ratio jumps to 5.46, which represents an 80% increase of the available flux- exchange sites. In Figures 4 and 5, two plane cuts of the polyhedrons are shown.

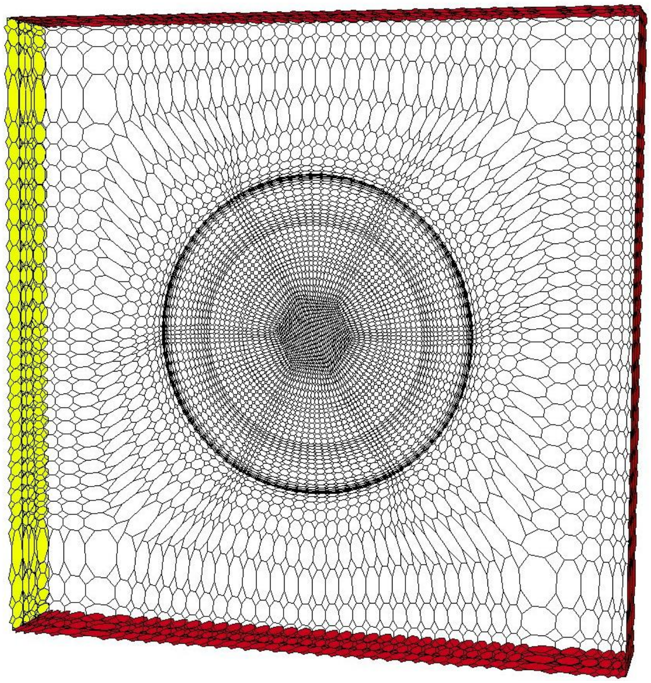

Figure 4 Partial boundary polygons and a plane cut of polyhedrons at an axis location near the dump plane of the combustor.

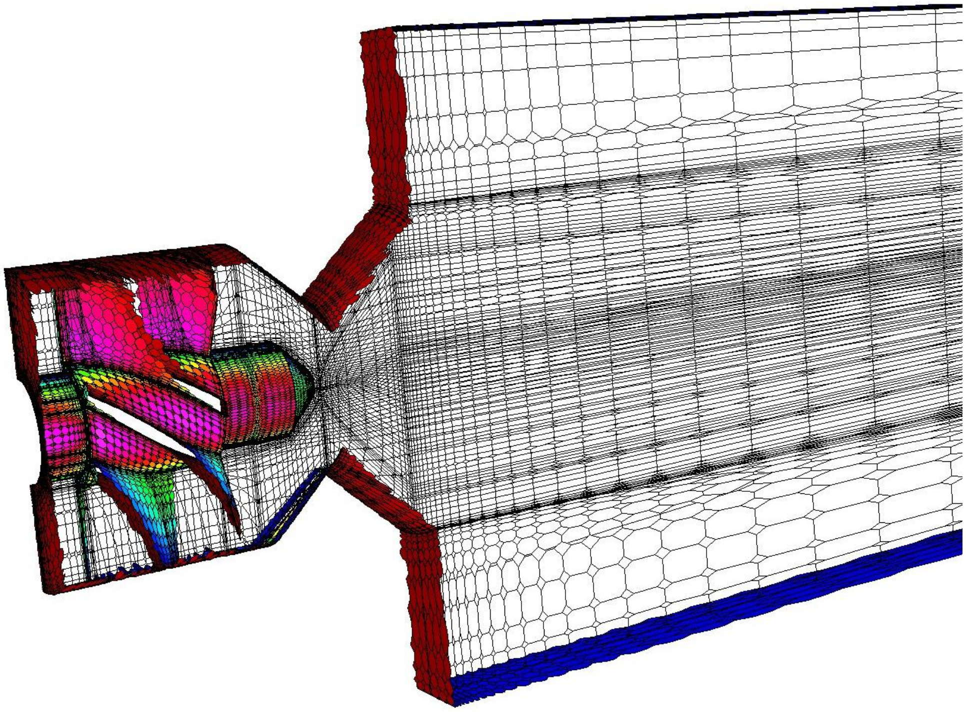

Figure 5 A plane cut of polyhedrons around the mid-plane of the combustor.

This set of polyhedrons and the following non-reaction gaseous conditions are used for simulation. At the inlet, the inflow velocity, the static temperature and the density of the gas phase are specified as 20.14 m/s, 294.28 K, 1.19 kg/m^3 respectively. At the outflow boundary, the static pressure is imposed at 101325 Pa. The generalized wall function is applied to solid wall boundaries. The thermal boundary condition for all the solid surfaces of the combustor is set to adiabatic. An URANS (unsteady RANS) simulation is carried out for this mesh.

Computed axial velocity on the plane at z=0 is shown in Figure 6. The pressure contours on the plane at z=0 is given in Figure 7. A low pressure center (corresponding to a vortex core) can be seen very clearly.

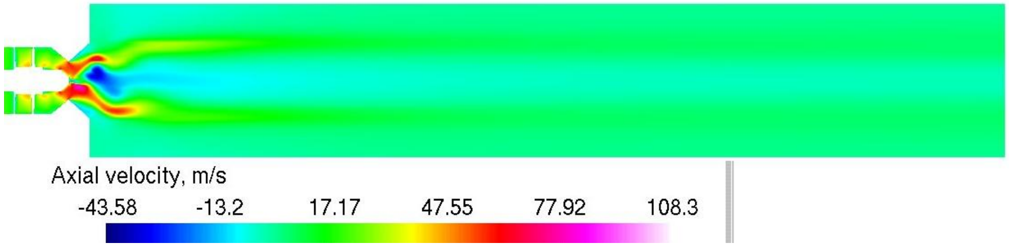

Figure 6 A snapshot of axial velocity contours at z=0 plane.

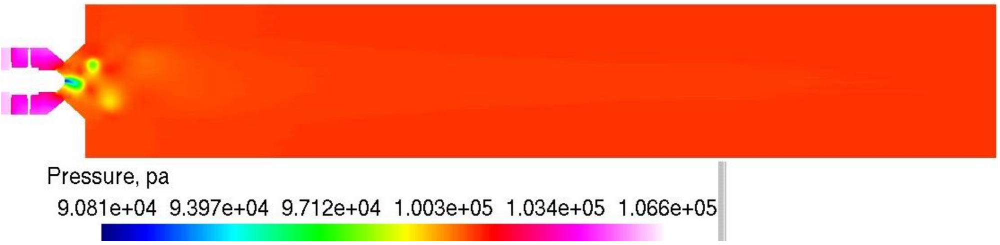

Figure 7 Pressure contours at z=0 plane. Cross sections of low pressure vortex core can be seen clearly.

(2) Multi-phase Reacting Cases

For the multi-phase reacting flows, two sets of denser grid are generated. One is a grid consisting of 1014784 hexahedrons, the other one consists of 1009034 polyhedrons. The latter grid is derived from a coarse grid of 494560 hexahedrons. The number of solution elements is compatible between the denser grids, but the surface definition of the former grid is better than that of the latter grid, because the re-projection to original geometry is not applied for the newly inserted boundary points of the latter grid.

In addition to the boundary conditions for the gaseous phase described above, the liquid fuel, C12H23, is injected into the computational domain from a pressure swirl injector. The diameter of the orifice is .0006 m. The fuel atomizer is set at a pressure 110kPa and delivers a flow rate of 0.025 kg/min. The spray has a 90 degree spray angle which is the same as the converging-diverging venture attached to the swirler. An initial droplet size distribution is prescribed to provide the liquid fuel injection condition,

$$\frac{dn}{n} = 4.21\times 10^{6}[\frac{d}{d_{32}}]^{3.5} e^{-16.98(\frac{d}{d_{32}})^{0.4}} \frac{dd}{d_{32}}$$

where n is the total number of the droplets and dn is the number of droplets in the size range between d and $d+dd$. This correlation also requires the specification of Sauter mean diameter, d_{32}, and the number of droplet classes. The equivalence ratio computed from the gaseous and liquid inlet boundary conditions is about 0.72. The adiabatic flame temperature is around 2100 K.

From Figure 8 to Figure 10, URANS results obtained with the well-mixed combustion model and from using the polyhedral mesh are shown for axial velocity, pressure and temperature in the center plane (i.e. z=0 plane), respectively.

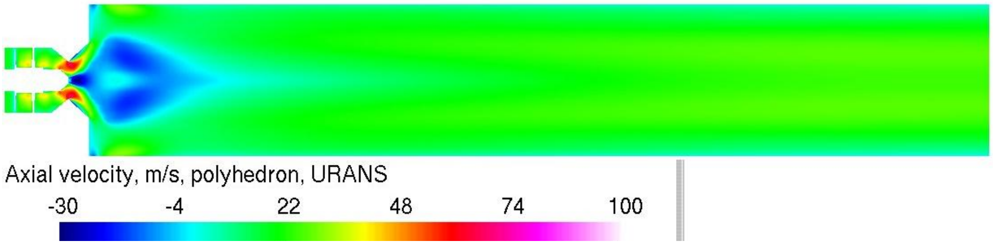

Figure 8 Axial velocity contours of spray-reacting mean flow at z=0 plane. (1009034 polyhedrons)

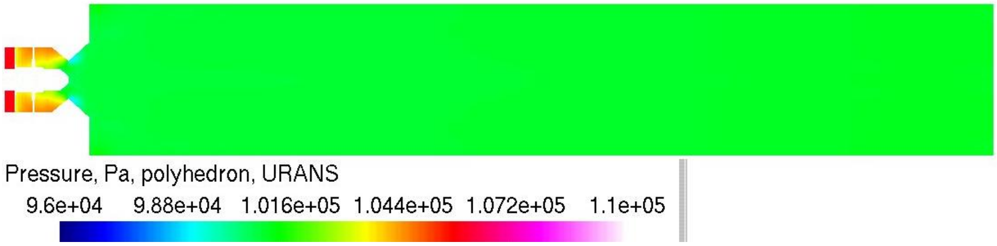

Figure 9 Pressure contours of spray-reacting mean flow at z=0 plane. (1009034 polyhedrons)

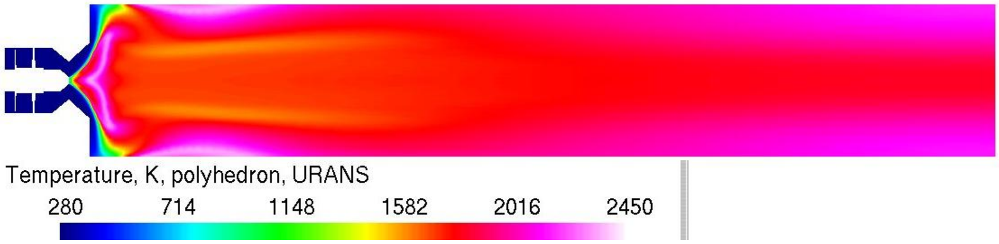

Figure 10 Temperature contours of spray-reacting mean flow at z=0 plane. (1009034 polyhedrons)

From Figure 11 to Figure 14, URANS results obtained with the well-mixed combustion model and from using the compatible-resolution hexahedral mesh are shown for axial velocity, pressure and temperature, respectively.

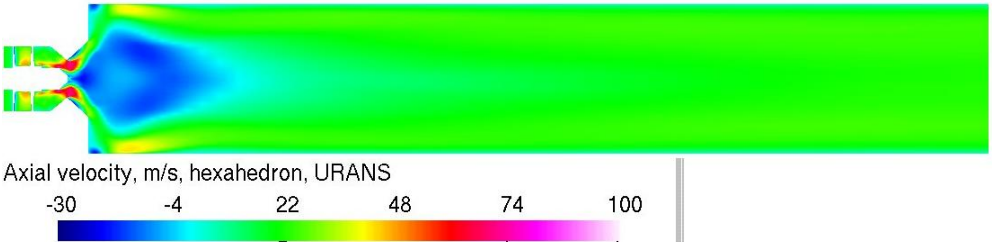

Figure 11 Axial velocity contours of spray-reacting mean flow at z=0 plane. (1014784 hexahedrons)

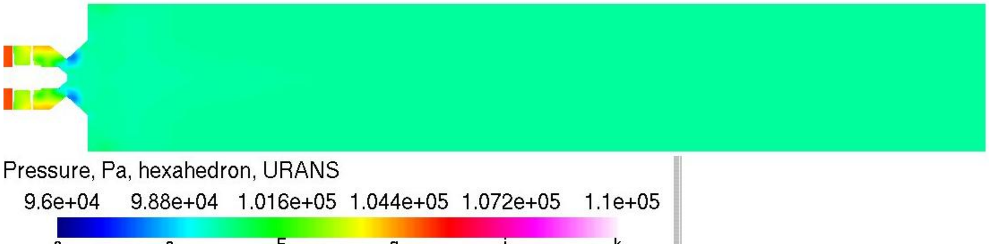

Figure 12 Axial pressure contours of spray-reacting mean flow at z=0 plane. (1014784 hexahedrons)

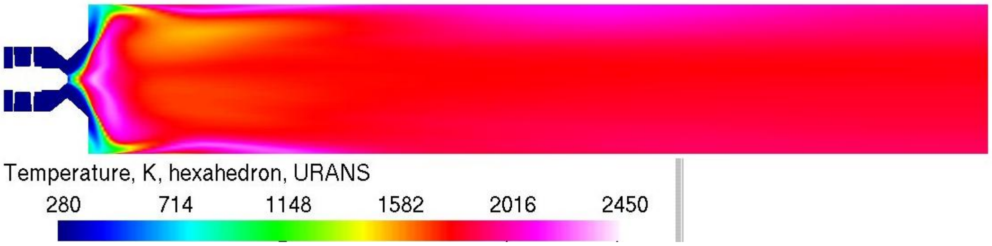

Figure 13 Axial temperature contours of spray-reacting mean flow at z=0 plane. (1014784 hexahedrons)

In the following, the reacting results obtained from the time filtered Navier-Stokes simulation (TFNS) using the linear eddy mixing (LEM) model for turbulent combustion will be presented. For sub-grid scalar field evolution, 24 LEM cells are used within each and every TFNS cell, and up to 40 sub-steps are set for the stirring event and molecular diffusion. The species mass fractions are provided by the solution of the LEM module, while the temperature, velocity and pressure fields of the overall simulation are provided by the TFNS module. It is also noted here that the solutions of URANS simulations are used as the starting condition for the LEM/TFNS simulations.

From Figure 14 to Figure 16, LEM/TFNS results from using the polyhedral mesh are shown for the time-averaged axial velocity, pressure and temperature in the center plane (i.e. z=0 plane), respectively.

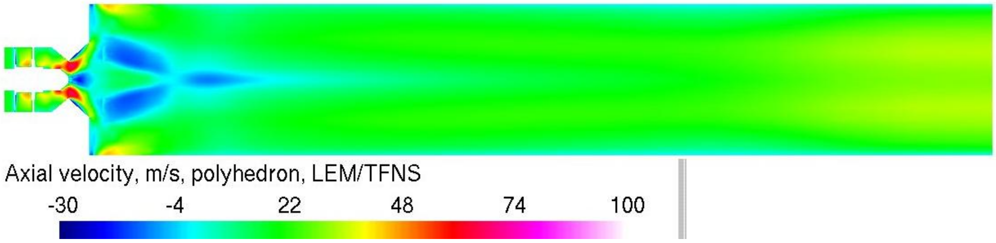

Figure 14 Contours of the time averaged axial velocity in the center plane by LEM/TFNS. (Polyhedral mesh)

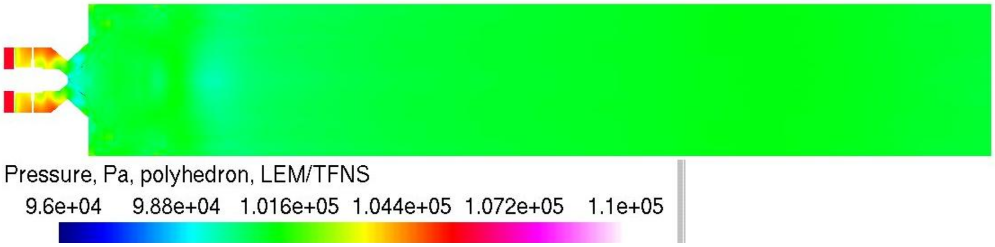

Figure 15 Contours of the time averaged pressure in the center plane by LEM/TFNS. (Polyhedral mesh)

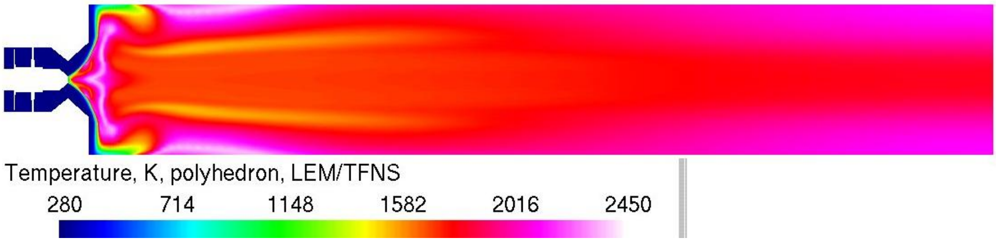

Figure 16 Contours of the time averaged temperature in the center plane by LEM/TFNS. (Polyhedral mesh)

Figure 17 shows a schematic of the evaporating jet fuel droplets in the center plane and x=0.04 m plane. The rapid disappearance of the spray droplets around x=0.04 m is perhaps influenced by the higher temperature zones around x = 0.04 m near the wall (see Figure 16).

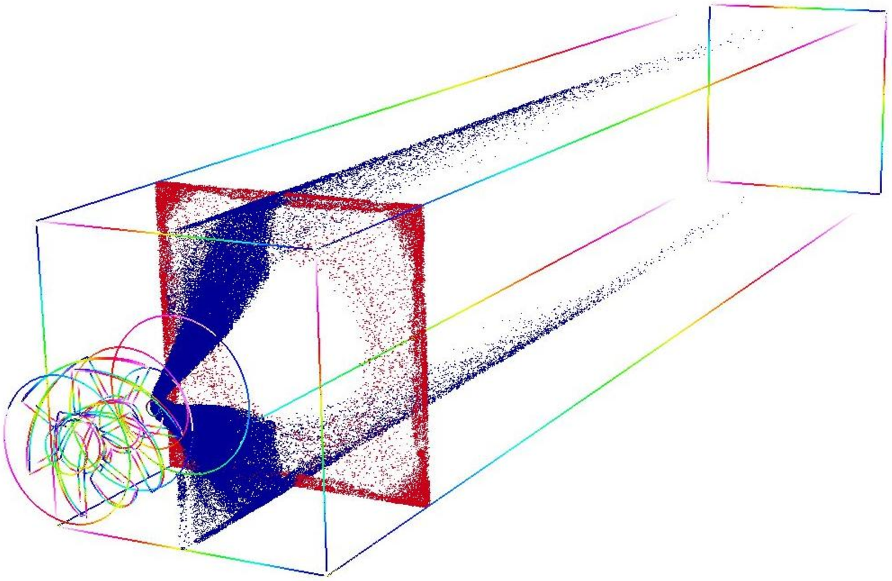

Figure 17 Spray droplet distributions on z=0 plane and x=.04 m plane. (polyhedral mesh)

From Figure 18 to Figure 20, LEM/TFNS results from using the compatible-resolution hexahedral mesh are shown for time-averaged axial velocity, pressure and temperature, respectively.

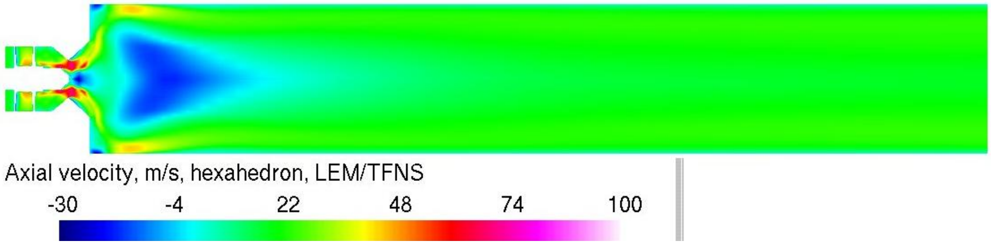

Figure 18 Contours of the time averaged axial velocity in the center plane by LEM/TFNS. (hexahedral mesh)

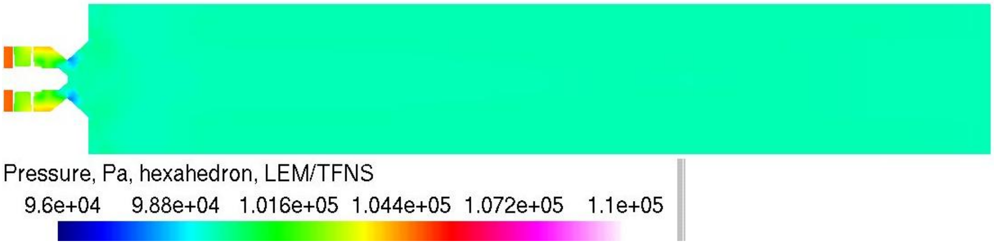

Figure 19 Contours of the time averaged pressure in the center plane by LEM/TFNS. (hexahedral mesh)

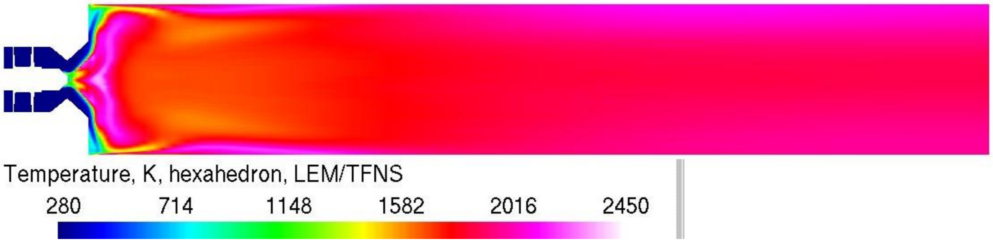

Figure 20 Contours of the time averaged temperature in the center plane by LEM/TFNS. (Hexahedral mesh)

The time-averaged centerline axial velocity and the centerline temperature are presented in Figures 21 and 22, along with the measured data.

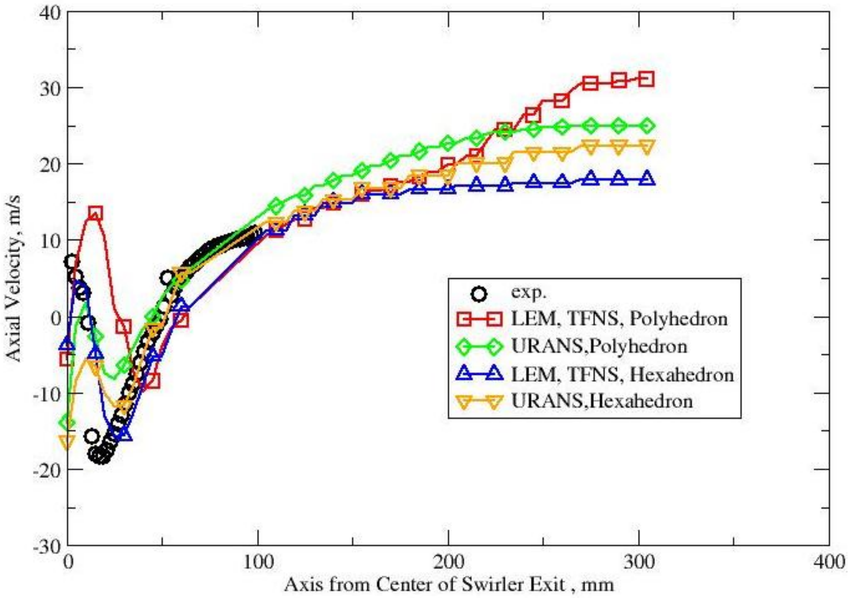

Figure 21 Comparison of the mean axial velocity along the center line.

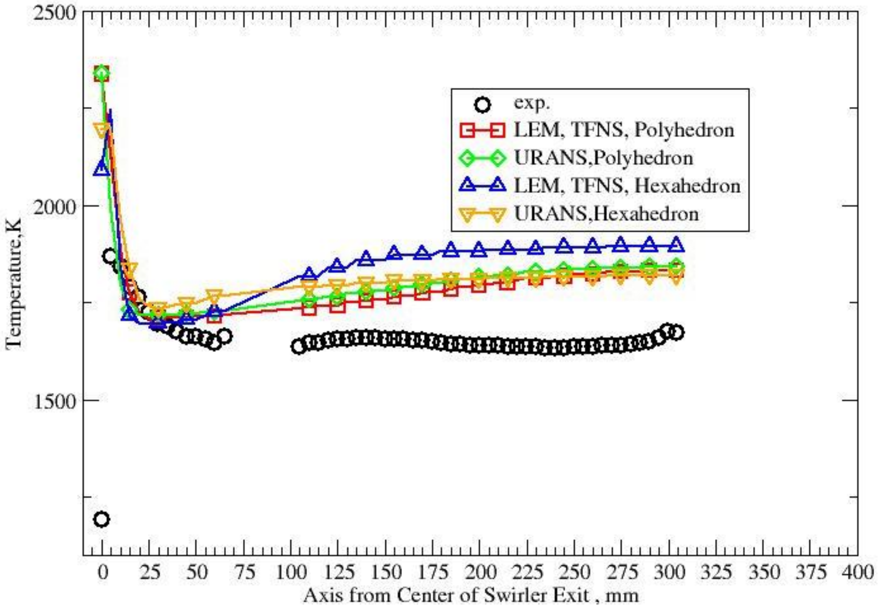

Figure 22 Comparison of the mean temperature along the center line.

Concluding Remarks

Capability of generating and using arbitrary polyhedral mesh for simulations of multiphase reacting flows is now embodied in a preliminary version of the OpenNCC, which is intended as the self-contained, releasable edition of the National Combustion Code (NCC). A stand-alone single-element LDI combustor is selected for the demonstration. It is observed that the results from a polyhedral mesh refined from an all-hexahedron coarse mesh are not better than those from a similar-resolution all-hexahedron mesh. And this is due to the larger disparity of volume distributions in the hexahedron-derived polyhedrons. Since the distribution of volumes of the type one and type two polyhedrons derived from the tetrahedron/triangle is much smoother, future effort will focus on the investigation of results from a polyhedral mesh derived from a tetrahedral mesh. This compliments an ongoing, parallel effort on improved adaptation of the LEM-like sub-grid turbulent combustion model into the TFNS framework.

Acknowledgements

This work is supported by the Subsonic Fixed Wing Project and the Supersonics Project under the NASA Fundamental Aeronautics Program.

References

[1] Liu, N.-S., “Assessment and Improvement of Engineering Simulation for Multiphase Turbulent Combustion in a Lean Direct Injection Combustor,” ISABE-2011-1108, 20th International Symposium on Air Breathing Engines, September 12-16, 2011, Gothenburg, Sweden.
[2] Liu, N.-S., Shih, T.-H., and Wey, C.T., “Numerical Simulations of Two-Phase Reacting Flow in a Single-Element Lean Direct Injection (LDI) Combustor Using NCC,” NASA/TM-2011-217031, July 2011.
[3] Wey, T., “Development of A Grid Generation Technique for Unstructured Meshes,” AIAA 92-2718, 10th Applied Aerodynamics Conference, June 22-24, 1992, Palo Alto, CA.
[4] Karypis, G. and Kumar, V. “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359—392, 1999.