Incorporating Computational Fluid Dynamics into the Preliminary Design Cycle

Jonathan K. Shelley[1], Nicole C. Giullian[2] and C. Greg Jensen[3]

[1]Pratt and Whitney, jonathan.shelley@pw.utc.com
[2]Brigham Young University, nicole.giullian@gmail.com
[3]Brigham Young University, cjensen@byu.edu

ABSTRACT

Industry is constantly seeking ways to bring new or derivative products to market in minimal time at minimal cost. To accomplish this, industry has adopted Computer Aided Engineering tools that perform structural, flow, manufacturing, and cost analysis. One of these tools industry often struggles with in the preliminary design phase is Computational Fluid Dynamics (CFD). Some of the challenges presented by CFD are the time required to create a CAD (air solid) model, generate a valid grid for analysis, obtain a solution, post-process results, and review the results. We present an approach that reduces the time to go from a concept to a solution ready for review. This approach addresses how to build a CAD model for use in downstream applications, automate the grid generation process, and automate the post-processing. In addition, it addresses how to automate results documentation with a Design Review Tool. This approach will enable the aerospace, automotive, and other industries to use CFD to more effectively explore the design space in the development of new and derivative products.

Keywords: CFD, automation, parametrics

1. INTRODUCTION

Today's highly competitive marketplace demands high quality, low-cost products in a timely, rapid fashion. Effective product development and release requires planning, concept development, system-level design, detailed design, testing and refinement, and production ramp-up [1]. Often, however, due to limited resources and competitive pressures, many concepts are carried forward to the detailed design phase based on only partial exploration of the design space. Most companies' detailed design phase is littered with technical challenges caused by this premature termination of the conceptual and preliminary design phases. Ideally, all possible concepts would be explored before selecting the one that best meets requirements. Tools like computer-aided design (CAD), finite element analysis (FEA), and computational fluid dynamics (CFD) software provide simulations to help engineers understand the problems and challenges of particular concepts. However, industry experience confirms that preparing the first CAD model(s) for analysis takes weeks to months [2].

As computing power continues to expand, CAD models are being integrated with CAE tools [3]. Integrated design tools allow numerous design characteristics to be considered. Parametric CAD models store the necessary information needed for the grid generation phase in object attributes. Parametric models help alleviate the problem; however, one of the current limitations to using a direct translator in an automated way is that parametric changes to the geometry can result in topological changes [4].

In industry, CFD modeling of complex geometry is commonly used in the design process [5]. However, the challenges facing designers in the use of CFD in preliminary design stagnate the efficacy of CFD in preliminary design. Air solids created from CAD models contain sliver faces and non-manifold topology, and these problems make grid generation difficult, even impossible. Even with a good model, grid generation "is the most labor intensive and time consuming part of computation aerosciences" [6], often taking weeks to months to go from CAD to the first CFD case [2]. To better utilize the array of modern CAD/CAE tools and the company's computing resources, a methodology is needed to reduce the time required to go from a CAD model to a mesh. Another major challenge to using CFD in conceptual and preliminary design is the amount of data that must be processed and presented. It can take hours to document each case manually. This process delays the review of the results until a formalized presentation is made. A method is needed to process and format data consistently for quick review. All of the above challenges take valuable time that could instead be used to incorporate analysis more fully into the concept generation stage or to explore more concepts before moving to the detailed design phase [7].

This paper presents one approach to solving these problems that has been effective in preliminary design of jet combustors. This approach involved first creating valid CFD grids in an automated way from parametric models. These models were created so that changes at the component level would be transmitted throughout the assembly. A significant amount of time was required to create the first parametric model, but this initial time investment dramatically reduces the time required for all future model analysis. The grid generation and boundary conditions for the model were done with macros, a method as robust as parametric modeling. A Design Review Tool (DRT) was created to standardize the output data and increase the turnover time from results to review. This tool created a modular output structure whose results sections can be modified, added, or removed.

2. METHOD

Currently, the CFD process is manual, error-prone, and too time intensive for use in the preliminary design of complex models. This work focused on the developing interfaces between the required CAx tools. Fig. 1 shows the necessary steps to incorporate CFD in preliminary design.

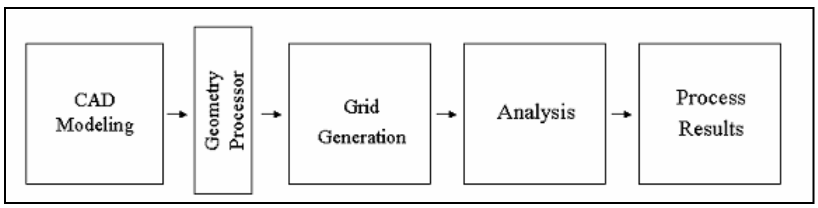

Fig. 1: Elements for Incorporating CFD into Preliminary Design.

The following approach describes in detail the steps used to handle topology changes as well as assembly component replacement.

2.1 CAD Modeling

The CAD modeling strategy consists of three steps:

1. Construct a Master Cross Sectional Sketch (MCSS)
2. Model the component using the MCSS references
3. Assemble the various components

2.1.1 Construct a Master Cross Sectional Sketch (MCSS)

The MCSS defines key references and features (datum planes and axes, assembly features, component locations, etc) for an assembly. A sketch on a datum plane defines an assembly plane or interface for two components. Using common datums in the MCSS to locate the different components reduces the time required to create the assembly and ensures that changes will consistently propagate through the entire assembly. Once the master sketches are created, the designer(s) can use the same sketches to ensure that the interface points and locations are consistent.

2.1.2 Model the Component Using the MCSS References

For the components to take advantage of the MCSS information, inter-part geometry and inter-part dimensioning must be used. Inter-part geometry linking allows components to reference the geometry defined in the MCSS. The linked geometry adjusts according to changes in the MCSS. If used properly, inter-part geometry linking helps to ensure that the overall assembly will maintain design intent during design iterations. Inter-part dimensioning passes information from the MCSS to the individual components by creating assembly level parameters in the MCSS then linking them to the individual components. Once the component is modeled, attributes needed for mesh generation are assigned to the different geometric entities. Defining the element size for each face and curve often requires a few iterations before the different features are properly resolved, but once resolved, this information can be used for subsequent designs. However, when the topology is changed, the analyst must recognize the changes, tag any new geometric entities, and ensure the new entities will be properly resolved.

2.1.3 Assemble the Various Components

Once the components are modeled, they can be added to the assembly. If modeled correctly, the various components require little or no information for correct assembly. However, interface surfaces may be required to close gaps otherwise left open in the air solid. Assembly expressions are used to toggle the parametric components and interface surfaces on or off. An input file is used to modify the assembly and thus represent different concepts.

2.2 Geometry Processor

The geometry process step sorts through the model and finds any entities, such as edges, that are not currently tagged then assigns attributes as needed. For predetermined topological changes, the geometry processor must recognize the new topology and tag the appropriate entities. The geometric processor must do the following:

1. Collect the attribute names that may contain topology changes.
2. Modify the expressions and update the CAD model.
3. Search the entities for specified attributes names.
4. Check entities to see if they are tagged.
5. Assign the proper attributes to any new topology.
6. Save the model.

2.3 Grid Generation

ICEM CFD (a commercial grid generator) was used to create the grid. An initial grid must be created interactively and the steps recorded for use in a script file. This script then may be used to perform the following steps necessary for gridding: adding density volumes, generating the grid, smoothing the grid, and saving the grid for review. The review step is necessary to verify and fix the grid, if needed, before it is preprocessed.

2.4 CFD Analysis

Boundary conditions are the inputs required to solve the differential equations used in CFD analysis. Some common boundary condition types are inlet speeds, walls, periodic faces, symmetry planes, etc. Initial boundary conditions must be set before the model is solved. Once these conditions are set they usually require only minor changes, if any, for the subsequence derivative models. When performing a CFD analysis, there are three steps that are required in order to obtain the results: preprocessing, analysis, and post-processing.

Standard practices would be used for the preprocessing. These standard practices include checking the grid for errors, assigning the node boundary conditions, and breaking up the model to be run on multiple computers. When the grid is ready, the next step is analyzing the model. Since no experimental data is available to validate the results, CFD results were assumed to be valid. When the CFD solver has converged to a solution the final step is post processing. CEI Ensight, a commercial visualization software package, was used to perform the post processing. To automate this step a command file (macro) must be created. In general, the macro should open the analysis model, load the results file, calculate the desired variables, generate images of the model, output the data, and exit the program, all in a consistent manner.

2.5 Design Review Tool

The Design Review Tool (DRT) organizes the results gathered from the post processing step in a timely fashion and formats the results for later use. A standard file structure is required to allow the DRT to organize the data into a set of web pages that can be easily reviewed by other design team members. The web pages contain images of the concepts, tables containing flow conditions and results, and graphs displaying the results Presenting this data in an automated fashion reduces the time required to prepare a design review. It also enables the design team to look at the results as soon as they are available as well as storing them for later use.

3. DEVELOPMENT

The method presented above was applied to a two stream non-proprietary mixing problem. A simplified representation of a combustor in a gas turbine engine was used. Although the problem is simplified, the method has been applied to complex problems such as jet engine combustors with similar results. The simplified model consisted of three different components: a front-end, which introduces the hot stream and the outer and inner liners, which introduce the cold stream. Different concepts and configurations were explored to determine which provided the most uniform temperature at the exit plane.

3.1 CAD Model

The foundation for the two-stream mixing problem was the parametric CAD model. The following sections present the steps used for the creation of the components and assembly.

3.1.3 Constructing the MCSS

For the two stream mixing problem, a MCSS was created. The envelope of the assembly, datum references, and assembly interfaces were specified in the MCSS. First, to define the origin, three datum planes were created. Second, a datum axis was created to define the centerline of the two-stream mixing problem. Third, the assembly envelope, containing the mixing chamber and the OD and ID plenums, was laid out. Fourth, the injection location for the second stream of air was added. The MCSS for the mixing problem is shown in Fig. 2.

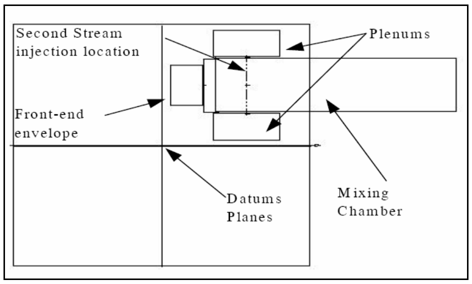

Fig. 2: MCSS for two stream mixing problem.

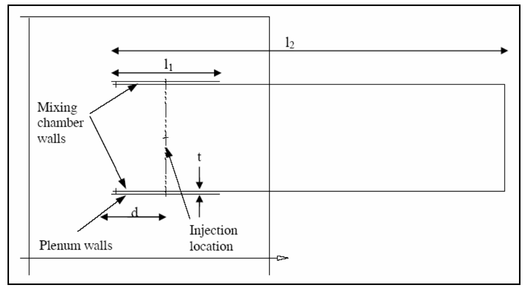

Fig. 3: Entities referenced from the MCSS.

The outer and inner lips indicate where the front-end mates up to the liners. The lines reference the datums and entities as shown in Fig. 3. These objects are used to define the inner and outer profiles, panel thickness, and the cold stream injection location. The dashed line defines the distance *d* downstream of the inlet, where the second stream is introduced. The shorter lines on the top and bottom represent the length l_1 of the outer and inner plenums. They also define the liner thickness *t*. The mixing chamber wall lines define the chamber length l_2. The points on the longer lines define where the front-end components will mate up with the liners.

3.1.2 Component Modeling

Each component was modeled by first adding the MCSS. After that, the datums, curves, and points in the MCSS were added to the component through a technique called waving. Standard dimensioning practices generally forbid chaining dimensions in this manner. But, by referencing these objects, any changes to the MCSS will propagate through all of the components. For the front-end, three different concepts were selected. The first concept has an array of four rectangular openings. The second has a cylinder with a flow separator in the middle. The third has a set of four cylindrical tubes. For the different liner concepts, three orifice shapes were selected: circular, elliptical, and triangular. The three different front-end and orifice concepts are shown in Fig. 4 and Fig. 5.

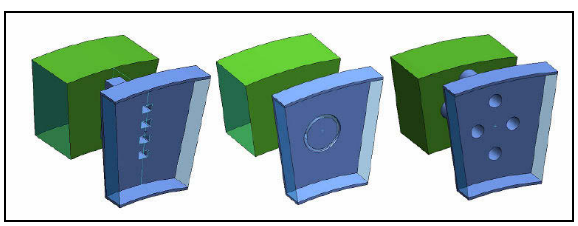

Fig. 4: Front-end concepts.

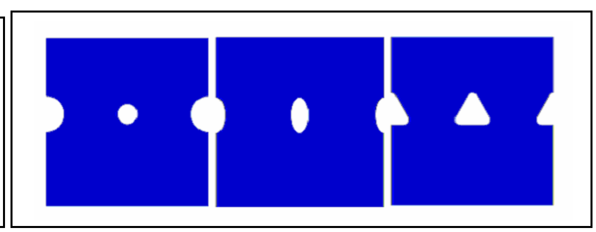

Fig. 5: Liner orifice concepts.

When constructing these concepts, certain modeling techniques were applied to ensure a valid model. First, the liner was revolved beyond the actual sector width. Second, the holes were added with the centroid of the area matching up with the second stream injection location. Each orifice was created separately to ensure the face IDs would not change. Third, all faces and curves needed for the mesh generation were tagged. Fourth, the liner was trimmed to the appropriate sector size. The reason for this is sometimes the holes will be on the periodic boundaries, which require one complete hole and two half holes. Other times the holes might not be on the periodic boundaries leaving only two complete holes. This difference can be seen in Fig. 6.

With the concepts modeled, the next step is to match the flow area for each component concept to ensure a fair comparison. To calculate the flow area (AC_d) needed for the desired pressure drop, Eqn. 3.1 was used.

$$AC_d = \frac{\dot{m}}{\sqrt{\frac{2\Delta p}{\rho}}} \tag{3.1}$$

A is the geometric area of the orifice, C_d is the discharge coefficient, $\dot{m}$ is the mass flow rate, ρ is the density of the fluid, and Δp is the difference between the total pressure upstream and down stream static pressure.

For round holes with sharp edges and a length over diameter (L/d) around 0.2 the C_d is approximately 0.7. For the oval and triangular holes the C_d is unknown and must be determined. To do this a C_d was guessed. A guess of 0.75 and 0.6 were selected respectively. To find the actual C_d for these holes, CFD simulations were performed. The C_d for the holes was calculated at the measured pressure drop from the CFD simulations. After front-end and liner concepts were modeled, they were assigned object attributes to resolve the geometry for the mesh generation step. The required faces, curves, and points were tagged using NX Mesher interface tool. For example, the faces tagged for a front-end concept are shown in Fig. 7.

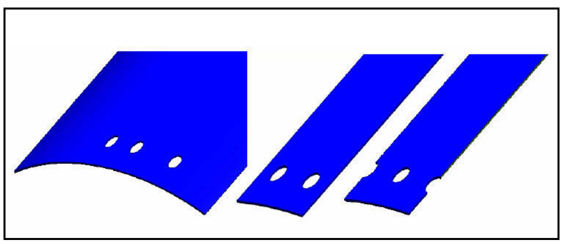

Fig. 6: Before and after the model was trimmed.

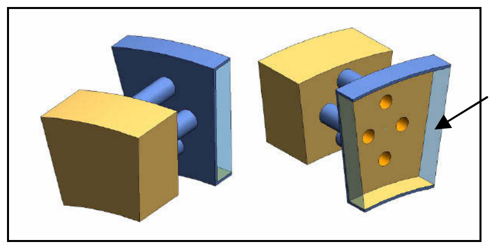

Fig. 7: Surfaces tagged for mesh generation.

The gray face on the right in Fig. 7 (marked with arrow) was tagged as the inlet family. The rest of the lighter faces were tagged and assigned to their respective families. The edges of the selected faces were also tagged. The other front-end concepts were tagged in a similar manner. For the liners, just the surfaces were tagged and assigned families.

3.1.3 Assembly Modeling

The assembly model was constructed by first adding the MCSS and then the individual components. To define which components were present for a configuration, “controlled by expression” parameters were used to toggle them on and off. Next, an air solid was created by revolving MCSS entities. Different front-end concepts have no solid portion for subtraction to occur due to the different topologies of each front-end. Therefore, the front-end concepts have two additional sheet bodies to provide periodic and inlet(s) surfaces that close the air solid.

Next, the outer diameter (OD) and inner diameter (ID) liners were subtracted from the air solid. When the components were added to the assembly and removed using “controlled by expression”, a few problems arose. First, a linked body was used to subtract the component from the air solid. When one component is toggled off and another on, the linked body of the inactive component remained, causing two of the same component to be subtracted from the air-solid. To resolve this issue, a feature set was created for each linked body and subtraction feature. Then, an expression was created that would suppress the feature set when the component was suppressed. When this was done the assembly model updated correctly for all configurations. With the air solid completed, the assembly was changed to a state which contained all of the potential periodic faces. This was done by moving the injection holes on the ID and OD liners off of the periodic faces.

In this configuration, the three faces were tagged as periodic faces. This was done to ensure that any new configuration of this model would have the periodic faces tagged. The model was then saved in this state as the master model. The model at this point was ready for the geometric processor.

3.2 Geometry Processor

The geometry processor was used to prepare the CAD model to be gridded. The geometric processor reads in the input file, opens the CAD model, imports the expression file, updates the model, and checks for the attribute names defined by the user which may contain topological changes. The topological changes include the number of edges and end points that are needed to resolve the geometry. The first line tells the geometric processor which CAD model to open. The second line indicated the name of the expression file to use to update the model. The rest of the lines define the families to check, the curve mesh size, and the name of the families to assign to the curves.

3.3 Grid Generation

To automate the grid generation process a Tcl script was created. This script was broken up into modules for reusability. The set command is used to set variables used in all of the modules. After the variables were set, the NX model was imported using one module then the grid input file was created using another module. Next, a file containing the density volumes was merged with the input file and saved; finally, the grid was created and smoothed. When the model was smoothed it was saved for visual inspection. Visual inspections are needed to ensure that the geometry was imported correctly and that the resulting grid was valid. The valid grids were then exported to a PATRAN neutral file format for analysis.

3.4 CFD Analysis

The analysis phase consisted of specifying the boundary conditions, preprocessing the grid generated in ICEM, analyzing the model, and post processing the solution to visualize the results.

3.4.1 Preprocessing

For the cases in this study, preprocessing was needed to read in the grid from ICEM, which converts it to a model file that can be used by the solver. After the model file was created, it is broken up for parallel processing. At this point the model was ready to have boundary conditions specified.

3.4.2 Boundary Conditions

For the two stream mixing problem, the mass flows of the two streams were specified. The hot stream flow rate was set to 0.033 lbm/s at 500 °F. The cold stream flow rate was set to 0.066 lbm/s at 70 °F. The desired pressure drop across the liners was 3%. To determine the flow velocity into the plenums, Eqn. 3.2 was used:

$$V = \frac{\dot{m}}{\rho A} \tag{3.2}$$

where V is the velocity, $\dot{m}$ is the mass flow rate, ρ is the density of the fluid, and A is the flow area. Using this equation, the mass flow rate for the first and second streams was calculated.

3.4.3 Analyzing the Model

To arrive at a solution, the solver iterated until the parameters converged to their final values. It was determined that it required around 2000 iterations for the solution to converge. When converged, the solution was checked to verify that the flow field had converged. Next, each case was post-processed to view the results.

3.4.4 Post process the Results

The post processing of each case was done using an Ensight command file, a macro file generated by interactively recording the post processing steps. The file was edited to repeat certain commands at different locations, calculate variables, and generate images of the flow field for the desired variables. For each case, the script first opened the model and read the results. Then, different screen grabs of the model and grid were taken. Next, a pressure map down the centerline of the model was created. This provided a visual picture of the pressure field as a sanity check. Then, the script took point measurements in the flow field to find the pressures and calculate the pressure drop. When the pressure portion of the script was complete, the script generated temperature slices of the flow field at different locations to visualize the mixing of the flow field. At each slice, a maximum and a minimum temperature were recorded. Finally, the script wrote the calculated scalar variables into an output file.

3.5 Design Review Tool

Upon completion of the post processing of a case, the Design Review Tool (DRT) was employed to document each solution. It provided a standard method to review each solution. The DRT is a Python script used to generate Hyper Text Markup Language (HTML) pages. The HTML pages document each case. To set up the DRT, certain variables were modified to indicate the location of the study directory and the prefix used for each CFD case.

To allow for code reusability, each section used on the case page was created using separate modules known as classes. To display more information or change the formatting, all that is required is to modify the respective class and rerun the DRT.

The first web page created by the DRT displays a listing of all of the cases with the predicted temperature spread at the exit plane. From the "Two Stream Mixing Study" page, the design team can see which cases are completed with the objective value and which cases are in progress. The finished cases are selectable and their results can be viewed by selecting a case number. By selecting a case number for a completed case, the case page is displayed. This page provides a quick glance at the different sections of the case as seen in Fig. 8.

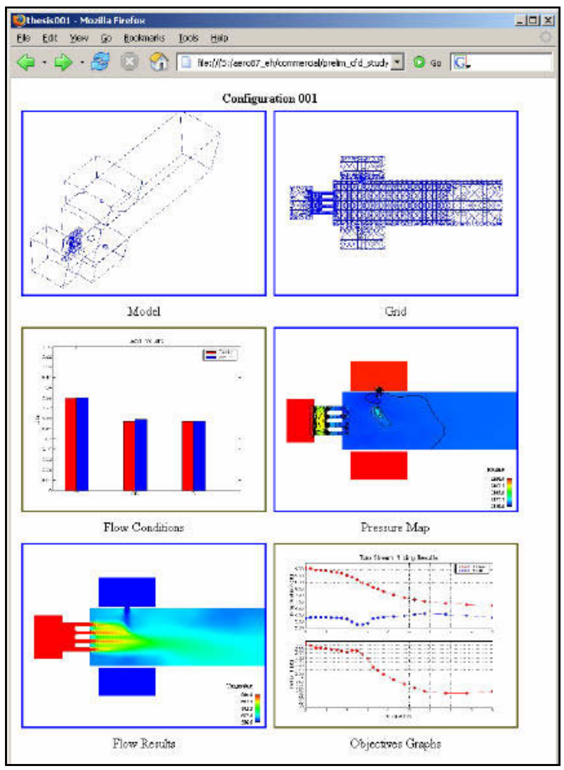

Fig. 8: DRT Case template.

Each section can be reviewed by clicking on an image. This allows the user to review the model and solution. The model section displays a wire frame view of the model along with the expression file used to update the CAD model. The grid section shows a slice of the grid, with a table listing the families and elements that are tagged as boundary conditions. The flow condition page provides a visual check of the flow conditions. The flow conditions and pressure map section help verify that the CFD solution was run properly. The final two sections, flow results and objectives, provide a visual representation of the results for each case. Only two variables were considered in the flow results section, however, more images could easily be added to this section by modifying the flow results module. From the flow results section page the different parameters are shown and can be reviewed by clicking on the image. When the temperature image is clicked, all of the temperature images generated are displayed.

The objective section provides a visual representation of the how quick the two streams are mixing and the temperature ranges of the flow as it progresses downstream. On the right hand side of the page a graph shows the maximum and minimum temperature in the flow field. Clicking on the graph displays an enlarged image.

4. DISCUSSION OF RESULTS

This section presents the results of the CAD model, geometry processor, and the grid generation as applied to the conceptual study of the two stream mixing problem. The application of the method resulted in a process that could be used to analytically screen many different concepts, not only combustors, quickly and efficiently. The results of the grid automation process and the Design Review Tool (DRT) are also presented. To better understand the work done to automate the grid generation process one must compare it to the current process in industry.

4.1 Results: Automated Grid Generation of Parametric Models

Automating the CAD-to-Grid process for parametric models was aided by the creation of the Master Cross Sectional Sketches (MCSS) and component modeling techniques. When these steps were added, the air solid assembly came together quickly and allowed for quick interchangeability of components. A standard process was then captured in a script to generate the grid for each new configuration.

4.1.1 Results: MCSS

The MCSS was essential in defining the critical assembly requirements: the injection point of the two mixing streams, the mixing chamber height and length. Creating the MCSS required approximately an hour, but provided the parametric framework for the different concepts. The MCSS ensured that each component in the assembly aligned properly. This approach added an extra step to standard practices used today, but reduced the time required to assemble each component to around 15 seconds and ensured they were interchangeable.

4.1.2 Results: Component Modeling

Each concept was constructed using geometric entities referenced from the MCSS. This allowed the concepts to have common entities and datums. In addition, tagging at the component level for the grid generation phase made it possible to interchange them in the assembly. When compared with the standard practice this takes upfront thought and time, but the payoff comes in the interchangeability of components.

4.1.3 Results: Assembly Model

Using the MCSS as a framework, the assembly model was able to swap the different front ends and liners in and out. The assembly model of the different cases shown in Fig. 9.

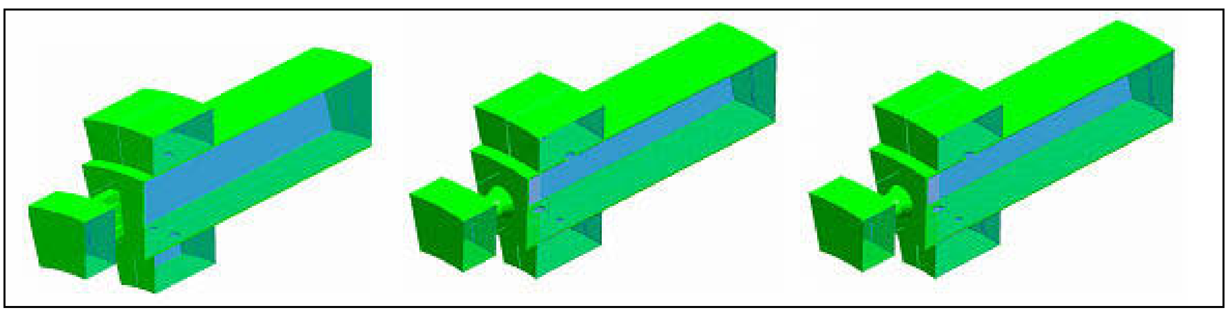

Fig. 9: Different cases.

Today's standard process of dealing with different assembly components involves quick, creative solutions to fix the model with little thought of reusability. Therefore, when the assembly is changed, one or more features would likely fail. In contrast, all of the nine processed models resulted in valid input files for ICEM without any human intervention.

Had this been done using standard practices, it would have required four to eight hours to modify and tag the nine models for ICEM instead of the 2 minutes per model required for the geometric processor.

4.1.4 Grid Generation

Using the scripting capabilities of ICEMCFD the grid generation step was automated. The script opened the CAD model and extracted the tagged geometry. Each input file was loaded into ICEMCFD. From the input file a grid was created and smoothed. A slice through the center of the automatically generated grid is shown in Fig. 10.

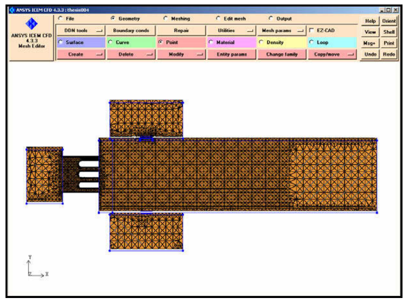

Fig. 10: The automatically generated grid.

To import the geometry, generate the grid, smooth it, and save it for inspection took around fifteen minutes. Standard practices would have required at least five to ten more minutes along with a human in the loop. Past experience has shown that it is necessary to check each grid to verify that the grid is error free of non-manifold elements, uncovered faces, low quality elements, etc. To develop the tools used in this thesis took around 400-500 hours. However, once they are developed, they can be applied to other studies with little additional development time. Tab. 1 shows a comparison between the current process and the one described in this paper. The new approach takes slightly more time in modeling set up for the first model than the current approach; however, it took ~ 40 % of the current approach time to generate nine models and review them.

	Current Approach		New Approach	
Steps	First Model	Next 8 Models	Single Model	Next 8 Models
Modeling	8-12 hrs	32-36 hrs	12-16 hrs	4-6 hrs
Grid Generation	8-12 hrs	32-36 hrs	8-12 hrs	3 hrs
Solution time	16 hrs	36* hrs	16 hrs	12** hrs
Report Results	4-8 hrs	32-56 hrs	4-8 hrs	3 hrs
Totals	36-48 hrs	132-164 hrs	40-52 hrs	22-24 hrs
Totals for nine solutions	168-212 hrs		62-76 hrs	
* Assumes that three solutions were run at a time due to the added time to create a model				
** All eight solutions were run together due to the decreased time to generate a model				

Tab. 1: Current approach vs. new approach.

4.2 Results: Design Review Tool

The intent of the DRT was to minimize solution documentation time. Currently, the standard practice in industry is to post process the CFD solution, generate images for a set of parameters, and then make a presentation. The presentation includes a discussion of the model, a review of the boundary conditions, and pressure field, and then a review of the results such as temperature, velocities, etc. If performed manually, it may take hours to document each case. When iterations of the model are performed, the previous presentation is used as a template and updated. This process of updating the presentation delays the review of the results until a formalized presentation can be made. The DRT reduced this time considerably. To determine the effectiveness of the DRT the following criteria were evaluated:

- Does the DRT present the results effectively?
- Are the results presented in a timely manner?
- Are results of each case setup to be archived for later use?

4.2.1 Does the DRT Present the Results Effectively?
The DRT provides a standard format for each case which was set up in a way that demonstrates the elements that are reviewed in industry today. This format consists of standard layout for each case. By using a standard format each case can be easily compared to the other cases in the study. This format can be changed as needed and quickly updated by modifying the DRT script and reprocessing the data.

4.2.2 Are the Results Presented in a Timely Manner?
When each case is completed, the DRT is executed. It takes around 2.5 seconds to create all of the web pages, organize the images, and create the graphs necessary to document each case. For cases with more results and images the time required to document the case would increase, but only slightly. When the DRT is finished running, each completed solution was ready for review. The standard practice used in industry today ranges from one hour to a day to document each solution. The time savings of the DRT imparts more time to review the results and minimizes the time to generate and format the results for review.

4.2.3 Are Results of each Case Setup to be Archived for Later Use?
The DRT uses a common directory structure. This directory structure places the results of the different cases in a common directory. The base directory has a folder for each case. Each case folder contains results as well as an images folder that contains all of the images. This enables the DRT to create web pages using a relative directory structure for each case. The relative directory structure allows the base directory to be archived and retrieved in a different directory while maintaining the functionality of the index page. To test out the archiving and retrieving capabilities of the tool, the base directory was compressed and moved to hypothetical archival location. After the study directory was retrieved, moved, and uncompressed, the index page still worked as designed when opened.

4.3 Results: Model Verification
The method described above significantly decrease the amount of time required to incorporate CFD into the preliminary design process. However, without an analytical verification, the ability to create more models is virtually useless. The models must be verified in order to be completely useful. Although the parts shown in this paper were not actual components, this method was conclusively verified on actual burners and other geometries. The models resulting from this method were inspected visually and were also compared to previous results achieved through already established methods, and the result was that the method was completely verified by experts in the field. Therefore, the new method is beneficial because it reduces the time required to create and analyze an accurate and useful model.

5. CONCLUSION
The objective of this paper was to describe a method for overcoming major challenges in the use of CFD in the preliminary design process. The method was applied to a simplified two stream mixing problem to demonstrate the numerous benefits of this new approach over the current approach to CFD. This was accomplished by creating valid CFD grids from parametric models in an automated manner. Creation of the parametric models required significant initial time investment, but the benefits of this method became clear further on. The geometry pre-processing and grid generation were completed using a variety of macro files. The incredible reduction in the time required for model preparation, analysis, and post processing allows time to be spent more efficiently exploring a wider variety of designs.

Perhaps the most influential facet of the approach is the DRT. This tool's modular structure allowed it to format each case for review in less than 10 seconds and to be modified rapidly. The ability to archive the results also was an integral part of the DRT.

The method and tools presented in this paper were formally verified using three different front-end and liner concepts for a two stream mixing problem. This verification demonstrated the tremendous potential of this method and tools to reduce the time required for CFD preliminary design analysis. It was found that by using the method outlined in this thesis that parametric models with predetermined topology changes as well as component replacement could be used to create valid grid in under an hour, where it currently takes anywhere from two hours to a week in industry. The general approach presented here will allow the aerospace, automotive, and other industries to evaluate more concepts before one concept is selected for the detailed design phase. It will also help them look at different features in the design space once the detailed design phase has begun. This work provides a starting point for future work in coupling CAD tools more closely with CAE analysis tools.

6. REFERENCES

[1] Ulrich, Karl.; Epinger, Steven: Product Design and Development, McGraw-Hill, 2000.

[2] Haimes, R.; Crawford, C.: Unified Geometry Access for Analysis and Design, Proceedings, 12th International Meshing Roundtable, Santa Fe, New Mexico, Sept 14-17, 2003, 21-31.

[3] Alonso, Juan J.; Martins, Joaquim R. R. A.; Reuther, James J.; Haimes, R.; Crawford, Curran A.: High Fidelity Aero-Structural Design Using a Parametric CAD-Based Model, AIAA 2003-4329.

[4] Hardee, E.; Chang, Kuang-Hua; Tu, Jian; Choi, Kyung K.; Grindeanu, I.: A CAD-Based design parameterization for shape optimization of elastic solids, Advances in Engineering Software, 30, 1999, 185-199.

[5] Crocker, D. Scott; Nickolaus, Dan; Smith, Clifford E.: CFD Modeling of a Gas Turbine Combustor from Compressor Exit to Turbine Inlet, International Gas Turbine & Aeroengine Congress & Exhibition, Stockholm, Sweden, June 2-5, 1998.

[6] Samareh, J. A.: Status and Future of Geometry Modeling and Grid Generation for Design and Optimization, Journal of Aircraft, 36, 1999, 97-104.

[7] Smith, R. J.; Leshchziner, M. A.: Automatic grid generation for complex geometries, Aeronautical Journal, 100 (991), 1996, 7-14.