2015-10-01

Using STAR-CCM+ to Evaluate Multi-User Collaboration in CFD

Kasey Johnson Webster
Brigham Young University

Using STAR-CCM+ to Evaluate Multi-User Collaboration in CFD

Kasey Johnson Webster

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Steven E. Gorrell, Chair
C. Greg Jensen
Chia Chi Teng

Department of Mechanical Engineering

Brigham Young University

October 2015

ABSTRACT

Using STAR-CCM+ to Evaluate Multi-User Collaboration in CFD

Kasey Johnson Webster
Department of Mechanical Engineering, BYU
Master of Science

The client-server architecture of STAR-CCM+ allows multiple users to collaborate on a simulation set-up. The effectiveness of collaboration with this architecture is tested and evaluated on five models. The testing of these models is a start to finish set-up of an entire simulation excluding computational time for generating mesh and solving the solution. The different models have distinct differences which test every operation that would be used in a general CFD simulation. These tests focus on reducing the time spent preparing the geometry to be meshed, including setting up for a conformal mesh between multiple regions in conjugate heat transfer models. Results from these five tests show a maximum speed up of 36%.

Keywords: computational fluid dynamics, collaboration, meshing, geometry preparation, integrated CAD and analysis, pre-processing, computer aided engineering

ACKNOWLEDGMENTS

This thesis would not have been possible without the help and support of many people. Thank you to Dr. Gorrell for helping, teaching, and motivating me over the last several years, and also to my committee members, Dr. Jensen and Dr. Teng. Thank you to Deryl Snyder, Aaron Bird, Cassandra Carpenter, Aaron Godfrey, Lisa Mesaros, Chris Penny, Prashanth Shankara, and Jeff Smith at CD-adapco. Thank you to CD-adapco and the Center for E-Design for funding. Thank you to Ryan Packer, Joe Becar, Mark Fernelius, Rory Stotts, Alex La, and Eric Wardell for help with multi-user testing, and Kurt Hinkle for their help on this thesis. Thank you to David French, Jonathan Sadler, and Scot Wilcox for help in developing the STAR-NX program. Thank you especially to my wife Kate and our families.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a	Speed of sound
API	Application Programmable Interface
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CDM	Common Data Model
CFD	Computational Fluid Dynamics
FEA	Finite Element Analysis
γ	Ratio of specific heats
M	Mach number
MUC	Multi-User CAE Tool for STAR-NX
p	Pressure
R	Universal gas constant
T	Temperature
v	Velocity

Subscripts, superscripts, and other indicators

$[\,]_t$	indicates total or stagnation measure or effective property
$[\,]_1$	indicates inlet value

CHAPTER 1. INTRODUCTION

An important tool for design analysis is Computational Fluid Dynamics (CFD) which allows analysis to take place simultaneously with design. CFD programs allow for the analysis of fluid flow, heat and mass transfer, and solid stress so that design changes can easily be made on computational models at an early point in the design process. Such changes are much less expensive than if a physical model was to be similarly built and analyzed.

Design and analysis is an iterative process. CFD programs improve this process because the cost savings of geometry updates and analysis are realized on each iteration, often resulting in more possible iterations and improved models. However, the use of CFD has its own costs, and this is heightened by the iterative design cycle. In each iteration that the current geometry is analyzed, necessary design changes are communicated back to the designer who then makes the change on the model. The model is then analyzed again as needed.

Collaborative design in engineering has been shown as a significant way to improve the design process, and has become increasingly important for innovation [1]. There are three reasons for this. First, time spent on a design can be reduced. This can often be advantageous even if the combined time of multiple users is greater than the time for one user, as in the case of an approaching deadline. Second, multiple users will have a greater amount of knowledge and experience than just one user. Third, multiple users are able to build on each other's work, catch errors, make suggestions, and as a result create better designs. The combination of these three things can be summed up simply as increased collaboration, and results in improved designs.

1.1 Collaborative Engineering Research at Brigham Young University

Collaborative engineering software has been developed at the Brigham Young University (BYU) site for the National Science Foundation Center for E-Design [2]. The benefits of multi-user collaboration in Computer-Aided Design (CAD) have been shown with the program NXConnect.

NXConnect allows multiple users to work together to create a model and reduce model creation time. Tests in NXConnect have shown model creation time of $\frac{1}{n}$ when n users work in collaboration [3].

1.2 Problem Statement

The benefits of collaboration shown at BYU with NXConnect and other CAD systems were anticipated to be shown in the set-up of a CFD model. Slotnick et al. [4] identified mesh generation as a major bottleneck in CFD due to the amount of work and time required to prepare the geometry of a model for meshing, and inadequate linkage from CAD programs to CFD programs. The file transfer between CAD programs and CFD programs requires the model geometry to be exported from the CAD program in a neutral format (.iges, .step, etc.) and then imported into the CFD program. The CAD representations from these file types do not produce surfaces in the CFD model that are suitable for mesh generation, often containing gaps between surfaces. Geometry preparation in CFD requires these gaps to be filled and often results in a de-featured model. The work required to prepare the geometry is time consuming, often taking longer then the model creation in CAD [5].

Because of the iterative design process where the geometry of a model is changed and reanalyzed multiple times before production, the costs associated with geometry preparation are amplified. Small design changes to the model can be made quickly in CAD programs, but the CFD simulation must be set up entirely for each iteration of the design due to the neutral file format transfer. This model set-up consists of the exporting and importing of the neutral format, preparing the geometry for meshing, defining properties and boundaries on the model for the solution, and creating reports and plots to be analyzed.

Similar to the benefits of collaboration shown in CAD, time could be saved in the geometry preparation and model set-up of CFD by allowing the collaboration of multiple-users and also through better integration of CAD programs and CFD programs. For the first time, the effectiveness and benefits of collaboration in CFD are evaluated in this thesis.

1.3 Evaluation of Collaborative CFD Using STAR-CCM+

The benefits of CFD collaboration were tested with STAR-CCM+ because its client-server architecture enables collaboration between multiple-users. STAR-CCM+ is a CFD program developed by CD-adapco. CD-adapco is a member of the Industry/University Cooperative Research group (i/UCRC) that funds the BYU site for the National Science Foundation Center for E-Design. CD-adapco has also created a program called STAR-NX, which creates a parametric connection between the CAD program NX, and STAR-CCM+. STAR-NX was used with NXConnect to test integrated CAD and CFD between multi-user programs. These abilities make STAR-CCM+ ideal for the research conducted for this thesis, and are explained more later in this paper.

1.4 Thesis Objectives

The purpose of this thesis is to demonstrate the benefits of collaboration in CFD and integration of multi-user CAD and CFD, and to show why commercial CFD software packages should be made to support multi-user collaboration. The metrics used to evaluate the effectiveness of collaboration in CFD were time savings, model improvement as result of collaboration, and the ability to catch and fix errors in the model. The overall impression of the users is also discussed.

These were evaluated by learning the multi-user capabilities, setting up multiple models with both a single user and two users working in collaboration, documenting the impressions of the users, documenting the time saving improvements over a single-user set-up, and documenting improvements that can be made to the collaborative capabilities of STAR-CCM+. STAR-CCM+ is the only commercial CFD program with capabilities for multi-user collaboration. Collaboration had not previously been formally tested on any CFD program, and the collaborative capabilities of STAR-CCM+ were unknown even to CD-adapco. The evaluation given in this thesis is the first time that collaboration in CFD has been formally tested with the documentation of results. The evaluation of these capabilities can be applied to the effectiveness of a CFD set-up using programs other than STAR-CCM+.

1.5 Outline

This research is separated into 7 chapters. Chapter 2 contains a review of previous literature relating to this thesis and a detailed explanation of geometry preparation that must occur in CFD. The literature presented is primarily related to the need for collaborative engineering, and engineering programs that have collaborative capabilities. Chapter 3 gives more background on STAR-CCM+ and the CFD model set-up, the client-server architecture, the operations that are required for a CFD simulation, and the collaborative capabilities of each of these operations. Chapter 4 describes the models that were used for the analysis of the effectiveness of collaboration, and how each model set-up is changed from a single-user to a multi-user approach. Chapter 5 explains the capabilities and limitations of STAR-NX, and also the method used to enable its use with multi-user programs. Chapter 6 discusses the results of the multi-user tests, and tests using STAR-NX. Chapter 7 discusses the conclusions of the research on collaborative CFD and the parallelization of multi-user CAD and CFD.

CHAPTER 2. BACKGROUND

Previous work has been done to show the importance of collaboration in engineering, and previous examples of collaborative programs. This chapter contains a review of the literature documenting previous work, describes the difficulties associated with geometry preparation in CFD, and explains the methods to repair errors in the geometry.

2.1 Tools and Benefits of Collaboration

Outside of engineering, software that both enables and promotes collaboration has become the norm. This includes Google Docs, conferencing programs such as Skype and WebEx, and cooperative video games. The prevalent use of these programs means that new engineers will be more capable of similar collaboration in the design and analysis process.

One benefit of collaboration is the ability to receive input from multiple users who are able to view the model as it is being worked on, or described in a presentation. Conferencing programs like WebEx allow a user's screen to be shared to other users anywhere around the world, and allows mouse control of the host computer to be given to any other user involved in that session.

Bidarra et al. [6] define concurrency and synchronization and how they relate to collaboration. "Concurrency involves management of different processes trying to simultaneously access and manipulate the same data. Synchronization involves propagating evolving data among users of a distributed application, in order to keep their data consistent." What this means is that effective collaboration programs must enable multiple users to work at the same time (concurrently) and that the document or model is kept current and consistent for each user (synchronized). Google Docs is a program that enables concurrent and synchronous collaboration, allowing users from any location to access and edit a file on the Google cloud.

Bercovitz et al. [1] say that the importance of collaboration is increased due to specialization of individual users, and the existence of interdisciplinary teams, the size of which have

increased each decade. The result of the existence of and reliance on teams is a desire and need for collaboration, but the necessary effective software tools that support collaboration do not exist. Google Docs and other examples of collaborative programs expose engineers to many forms of collaborative software outside of engineering. As these types of programs are becoming more common, engineers now entering the field are more capable of similar collaboration in the engineering process of design and analysis. This makes the need for collaborative engineering tools even greater as new engineers are able to fully take advantage of the benefits of collaboration in engineering applications. Having the correct tools to collaborate is the current missing link to effective engineering collaboration. This bottleneck to collaboration in Computer Aided Engineering (CAE) software is described by Red et al. [7]: "our modern operating systems and engineering applications are based on architectures designed for single users: one active application and one active cursor." For collaboration with CAE tools to be effective, synchronous and concurrent collaboration must be possible, and multiple users must be able to view, and make suggestions and changes to improve the model.

Collaborative CAE programs do exist, and have provided the motivation for exploration of a collaborative model set-up with CFD. Many of these programs have been developed at the BYU site of the Center for E-Design. These programs include multi-user versions of NX (NX-Connect), Catia (Catia-Connect), and Inventor (Inventor-Connect) for CAD, and CUBIT (CUBITConnect) for finite element analysis (FEA). These programs are described in chapters 2.2 and 2.3.

2.2 Multi-User CAD Tools

Collaboration during the design phase of the engineering process is a way for multiple users to create and edit a CAD model concurrently and synchronously. Multiple users edit a CAD model similar to the way multiple users can collaborate in Google Docs. In real time each user would be able to see changes made by the other users, and work together to create an engineering model.

BYU has developed multi-user versions of three CAD tools: NXConnect, Catia-Connect, and Inventor-Connect [2]. Each of these programs utilizes the Application Programmable Interface (API) that is revealed by the developers of the original CAD programs. The API allows geometry and feature information, and operations to be sent from all users to the server and back. These programs are thus limited by the amount of API that is revealed. Red et al. [2] explain that the lim-

ited API access keeps some operations that can be performed on the single user version of these CAD systems from being possible on the multi-user versions, and frequently requires extensive programming to obtain some of the model parameters that are required for other operations. Although this limits the capabilities of the multi-user CAD programs developed at BYU, the benefits of the multi-user approach are still shown.

Using a different approach, Bidarra et al. [6] at the Delft University of Technology created a collaborative design program named webSpiff. Rather than creating collaborative capabilities for a pre-existing commercial CAD system, webSpiff was created for Spiff, a modeling program developed at Delft. Because of this, the multi-user capabilities were able to be programmed directly into the program's source code. Although this did not require the same workarounds as the programs developed at BYU, the modeling functionality was limited to the functionality of Spiff.

There are three main benefits that can be anticipated from concurrent and synchronous collaboration. First, multiple users are able to see the model at the same time, and see the changes that other users are making. This means that errors in the model will more likely be caught and corrected, and the model will be improved. Second, the users are able to combine their strength and knowledge of different aspects of modeling. Because each user has different experiences, collaboration enables each user to work on aspects of the model that play to their strengths. Users can work on the aspects of the model that are in accordance with their strengths and create better models. Third, the models will be completed faster. Research at BYU has shown that a model being created by n users can take only $\frac{1}{n}$ of the modeling time that would be required for a single user [3]. The benefit of $\frac{1}{n}$ modeling time is that the total amount of man hours is the same for any number of users. In a situation where the modeling time is greater than $\frac{1}{n}$, there would still be the advantage of reduced development time. It would speed up the rest of the design process, getting the model more quickly to analysis, prototyping, and a final design. This reduces the down time of other engineers and would lead to reduced cost for development and a quicker gain in revenue from the design. This would especially benefit a modeling team working under a deadline.

2.3 Multi-User Analysis Tools

Computational analysis is another aspect of the engineering process that can benefit from the collaboration of multiple users. Computational analysis tools include those used for Finite

Element Analysis (FEA) and CFD. These programs often require preparation to the geometry as part of a simulation model set-up that is time intensive and can be a cause for errors in the model. These issues are described in this chapter.

In addition to the multi-user CAD systems developed at BYU, there has been work done on multi-user FEA using the program CUBIT [2]. CUBIT is an FEA pre-processing program developed at Sandia National Laboratories [7]. A benefit of CUBIT that lead to the development of CUBITConnect is the access that the multi-user developers at BYU had to the source code of CUBIT. This way API calls were not required to create the multi-user prototype, and this allowed more commands to have collaborative capabilities, and required fewer workarounds and less effort [7].

Synchronous collaboration between multiple engineers during the design and analysis of a Heating Ventilation Air-Conditioning system was demonstrated with a collaborative CFD program created by Borrmann et al. [8]. This program allowed the engineers to work on the same model from different locations, and see the updates from all other users as the model is updated. Parameters were passed to other users through buttons on a toolbar, and the clients were notified about events in the simulation [8]. No discussion was made about the results or benefits of this collaboration.

2.4 CFD Geometry Preparation

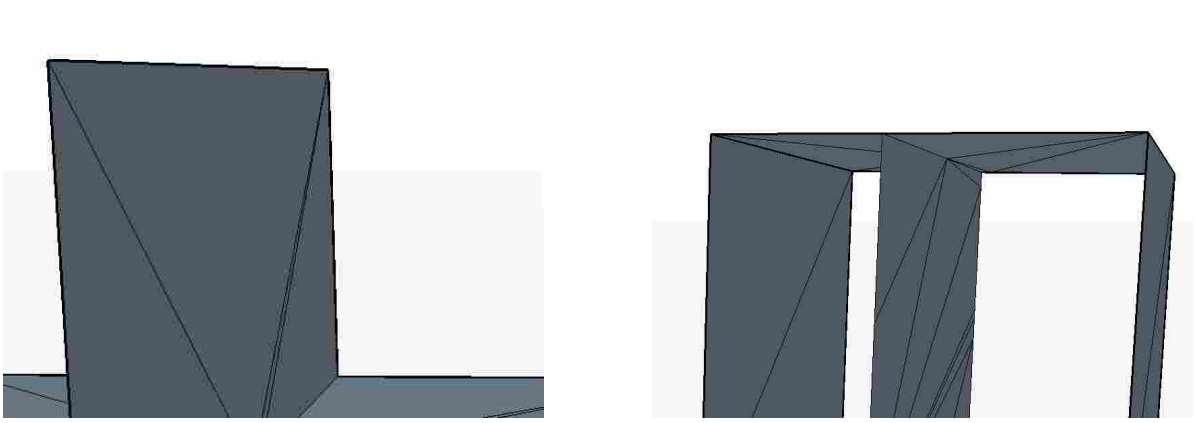
When CFD simulations are run on complex models, the simulation set-up often becomes more complicated due to extra steps that must be taken in order to ensure that the geometry is ready to be meshed and solved on [9]. It has been shown that the process of preparing the geometry of a model for CFD use can take longer than the initial creation of the CAD model [5]. The types of errors that typically must be fixed, and several methods of surface preparation are described in this section.

2.4.1 Closed and Manifold Volumes

Often the transfer from CAD software to CFD will create model volumes that are not closed or manifold. A closed volume is one with no holes, and is often referred to as “water tight” [4].

The volume mesh generation requires each region to be fully closed by the region's boundaries. A volume that is not closed does not represent a discrete region, and can not be meshed. The holes that exist on the region boundary must be closed by another surface.

A manifold volume is one where the edge of each surface in the model is formed between exactly two surfaces. A non-manifold volume is generally shown as a baffle face where the edge of one surface is not connected to any other surface or where three surfaces are connected to a single edge. This can happen when a surface intersects another closed and manifold volume. These two common occurrences of non-manifold volumes are shown here in figure 2.1.

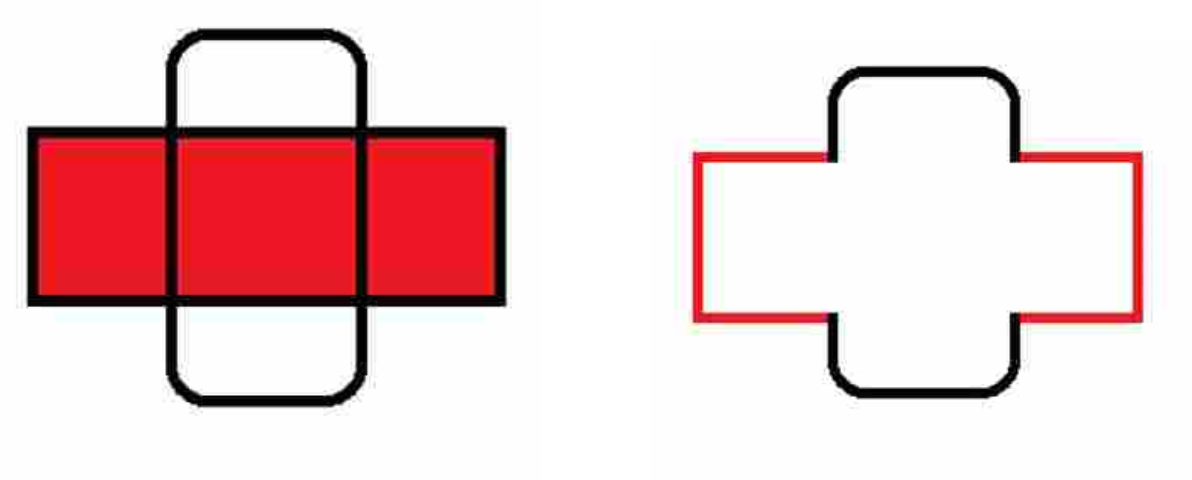


(a) A Baffle Face (b) Three surfaces connected to an edge

Figure 2.1: Cross-sections of non-manifold volumes

Volumes that are not closed or manifold will either be part of the same domain or separate domains. Each case has a different method of repair. When two volumes of the same domain or region intersect (figure 2.2), the surfaces of each volume that intersect the other must be deleted. When working with a shape that is imported from CAD this will generally be two solids that intersect as shown in figure 2.2a where the shape filled with red intersects the shape with no fill. The process of fixing this requires all surfaces that are shared or intersected by the two solids be deleted. The resultant single volume (shown in figure 2.2b) contains the outer surfaces of the original parts but none of the internal surfaces.

When two volumes that correspond to separate regions intersect (figure 2.3) the process is slightly more difficult. Even with multiple regions, a single area in space can only be occupied by



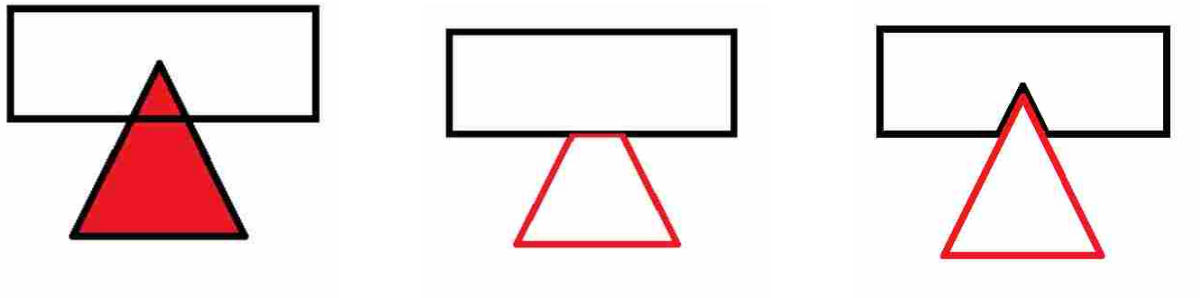
(a) Non-manifold volume for a single region (b) Repaired non-manifold regions

Figure 2.2: Fixing non-manifold surface of the same region

a single region. In the case of figure 2.3a the intersecting boundary of one of these regions must be removed. The issue that arises is that the specific surface that does not get deleted only belongs to the part it was originally on. This would result in a non-closed volume for the region with its boundaries deleted. In order to have closed and manifold surfaces, each region must contain the boundary where there is an interface between the regions. This means that each region must contain the same boundary. The geometry that is used in this model will determine which region will maintain its original geometry and which region will be altered (see figures 2.3b and 2.3c). The incomplete interface boundary must be duplicated and added to the other regions that contains the interface.

2.4.2 Preparing Geometry in STAR-CCM+

There are several tools for repairing surface errors in STAR-CCM+. These tools include Surface Repair mode, boolean operations on geometry parts, and the surface wrapper meshing model. Surface Repair mode and boolean operations are described in greater detail in the next chapter. Boolean operations and Surface Repair mode both deal with manipulating individual cells, edges, and vertices of the tessellated surfaces from the imported geometry. The user has control over which cells are changed, and how they are changed.



(a) The two regions (red and black) intersect, and the shared surfaces are non-manifold

(b) Intersecting boundary of the rectangular region is kept as boundary of both regions

(c) Intersecting boundary of the triangular region is kept as boundary of both regions

Figure 2.3: Fixing non-manifold surface of different regions

The surface wrapper creates a new surface that is able to make volumes closed and manifold. Rather than changing the existing imported geometry as is done with boolean operations and Surface Repair mode, the surface wrapper creates a new surface that overlays the existing surface, allowing the existing representation of the surface to still exist. The surface wrapper spreads from a specified point and creates a mesh on all surfaces or boundaries that can be reached from that point [10]. It will fill gaps in the surface that are smaller than the specified mesh size thereby closing the volume. It will also get rid of the intersection of any surfaces, leaving only the outside surfaces in a closed and manifold volume. For example, if spread from outside both surfaces of figure 2.2a, the result would be the encompassing boundary shown in figure 2.2b. If spread from inside the part of the triangle not included in the rectangle of figure 2.3a, the result would be the red trapezoid in figure 2.3b. In many cases of complex models the surface wrapper is a tool used to decrease the time spent repairing the surface to enable the volume mesh to be generated.

2.5 Multi-User Architectures

All collaborative programs that have been described in this paper utilize a client-server architecture that allows multiple users (clients) to connect to a single model through the server. Client-server architectures allow concurrency and synchronization to occur in these programs. Bidarra et al. [6] describe the two main types of architecture, and list the strengths and weaknesses of each. The two main types are thin client and fat (thick) client.

A thin client is used in conjunction with a thick server, and this architecture has operations and computations performed on the server. The benefit of a thin client architecture is that high performance is not necessary for each of the clients. In this case a high performance computer can be used for the server, and this will increase the performance on each client as well. The downside of this architecture is that information must be continuously sent between the clients and the server, requiring high network traffic.

A thick client is used with a thin server, where operations are done on each client. In this case nearly all capabilities of the program are performed on the machine of each user, and the server keeps the model consistent on each client. A benefit to a thick client architecture is that the experience of each user is not affected by the power of the server or the speed of any other clients. The downside of a thick client architecture is that "preventing data inconsistencies between different clients becomes a crucial problem" and each client has the same "computing power requirements of typical CAD stations" [6].

Each of the CAE programs described in this section use a thick client - thin server architecture. STAR-CCM+ uses a thin client - thick server architecture [10], and more about this program is described in chapter 3.

2.6 Integration of Design and Analysis

A difficulty that arises in CAE tools comes from the exchange of data between programs. After a model is created in CAD, that model data is then transferred to an analysis system via a file in a neutral format (.iges, .step, etc). This is due to the fact that geometry is represented in different ways for different CAE systems. In CAD, the geometry is represented by trimmed NURBS surfaces, and geometry in analysis programs is represented by tessellated surfaces [11]. There are often errors that result from the model transfer, such as non-closed volumes that must be repaired prior to meshing [4].

The engineering process consists of design of the CAD model using CAD specific software, transfer of the model, and FEA or CFD analysis on analysis specific software. This process is repeated as iterations of the model are improved based on the results of the analysis. This process results in better designs, but the problem lies in the fact that a different model is used for each stage of the design process, and for each iterative loop. This difficulty is increased because the CAD

model often contains "excessive detail not essential for analysis" which requires de-featuring of the CAD model during the analysis pre-processing [4]. This process of preparing the geometry for meshing, and the rest of the analysis set-up must be repeated each time for the updated model. This repeated simulation set-up becomes a bottleneck in the engineering process.

Integration of design and analysis has often been looked to as a way to reduce the time spent setting up analysis models. The commonly attempted method of integration is to improve the transfer method between programs that use different model representations. There are several examples of these attempts, and two are discussed here.

Gujarathi et al. [12] created a method that uses a "Common Data Model (CDM)" as a way to integrate CAD and FEA. In this process the CAD model is automatically converted to the tessellated surfaces of an FEA model. Parameters for both the CAD model and the analysis set-up are stored on the CDM, and updated automatically as the model changes.

Another method of integrating design and analysis models is through the development of added functionality to existing CAE tools. An example of this is NX Advanced Simulation, a program developed by Siemens to integrate analysis in the NX modeling program [13]. This program incorporates FEA pre-processing and post-processing into the same program with which the geometry model is created. This program manages files that keep the geometric part file separate from the idealized geometry, mesh properties, and solution properties, but still allows the geometry of all models to be updated from a change to the original geometry model. An added benefit to NX Advanced Simulation is the parallelization of the serial process that flows from model creation to de-featuring of the idealized model to mesh set-up to simulation set-up. With this parallel process, multiple users can easily collaborate, with each user synchronously working on a different stage of the process [14]. A team of three users was able to complete the model set-up in 55% of the time of a single user.

The research in this thesis presents another method of the integration of design and analysis. This method integrates CAD and CFD using STAR-NX and is described more in chapter 5.

CHAPTER 3. USING STAR-CCM+ TO EVALUATE COLLABORATIVE CFD

STAR-CCM+ is a CFD and Continuum Mechanics program created by CD-adapco. CAD and mesh geometry can be imported or created in the program to represent the domains that can be solved for fluid flow, heat transfer, and stress. Because of the ability to import three-dimensional models, STAR-CCM+ lends itself well to simulating complex geometries, and its client server architecture allows for multi-user collaboration. This chapter discusses the client-server architecture of STAR-CCM+ and contains an evaluation of its collaborative capabilities.

3.1 Client-Server Architecture

Collaboration is enabled in STAR-CCM+ by its client-server architecture (see figure 3.1). When a simulation is created or opened through the STAR-CCM+ workspace a client and server are created that correspond to that particular simulation. In the thin client - thick server architecture of STAR-CCM+ the client is where the user can set-up the simulation through the graphical user interface (GUI) or a batch script, and the server is where commands and operations are executed and then sent back to the client to be displayed. The main benefit of this type of architecture is that the data on each of the clients is kept up to date. Because the majority of the work is done on the server, high performance is not needed for the clients, and the speed of the simulation set-up is not determined by one slower client.

There is a major difference between the architecture of STAR-CCM+ and the CAE programs already described. The other programs have a centrally located server, and store each model used on that server. A user connected to the server is able to open, create, or join other users in a model on the server. The server exists on a computer dedicated specifically for the server. With STAR-CCM+ each simulation file is associated with its own server, and the server can either be created on a remote computer or on the computer where STAR-CCM+ was opened. The simulation files are stored on the computer or network drive of the user. When a simulation file is opened

or created, this creates a new server that exists for as long as a process remains running for that server which generally exists on the computer of this user (an exception being a simulation that is created or opened remotely on a different computer). For example, a new simulation is created by a single user (client 1), a new server for that simulation created with a specific port for the process. Other users can join this server if they know the server's host name and port number (clients 2-n). Any of these users, including the initial user, can then disconnect from the server while the server continues running. If other clients are connected to the server and the user of client 1 closes STAR-CCM+, the server will continue running, and the other clients will be able to continue working on the server. The server will exist until one of three things happens. First, the server will be closed if all clients close STAR-CCM+ and the last remaining client selects to close the simulation (alternatively the last client can select to disconnect and leave the server running). Second, the server will close if the computer containing the server is shut down. Third, the server can be closed by terminating the process in the task manager. This can only be done on the computer containing the process.

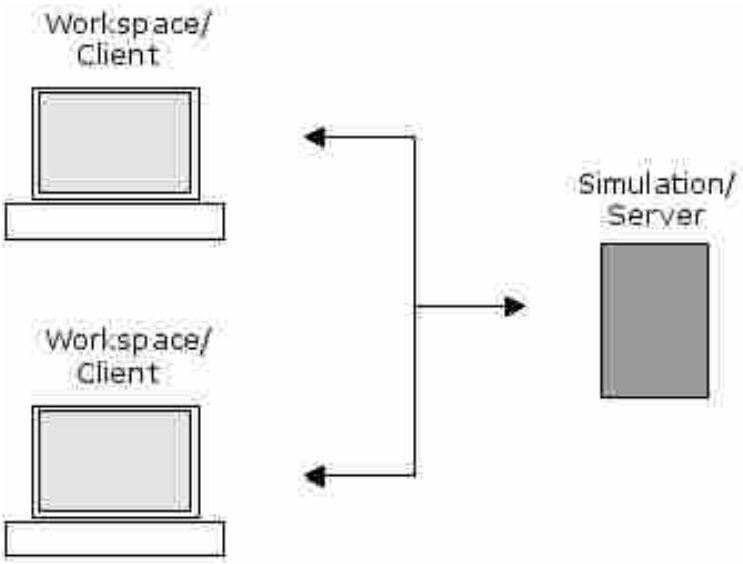


Figure 3.1: Client-Server Architecture

CD-adapco gives three primary benefits of this architecture. First, the client can be disconnected from the server and closed to free up machine resources while the server continues running in the background. This same idea becomes useful if the STAR-CCM+ client crashes. In this sit-

uation the server will keep running and can be reconnected to by a new client without losing any work. Second, the client server architecture allows remote connections between a client and server. This helps speed up the simulation when the server is run on a computer or cluster with high computing speed, and the client on a machine with high graphics speed. Third, the architecture enables collaboration between multiple users. Additional users can connect different clients to the existing server and interact with the server in the same way as the original client [10]. Surprisingly, despite this third point being documented, there has been little if any testing of the multi-user capabilities or resultant time savings in STAR-CCM+. Due to the effectiveness shown by multi-user CAD [7], the ability to set up complex CFD simulations is anticipated to be an area where significant time savings can be achieved in the engineering design process.

3.2 STAR-CCM+ Operations & Capabilities

The capabilities of multiple clients working collaboratively in STAR-CCM+ were discovered as multiple simulations were completed by two users. Testing was performed on STAR-CCM+ versions 8 through 10. These simulations are described in greater detail in chapter 4. The simulations were selected as a way to test every operations of a STAR-CCM+ simulation for collaborative capabilities. Two main things where examined while evaluating the collaborative capabilities: that each operation could be performed synchronously with other operations, and that each operation could be performed synchronously with the same operation being performed by a different user. The operations were tested using multiple clients controlled by a single user, multiple clients running java macros at the same time, and with two clients connected to a server, each client controlled by one user. Throughout the simulation set-up of each model used in this research, the capabilities were documented, and the time savings measured.

In addition to documenting the collaborative capabilities of each operation in STAR-CCM+, it is also be necessary to explain the roles played by each of these operations in a simulation set-up. The descriptions of these operation will be best understood by an experienced user of STAR-CCM+ who is familiar with the operations, but can also help a user with experience in a different CFD program to relate similar operations of different names. The operations are classified according to the section of the simulation set-up in which each takes place, and each operation and its collaborative capabilities are described. The tables in this chapter document if the same operation can

be done simultaneously on multiple clients. Any other necessary explanation of the collaborative capability is given as the operation is described.

3.2.1 Geometry

The initial preparation for the model is done on the imported geometry, and the collaborative capabilities are shown in table 3.1.

Table 3.1: Multi-User capabilities of geometry operations

Operation	Yes	No
Combine Parts	X	
Combine Surfaces	X	
Surface Repair		X
Splitting Surfaces	X	
3D-CAD Models	X	
Create Part	X	
Composite Parts	X	
Boolean Operations	X	
Retessellate Parts	X	
Apply Tags/Filters	X	

Combining parts is an operation that brings the leaf level surfaces and curves of separate bodies into one part. Being able to select the specific surfaces for a part is important when selecting a part on which to do surface repair and in creating a region from that specific part.

Combining surfaces is done to exactly specify a specific boundary of a part. Multiple surfaces within a single part can be combined into one. If the surfaces to be combined are in different parts the parts must be combined first, and all surfaces that are desired for a separate part can be selected and a new part created.

Surface Repair mode is the only geometry based operation that is unable to be executed by multiple users. The operation has the capability of looking at entire surfaces, edges, and individual

cells and manipulating them to create a closed and manifold volume. It is in this mode that edges can be merged together, vertices can be created on intersections, and individual cells deleted among other operations important to creating a sealed volume to be meshed. If a user attempts to use the surface repair mode while it is in use by another user a message is displayed: "Warning: Already editing surface, shutting down repair session."

In addition to shutting down the surface repair session of the original user, there will also often be errors in the scene of that user.

Splitting parts and surfaces can be done in many different ways. Surfaces can be split into separate non-contiguous surfaces, by part curves, by angle, and by patch. Non-contiguous surfaces are surfaces that are not connected to the ones around them. Patches are inherent to the imported geometry of the surface, and when splitting a surface by patch the user can select multiple patches to constitute a new surface by renaming those selected patches. The non-selected patches remain as the default surface. Parts can be split into separate non-contiguous parts, or by surface topology where a part is separated into separate closed and manifold volumes that it contains.

Surfaces can be split simultaneously by multiple users, which may lead to confusion between users. When a surface is split both resultant surfaces retain the same name but one will be followed with a number. If multiple users simultaneously select the same part to split, the part will never actually be split at the same time. The first user to execute the split will split the surface as specified, but the next user will only split the resultant surface of the same name from the original split.

Splitting surfaces by patch is done in the GUI with a surface splitting scene where the geometric patches can be selected to be made into a new surface. Multiple users can enter into this mode and split surfaces simultaneously, but the scene and list of surfaces will not be updated for the other users when one is split. This results in the ability for another user to select a surface that has already been split and do the same operation over. In this case the operations are not overwritten, but duplicated. There will be multiple surfaces that are identical, each with the name given to them by the user who executed the operation.

The 3D-CAD Modeling mode is the platform in STAR-CCM+ to create CAD parts. In this mode CAD files can be created or imported from other CAD software and edited. In this mode multiple-users can collaborate but there are restrictions to its collaborative capabilities.

The restrictions of the 3D-CAD Modeling mode are that it has only one associated scene and no new scenes can be opened even when a new model is created by a separate user. If multiple users are building the same CAD model, sketches made simultaneously can only be made on the same plane, and any rotation of the view will be done on all clients using the 3D-CAD Modeling mode. This also causes the last body or surface selected by a user to be highlighted in the scene of all users.

Features such as extrudes and revolves can be done and edited simultaneously, but once a feature has been performed it can only be edited in a certain order. When a user edits a feature, another user can only edit features that were executed after.

New parts can be added to the simulation using the create part option. This operation includes the creation of a block, cone, cylinder, or sphere. These shapes are created with a user selected coordinate system and the properties of each shape.

A composite part is created as a base level organization of separate parts. The parts both exist separately but are stored within the composite part on the geometry tree.

Boolean operations can be done on two or more parts. Each part must be closed and manifold before the operation is done. The operations include uniting, intersecting, subtracting, and imprinting parts. Uniting joins the individual parts into a single part. Uniting is the process that removes the surfaces where two parts intersect, leaving only the outer surface of the part. Intersecting creates a part that consists of only the surfaces and volume that all selected parts share. Subtracting will remove the volume of one part from a selected target part. Each of these three operations will create a new part on the tree that is a result of the operation, and leave the selected parts unchanged. Imprinting parts is different because it edits the surfaces in the parts that are selected without creating a new part. Coincident and nearby surfaces are merged into a single surface that is shared by each part. The tolerance for surface proximity to be merged is selected by the user.

Boolean operations can be used as a way to have multiple users do surface repair. One user is able to work in surface repair while the other can do similar operations on another part using Boolean operations. Boolean operations require the parts included be closed and manifold.

Parts can be retessellated only if they are imported CAD parts. This operation changes the mesh density between very coarse, coarse, medium, fine, and very fine or the user can select tolerances for the tessellation size.

Tags are a way of organizing parts and surface that are similar. Filters can be used to limit the options of which parts can be selected, by selecting the filter of the desired tag. When parts or surfaces need to be selected from the three, the filter can be applied to only show parts that have been tagged and filtered.

3.2.2 Mesh

The geometry of the model surface is further discretized into specific cells, the size and shape of which can be specified by the user. These cells are meant to be a more uniform shape and size than the tessellated surfaces of the imported geometry. The surface mesh cells are then used to create a mesh in the volumes bound by these surfaces. The mesh represents finite volumes on which the CFD properties of the models are solved. The mesh operations are described in this section and the capabilities are shown in table 3.2.

Table 3.2: Multi-User capabilities of mesh operations

Operation	Yes	No
Create Mesh Continua	X	
Generate Surface Mesh		X
Generate Volume Mesh		X
Contact Prevention	X	

Mesh continua specify the meshing models that will be used, and attributes of that mesh. Multiple mesh continua can be created, and applied to specific regions as needed. Mesh values are assigned in the properties of the mesh continua and can be customized to mesh different sizes within a defined volume or near curves.

Once the mesh properties are specified, the mesh is generated on those regions. The surface mesh is generated first, as cells lying on the surface of the region, and then the volume mesh is created in the volume enclosed by the regions.

No operations can be executed while a mesh is being generated, either in parts or region based mode. The only thing that can be done during meshing is editing the properties of operations that have already been performed.

Contact prevention is used to prevent surfaces in close proximity from being merged while using the surface wrapping tool. This merging occurs when the surfaces are closer together than the base mesh size of the surface wrapper. Contact prevention decreases the mesh size to a user defined minimum in areas where the specified surfaces are in close proximity in order to keep them separate without wasting computational resources creating a finer mesh in the other areas where it is not needed. Setting up and specifying contact prevention can be done simultaneously with other operations.

3.2.3 Regions

Regions are the specified domains on which the final mesh is created and the simulation is solved. The continua cannot be solved on geometry parts. It is in the regions where boundaries are created and defined from part surfaces, and also where the boundary conditions are set. All operations in the regions tab are compatible with multi-user collaboration (see table 3.3).

When assigning parts to regions the user can choose to have each part as its own region or create a region containing all parts. Defining boundaries is also done while assigning parts to regions. Boundaries can be created for each surface, one for all surfaces, or one for each part.

Physics continua are created to set initial conditions and physics models for domains of the simulation. The initial conditions are created to apply to entire volumes rather than individual boundaries and then applied to the regions with the corresponding initial conditions. Each region can only have one set of initial conditions, but the initial conditions can be edited for specific boundaries in the regions tab.

Each continuum for both mesh and physics can be applied to one or more regions. This is selected in the properties window of that region.

Table 3.3: Multi-User capabilities of region based operations

Operation	Yes	No
Assign Parts to Regions	X	
Defining Boundaries	X	
Create Physics Continua	X	
Specify Applied Continua	X	
Splitting Feature Curves	X	
Create Interface	X	
Define Initial Conditions	X	
Define Boundary Conditions	X	
Setting Boundary Types	X	
Edit Boundary Mesh Values	X	
Edit Boundary Physics Values	X	

Feature curves can also be created when the parts are assigned to regions. These curves can be created in order to split the boundaries of the regions and to specify mesh values near the feature curve.

Interfaces are created when there is a need for a conformal mesh where two regions meet. A conformal mesh is where the meshes of two regions line up exactly where the regions come together. A conformal mesh is important for conjugate heat transfer problems because the aligned mesh allows for accurate heat transfer between regions. The creation of the interface is applied to two surfaces that are identical in separate regions. The interface must be created in order for the cells along the surface of the shared boundary of two regions to be lined up.

Initial conditions are defined within the physics continua. They specify the initial properties of all nodes on the volume to which they are applied.

Boundary conditions define a unique solution to a simulation, and include properties at the boundary such as velocity, temperature, and pressure. The boundary conditions are applied to each boundary where needed.

Boundary type defines what exists at that boundary. Examples are a wall, inlet, outlet, free stream, axis, or plane of symmetry.

The mesh values can be specified for each boundary in the mesh properties of that boundary. This enables mesh size to be adjusted in areas where needed. All boundaries that do not have mesh values specified will use the values that were specified in the mesh condition.

Similarly the physics values can also be customized for specific boundaries. All other boundaries not specified will be the default of the physics continua.

3.2.4 Post Processing

Post processing is where the solution is analyzed. This analysis can be set up anytime there are regions created, either before the solver is run or after. Properties of the analysis can also be edited while the solver is running, similar to mesh generation. As shown in table 3.4 all post-processing operations can be performed in collaboration.

Table 3.4: Multi-User capabilities of post-processing operations

Operation	Yes	No
Create Scenes	X	
Generate Reports, Monitors, and Plots	X	
Visualize Report Solutions in Scenes	X	
Select Convergence Criteria	X	
Create Derived Parts	X	

Scenes can show the scalar and vector properties of solid and fluid flow domains. Results of the solution can be viewed and analyzed using scenes.

Reports are specified for the calculated solution values that the user wants to know. These can be monitored and plotted throughout the time the simulation is solving. The plots can be added to any scene for a visualization of combined items of interest.

Convergence criteria determine the length that a simulation should run and can determine when the solver has arrived at an answer. The convergence criteria are set up prior to solving.

Derived parts are created to help in viewing simulation results. Derived parts can be used to visualize the cross section of a part, and view streamlines. Derived parts are created as being part of the geometric parts or regions, but do not affect the meshing or solving of the simulation.

CHAPTER 4. TESTING

With the multi-user capabilities of the client-server architecture known, tests were performed with two users to see the time savings and other collaborative benefits that could result. Tests were originally performed on three STAR-CCM+ tutorials: Control Valve Meshing, Multi-Region Heat Exchanger, and Transonic Flow: RAE2822 Airfoil. In addition to these tutorials tests were performed on three more complex conjugate heat transfer models. Two of these models were provided by CD-adapco. The first was a rotating turbine blade cooled from the inside. The turbine blade contains a solid region and two fluid regions. The second was a helicopter exhaust, where the surrounding moving air cools the exhaust and exhaust housing material. The last model was a turbine vane provided by the US Air Force Research Laboratory (AFRL), with internal film cooling and external blade flow, containing many cooling passages and holes. This turbine is more complex than the turbine blade from CD-adapco and contains a single fluid region. Each model is described in greater detail as the set-up steps are presented in this chapter.

4.1 Model Testing Process

In order to reduce the effect that the familiarity of the users with the mode had on time reduction, single-user and multi-user set-ups were alternated for each model (one single-user set-up for each user, then one multi-user set-up, then single-user, etc.). In this way the multi-user tests can be looked at as more than just the natural improvement based only on experience with setting up the simulation multiple times. In the case of models that required surface repair, user 1 used Surface Repair mode due to having more experience. The tasks performed by each user in all simulations were chosen to be tasks that built upon themselves and not on tasks of the other user. Each task other than surface repair was estimated to take the same amount of time to be performed by each user, therefor the plan was only to organize and assign the tasks based on a logical progression of the set-up, not on user ability. This plan was developed by a single user

prior to any of the timed testing. The plan helped identify what surfaces needed to be repaired, what tolerances were required for repair, and also where the mesh would need to be refined. The values for boundary and initial conditions of the models were taken from documentation included with the models. On the tutorial models these were included in the step-by-step instructions, and on the other models the information was provided from previous simulations of the model. Although the inclusion of model set-up planning and further refinement of the mesh and other aspects of the model set-up after mesh generation would affect the results shown in this thesis, and give a more realistic impression of how a CFD set-up is performed, this could only be performed if each individual set-up was performed with a different user, and each collaborative set-up was completed by different users who did not participate in the individual set-ups. This was not possible due to the limited number of users who knew STAR-CCM+ well, or were able to put in the time to learn. If planning had been included in the individual set-ups, then the results of all subsequent tests would have been skewed in favor of increased time reduction for a collaborative set-up. As a way to decrease the effects of familiarity with the model from the results, the single-user and multi-user set-ups were alternated, and did not include planning time.

As is expected with complex CAD geometry, as these models became more complex the need to repair errors on the surface became greater, and limitations and pitfalls in the collaboration capabilities became more apparent. What also became very apparent was the importance of communication. Red et al. [7] state that communication is the aspect most important for effective team collaboration in industry. Tests were done following a very specific plan designed to reduce the downtime that a user would experience while waiting for other operations to be finished by the other user, and there were also simulations in which only a very basic plan was followed but the users communicated what was being done and what the next step would be. The results of these timed tests show that even a very detailed plan does not eliminate the bottlenecks that will stall the work of one user because the time required to do specific tasks is only and estimate when forming the plan. With more complex models, these bottlenecks occurred more frequently, in spite of having a well thought out plan. In such cases, deviating from the plan resulted in less downtime for an individual user, and a more significant reduction in set-up time. An important aspect of these cases is that the users communicated what was being done when the plan was changed in order to maintain a constant work flow and avoid confusion. Communication was particularly impor-

tant when repairing surfaces, which was necessary to create closed and manifold parts. The users communicated mainly to avoid opening Surface Repair mode at the same time, which would cause future errors in the model (as described in chapter 3). Not only did this help to avoid confusion and errors, but it also naturally changed the planned order of the set-up in a way that nearly eliminated the downtime of any user.

For each test the methods used to repair the surface errors were Surface Repair mode and boolean operations. The surface wrapper was not used for these models because it cannot be used in collaboration. The benefit of collaboration for surface preparation can best be seen by having more user interaction with the model, and this was accomplished using both Surface Repair mode and boolean operations.

In order to simulate collaboration among users in different geographic locations as frequently occurs in engineering, all communication was performed via Google Hangouts. Although the two users sat in the same room during each of these tests, at no point did the users look at each others screens or communicate face to face. Both the plans for a single-user set-up and for how to partition the model to a multi-user set-up with little downtime were written down and given to the other user. Similarly, at no point during the single-user set-up did the user receive corrections about the model set-up. This allowed the users to review each others work by using the scenes and tools existing in STAR-CCM+, and required the ability to communicate aspects of the simulation set-up effectively with fewer words and visualizations.

The testing on these models explores and shows the effectiveness of synchronous collaboration in a CFD model set-up, but does not include the computational generation of the mesh or solver. This focuses on the time that users spend interactively setting up the model, and not on computation time. The geometry models tested and shown here have been previously validated. Thus the testing described in this chapter does not include a grid or time dependence study for these models and instead focuses on model set-up, which is defined here as only including the specific CFD operation used in the following general steps:

- Opening the model in the CFD system
- Preparing the geometry to be suitable for meshing
- Defining mesh type, size, and boundaries

- Specifying boundary surfaces, types, and physics values
- Defining initial conditions for the regions
- Creating relevant reports, monitors, and post-processing visualizations

The steps required for model set-up were defined prior to testing, either within the tutorials or decisions made based on user experience on how to achieve the pre-defined output of that model. All set-ups for a given model were completed using the same pre-defined steps. This eliminated uncertainty in the results that would be caused by the need to refine mesh in areas, or update any parameter in the model after the mesh had been generated or the solver run. These situations would likely occur in industry, and would increase the time spent in model set-up. However, these tests were designed to demonstrate how time could be reduced when the necessary model set-up is done correctly the first time.

4.2 Control Valve Meshing

The Control Valve model is a tutorial used to show ways of generating different types of meshes. For the tests performed on this model, only one type of mesh was generated and physics conditions were created and applied to the model in order to test a full model set-up. The simulation set-up consisted of splitting and renaming surfaces from the imported geometry and extracting feature curves in order to define the boundaries of the geometry, creating a mesh and physics continua, assigning the parts to regions, defining boundaries and types in the regions, specifying mesh size on specific boundaries, and creating scenes to visualize the flow velocity inside the valve. A report and scene were created to show the velocity of fluid on the symmetric plane of the model (figure 4.1). User 1 worked with the geometry to create surfaces and assign the parts to regions while user 2 created the mesh and physics continua. When the regions were created user 1 began setting up the scenes and user 2 assigned the boundary values and specified mesh size.

This model required a very serial process of splitting the geometry into curves and surfaces where each step depended on the previous; therefore it was difficult for multiple users to work on the geometry collaboratively. The plan followed in this set-up meant that user 2 could only perform

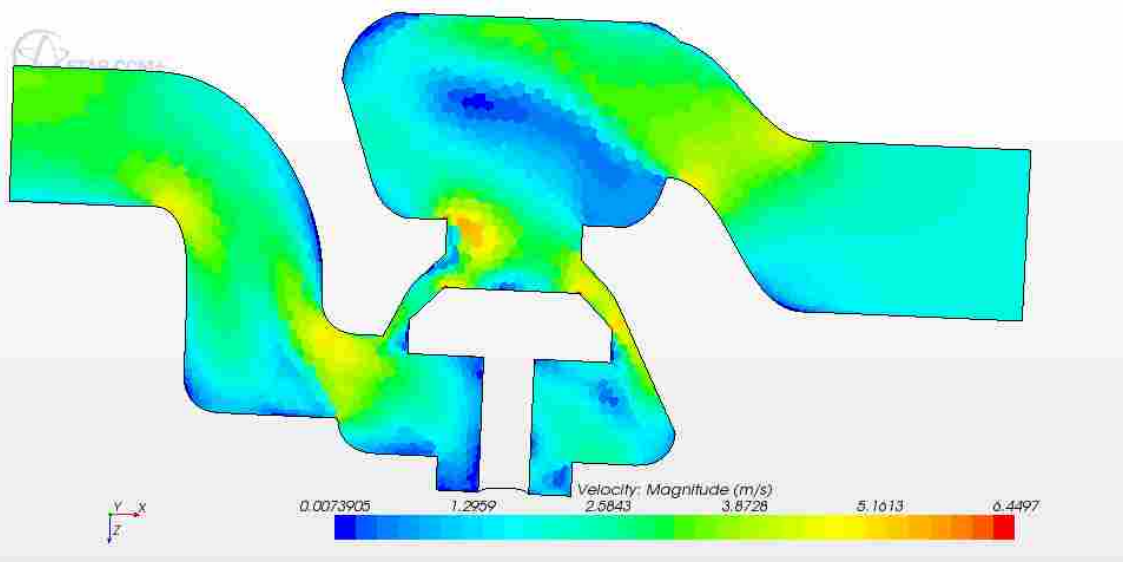


Figure 4.1: Control Valve model scene showing the scalar velocity profile inside the valve

a limited number of operations until user 1 finished the work of geometry preparation. Once the physics continua had been created, user 2 had to wait for the regions to be created, and could not apply the boundary conditions until then. On a simple model like this the difficulty of planning the work flow was low, and the time spent waiting for the other user was also low, but it can be seen how both would increase on a more complex model that does not partition intuitively. The set-ups of this model showed that although the operations used for this model set-up all worked in collaboration, care must be taken to ensure that bottlenecks are avoided.

4.3 Multi-Region Heat Exchanger

The Heat Exchanger model is also a tutorial to demonstrate mesh generation. The model is a conjugate heat transfer model so the created mesh must be conformal between the fluid and solid domains. The conformal mesh is lined up on the boundary between two regions and improves the accuracy of heat transfer between regions. Because the tutorial was created to only show mesh generation, once again physics conditions were added to the model in order to complete a full simulation set-up. The model is directly imported as regions because there is no need for the surface to be repaired. With two regions imported, the two users worked collaboratively by each focusing on a single region. The boundaries on each region were split and types and names were

assigned to the newly created boundaries. In this case user 1 defined the boundaries of the fluid region and user 2 defined the boundaries of the solid region. User 1 then specified the meshing models and properties for the entire model. At this time user 2 created in-place type interfaces between the solid and fluid boundaries and enabled prism layers to be grown on the interfaces. User 1 then specified the fluid physics models and set boundary and initial conditions for the fluid domain and user 2 did the same for the solid domain. At this point the surface and volume meshes were generated for the model. With a mesh model, each user was able to set up reports for the region in which they were working. User 1 created a report on fluid velocity and created a scene showing velocity vectors, and user 2 reported and created a scene showing pressure on the heat exchanger walls and temperature of the solid (see figure 4.2).

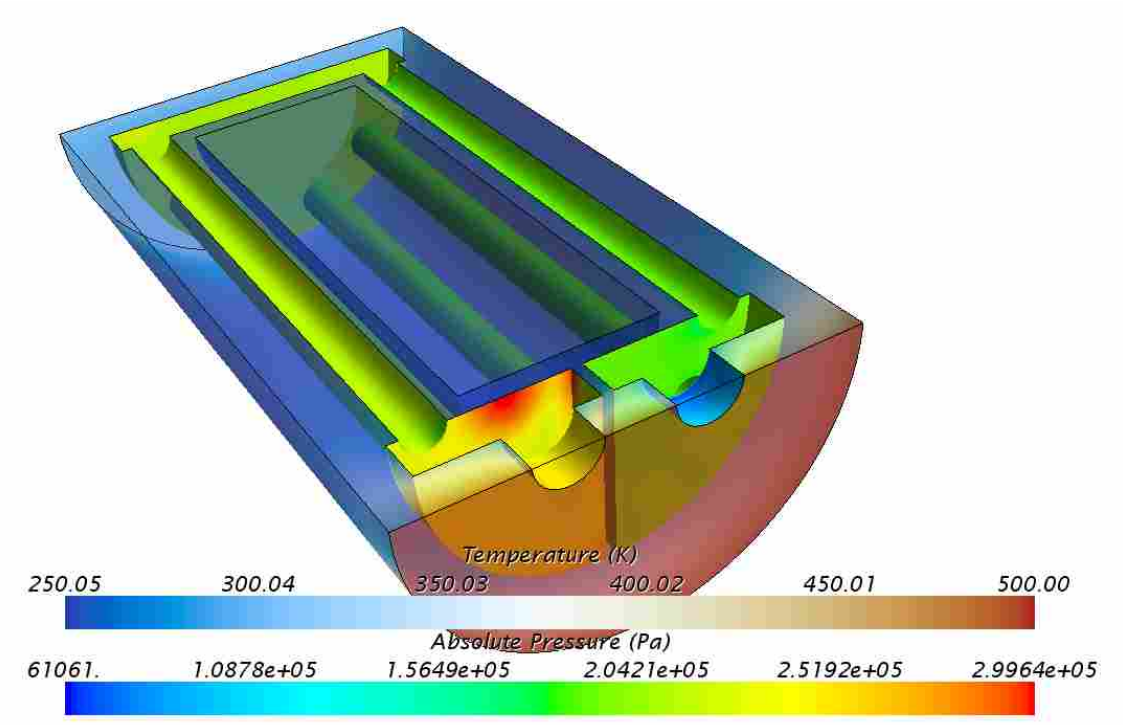


Figure 4.2: Multi-Region Heat Exchanger model. This scene shows the boundary temperature and absolute pressure on the heat exchanger walls

A very helpful aspect to collaboration in this simulation involved the two separate regions. By having distinct regions that each user worked in, there was no need to plan around the operations of each user. The users were able to set up the continua and apply the physics values to the model

without waiting for the other user at any time. The set-up was intuitive for each user, and only required communication of when each user had finished the set-up needed prior to meshing and prior to solving.

4.4 Transonic Flow: RAE2822 Airfoil

The airfoil simulation involves compressible flow over a 2-Dimensional airfoil. This simulation begins with an already meshed surface file that is converted to 2D. Because the mesh is imported for the simulation, the set-up process does not involve any mesh set-up. While working in collaboration, user 1 worked on the physics of the model while user 2 set up the solving parameters and post processing, developing a report and scalar scene showing the Mach number of the fluid around the airfoil (figure 4.3). User 1 selected the physics models and applied the initial and boundary conditions. User 2 set up a new coordinate system in order to specify the airfoil's angle of attack and the initial and boundary conditions were set to this coordinate system, and then increased the Courant number to speed up convergence. Both users then worked on specifying boundary conditions and visualizing the solution. A scalar scene was used to show the Mach number of air around the airfoil (figure 4.3) and a report, monitor, and plot were created to show the lift and drag coefficient.

The ease of this model had much to do with the fact that the mesh had already been created. After this point there are not many operations in the set-up that need to be done in a certain order. The operations of both users were independent and although a plan had been made to which operation would be done by whom, the work load could have been partitioned in many different ways. However, because there was only one region with no meshing or mesh repair required, this model does not represent a model that would be used in industry, or would be most improved by multi-user collaboration.

4.5 Cooled Turbine

The Cooled Turbine is a training model used by CD-adapco that models conjugate heat transfer in a rotating reference frame. Hot air travels through the main engine passage over the turbine blade as it rotates, and cooling air travels through an internal cooling passage of the blade

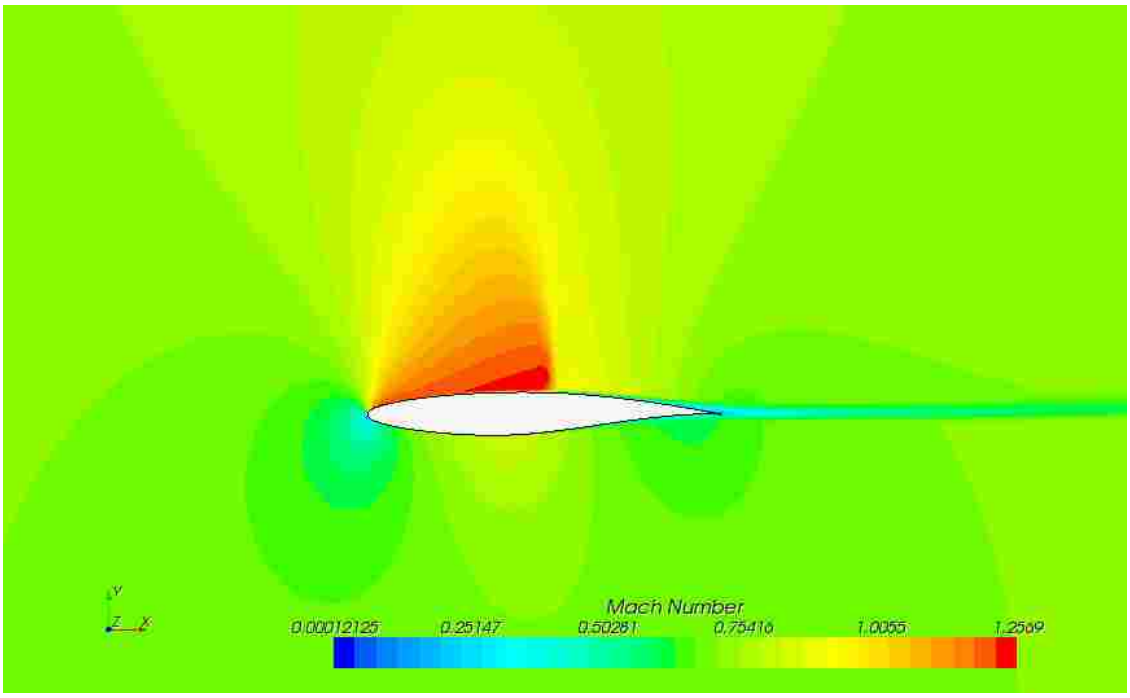


Figure 4.3: Transonic Flow: RAE2822 Airfoil model scene showing the Mach number around a cross section of the airfoil.

to cool the solid. Three regions are used to represent the main passage flow, internal cooling flow, and solid blade.

This model is unique to the other models in this paper because it uses the 3D-CAD Modeling mode of STAR-CCM+. This enabled the testing of the operations used in this mode, and also the time savings that resulted from geometry preparation performed mainly in this mode.

User 1 first imported the model from a parasolid file into the 3D-CAD Modeling mode where the volumes for each separate region were extracted. The model of the blade only was imported, and a box was sketched and revolved to match where the boundary of the hot flow exists for each blade section. Boolean operations were used to define the domains. A subtraction separated the blade from the hot flow region, and a volume was extracted from the already existing inlet and outlet surfaces of the cooling flow in order to define that domain. After these steps 3D-CAD Modeling mode was closed, and geometry parts were created from the CAD model. Because the model domains were defined in 3D-CAD Modeling mode, and the imported geometry was set-up for CFD use, the volumes were made closed and manifold without the use of Surface Repair mode. While user 1 worked in 3D-CAD Modeling mode, user 2 defined the mesh and physics

models and set initial conditions. Once the parts and then regions were created from the 3D model, the users worked together to define boundary types and conditions, and create the reports and a scene before solving. The scene was created to show a turbine blade cut through the middle, displaying temperature on the blade solid, and pressure of the streamlines around the blade (see figure 4.4).

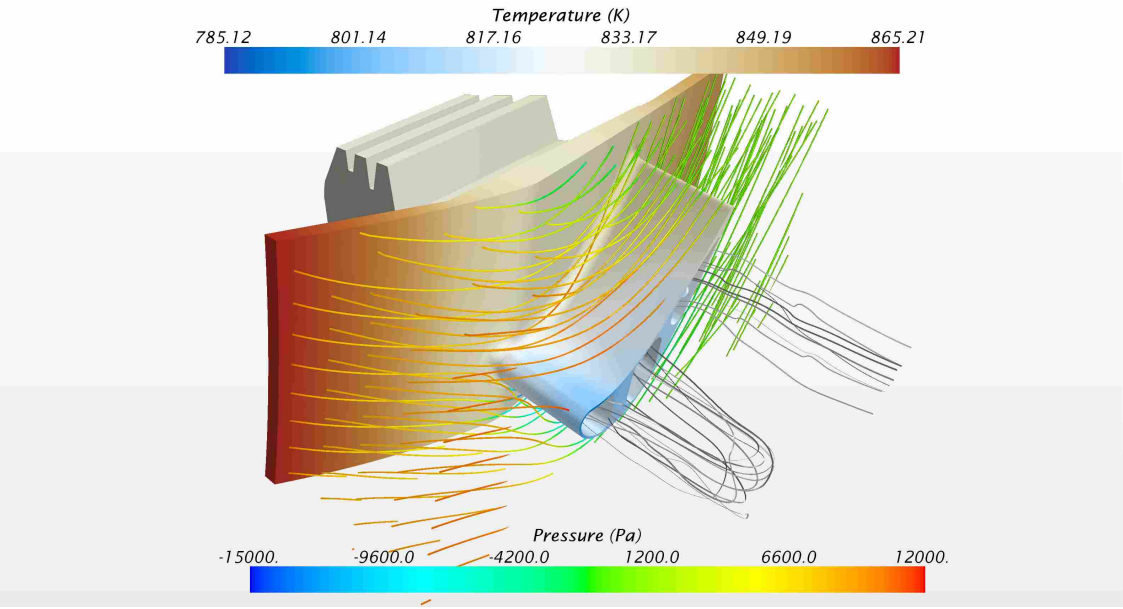


Figure 4.4: Conjugate Heat Transfer Simulation (Cooling flow within the turbine blade cools the solid blade)

This model did not lend itself as well to working in collaboration. Particularly this was because of the 3D-CAD Modeling part of the set-up which took about $\frac{1}{5}$th of the set-up time to complete, and could only be done by one user due to 3D-CAD Modeling mode only having one scene for both users. During the time when the 3D-CAD Modeling mode was being used there was much less ability for the users to work collaboratively The other user was only able to set-up continua which became a bottleneck until the geometry parts were created.

4.6 Crescendo Case

The Crescendo case model was also received from CD-adapco, and is used as a demonstration model, particularly for the benefits of using the surface wrapper. The model is of a helicopter

exhaust enclosed by the engine compartment housing, and includes many small surfaces such as hoses, bolts, and hinges, and also has many holes in the surface. On this model the solid volumes were closed, but intersecting at many points. The volume of cooling air is defined by the fluid volume outside the exhaust and inside the housing. However, the solid of the housing has many holes and intersecting surfaces that must be fixed prior to having a closed and manifold volume for the cooling flow. These areas include gaps where the front and back of the housing come together, and holes in the back of the housing. Because of these areas, a large amount of the time spent on the model set-up involved using boolean operations and Surface Repair mode to prepare the model for meshing.

In the case of this test, the use of Surface Repair mode was selected over the surface wrapper, to better show the time savings of collaboration, without including meshing operations for geometry preparation.

For a two user set-up, the model was imported as one part, with one surface, and then split into surfaces that correspond to the back of the housing, front of the housing, four sections of the exhaust, and one for all of the small surfaces described above. Two parts were created from the surfaces: one for the housing, and one for the exhaust. The small parts, hoses, and bolts were deleted in order to de-feature the model and make the surface repair easier. User 1 worked mainly with Surface Repair mode in order to combine the two solid volume of the housing, and delete the hinges, in order to create two closed boundaries that would enclose a solid volume: the interior housing and exterior housing. This required the intersecting surfaces to be deleted where free edges existed, and also surfaces to be merged together and deleted in other areas where there was no intersection. User 2 primarily worked with boolean operations in order to create closed and manifold volume of the exhaust solid.

The creation of inlet and outlet boundaries enabled the surface interfacing the hot air flow on the inside of the exhaust to be separated from the surface that intersecting the cooling air on the exterior of the exhaust. The next step was to create separate regions. The easiest process was creating one part for the solid, one part containing only the inlet and outlet surfaces of the external cooling flow, and another containing only the inlet and outlet surfaces of the internal heating flow. Each of these parts became a separate region once made closed and manifold. In this case, the two fluid volumes were not yet closed or manifold. The interface between fluid and

solid that completed the fluid volumes was created from surfaces on the solid part. At this point the solid part had all surfaces necessary to be closed and manifold, but had to be opened in Surface Repair mode to merge all free edges together before the volume was fully closed. This part was duplicated twice to create three copies of the solid. Each of the two new copies was combined with the individual fluid domain parts, resulting in a region defined by the fluid-solid interface and inlet and outlet boundaries for that domain, as well as containing surfaces that were not part of either fluid domain. This is shown in figure 4.5, where the tan volume is the internal flow, the green volume is the external flow, and the red surfaces are the parts of the solid that do not share a boundary with any of the fluid domains. In each of the fluid flow parts, the surfaces that were not part of any interface related to that part were deleted, but the interface surfaces remained exact duplicates of the interface surface of the solid part. For example, for the internal heated flow of the model, all green and red surfaces were deleted because they were not boundaries of the internal flow domain. The remaining tan surfaces other than inlet and outlet were exactly the same as those surfaces in the solid part, but a copy of each of these duplicate surface needed to exist on both the solid part and the respective fluid part. After the free edges were zipped together in Surface Repair mode, and regions were created from the parts, interfaces were created from the duplicate boundaries in the solid and fluid regions. The same process was used for the external flow where only the green surfaces were kept and all others deleted. If all surfaces that were copied from the solid part - including the surfaces that were not wanted for that part - were combined, the surfaces could be split into the three domains shown in figure 4.5 because they were non-contiguous. The rest of the set-up process then became similar to the process of the previously defined models, consisting of defining the physics, setting up reports, and setting up scenes.

It can be noted that the inlet of heating flow does not begin at the surface boundary, and that there is a volume within the exhaust that is not occupied by any regions (see figure 4.6 where the interior of the exhaust does not show a fluid temperature on the created plane surface). This is because a boundary cannot be both an inlet and interface at the same time. The interface that would have been created between the inlet boundary and solid boundary would take over the boundary properties, and the flow would not exist.

Because of the amount of time required to repair the surface of this model, finding the best way to set up this model was a process of trial and error. Each of the three times the model was

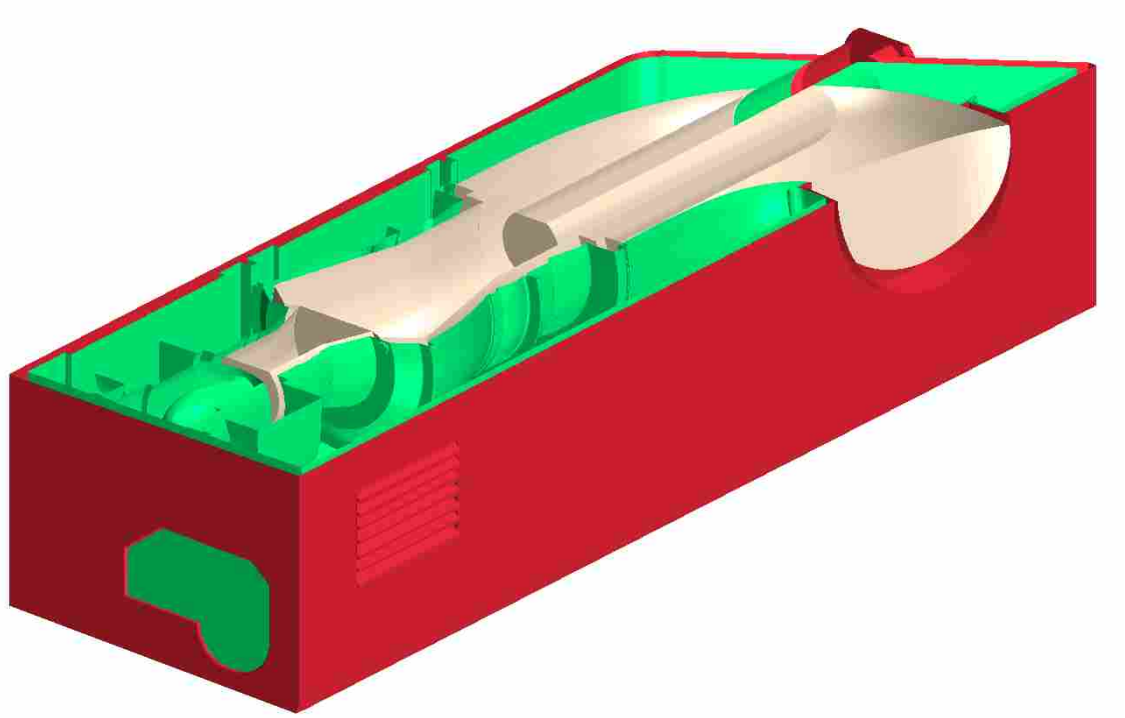


Figure 4.5: Closed and manifold volumes of each domain. The tan volume is the internal heated flow, the green volume is the external cooling flow, and the red surfaces are from the solid domain.

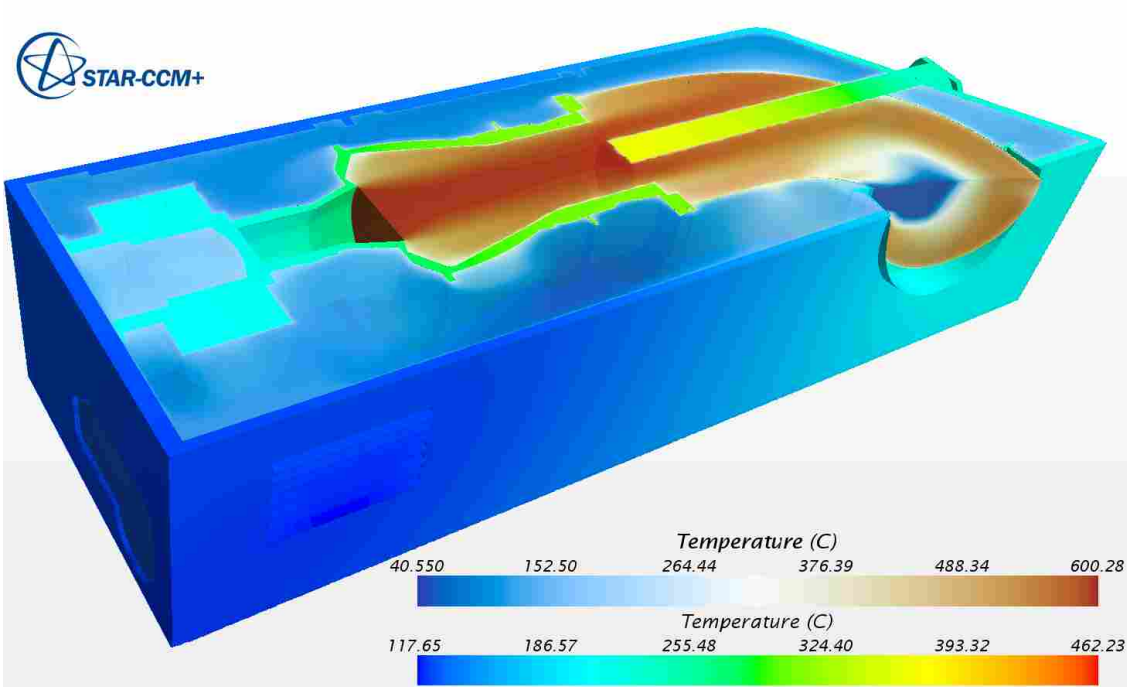


Figure 4.6: Crescendo case model. This scene shows the temperature of the exhaust and cooling flow as well as the surface temperature of the exhaust and housing.

set-up by multiple users, the approach for how the users would interact changed. The original well developed plan involved all surface repair being done by one user while the other user set-up physics and meshing models before both could work together on defining the boundaries. This plan resulted in a lot of down time for user 2 while user 1 worked in Surface Repair mode. The next plan involved user 2 repairing the exhaust surfaces with boolean operations. Through this plan many of the bottlenecks of surface repair were reduced, but there was still some down time for user 2 while Surface Repair mode was being used by user 1. The last plan did not detail an exact step by step process, but followed the previous two plans in that user 1 repaired the surface of the housing, and user 2 repaired the surface of the exhaust. One advantage that the Crescendo Case model provided to a collaborative set-up was the independence of each part which allowed the different users to do their respective surface repair at the same time, using either Surface Repair mode or boolean operations. Conjugate heat transfer problems are a great example of when surface repair is required, but the associated bottlenecks can be avoided. Communication became critical for this model at moments where the users could look ahead to plan around the down time that might be experienced. Without following a detailed plan, the two users were able to coordinate entering Surface Repair mode, and avoid errors and bottlenecks. As user 1 exited the surface repair mode to split, combine, and rename specific surfaces, user 2 was able to enter surface repair when it was needed for quick operations on the exhaust. Frequent communication between users while using a very loose plan worked much better than a strict plan because it allowed the users to adapt while working, and particularly allowed both users to take advantage of the benefits of using Surface Repair mode. Because the users made sure to communicate before entering Surface Repair mode, there was never any problem with trying to enter at the same time.

Bercovitz et al. [1] note that coordination capabilities are developed by teams who go through repeated interaction in collaboration efforts. This was seen throughout the work on this model in how by diverging from the plan the users were able to coordinate and further increase the time reduction. The best practices that were learned as this model was tested were documented by the users and applied in subsequent testing of the model. The most important is the use of boolean operations synchronously with surface repair mode. This eliminated the bottlenecks caused by the single user nature of surface repair and the necessity of extensive surface repair on this model, and also encouraged more communication between the users. Other best practices were the separation

of parts for separate scenes used by each user, and making sure that each user worked on separate parts during the surface repair process. These best practices are explained further in chapter 6.

4.7 AFRL Film-Cooled Turbine Vane

This geometry was provided by the US Air Force Research Laboratory [15]. The model is of a single vane from a low pressure turbine. Because the goal of these tests was to test collaboration on complex models, the model's complexity was increased by duplicating and rotating the vane to form a full annulus. The creation of the full annulus presented challenges that were unique to this model. The full annulus consisted of 23 vanes, but the model did not contain the full 1/23rd section of the annulus. The gaps in the geometry that were caused by the rotations had to be filled as part of this model set-up. Another difficulty presented by this model was the determination of the center of rotation required to create the full annulus. The last difficulty, which made this model particularly different than the others, was the model's size. The single vane geometry contained nearly 650 cooling holes. The created full annulus model with 23 blades resulted in a surface consisting of seven million tessellated surfaces. The process of duplicating and rotating blades, and repairing the surfaces required more computing time then would be required for smaller models. This created bottlenecks in the process, where the users had to wait for operations to execute for a longer time than had been previously experienced with other complex models.

The boundary conditions for the model were calculated with the following equations:

$$T_1 = \left(\frac{1}{T_{t_1}} \left(\frac{\gamma - 1}{2} {M_1}^2 \right) \right)^{-1} \tag{4.1}$$

$$p_1 = \frac{p_{t_1}}{\left(\frac{T_{t_1}}{T_1} \right)^{\frac{\gamma}{\gamma - 1}}} \tag{4.2}$$

$$a_1 = \sqrt{\gamma R T_1} \tag{4.3}$$

$$v_{inlet} = a_1 M_{inlet} \tag{4.4}$$

Initial Conditions		
	Inlet Pt [psia]	65.82
	Inlet Tt [R]	550
	Inlet Alpha	0
	Inlet Phi	0
	Inlet M	0.08
	Outlet Ps [psia]	38.72
	Outlet Alpha	76.88
	Outlet Phi	0
Boundary Conditions		
Core Flow		
	Pt_Core [psia]	65.82
	Tt_Core [R]	790.8
	Alpha	0
	Phi	0
ID Cooling Flow		
	Pt_ID [psia]	68.49
	Tt_ID [R]	435.9
	Alpha	0
	Phi	90
OD Cooling Flow		
	Pt_OD [psia]	66.86
	Tt_OD [R]	439.1
	Alpha	0
	Phi	-90
Outlet		
	Ps Exit [psia]	38.67
Metal Attachment		
	T_attachment	549.7

Figure 4.7: Boundary conditions of the AFRL Film-Cooled Turbine Model

These equations were calculated assuming $\gamma = 1.4$ and $R = 1716.27\frac{ft^2}{s^2R}$ using boundary values provided by the AFRL. These values are outlined with green boxes in figure 4.7. The static temperature of the inlet of the main flow was calculated from T_{t_1} in equation 4.1, and the static pressure was calculated from p_{t_1} in equation 4.2. The speed of sound was calculated at the given Mach number with equation 4.3 and the inlet velocity was calculated with equation 4.4. The calculated values and all input parameters to this model are given in appendix A.2.

The set-up of this model consisted of three phases:

1. Defining surfaces on the initial vane
2. Duplicating, rotating, and repairing the vanes
3. Defining meshing, solving, and post-processing conditions

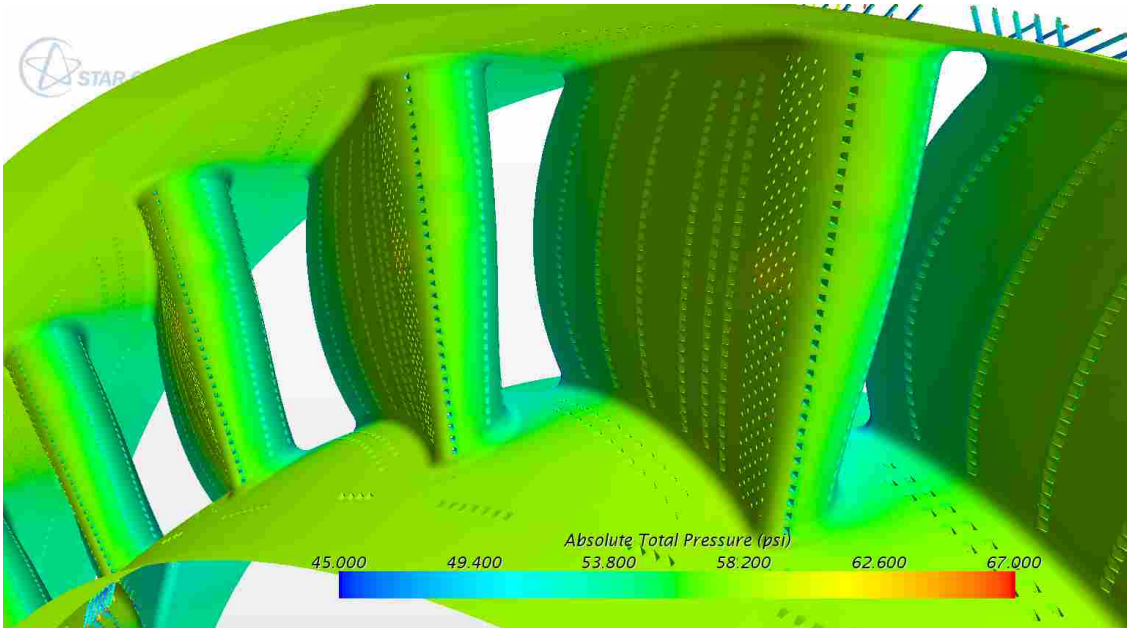


Figure 4.8: AFRL Cooled Turbine model scene showing the pressure on the surface of the turbine vanes

In the first phase of a two user test, user 1 imported the model into 3D-CAD modeling mode as two parts that almost defined the solid and fluid surfaces. The parts were merged together to delete the duplicate surfaces, and create more defined solid and fluid volumes. While 3D-CAD was being used by user 1, user 2 created a new coordinate system with the origin at the center of rotation for the annulus. After creating the coordinate system for rotation, user 2 was able to create a part from the 3D-CAD fluid part while user 1 used 3D-CAD Modeling mode to split the blade into surfaces for the solid wall, and interfaces for the outside of the blade, cooling holes, and inside of the blade. 3D-CAD Modeling mode was closed and both users deleted the duplicate surfaces and periodic surfaces. User 2 specified the models and properties for the physics and mesh as user 1 created a new part for just the interface surfaces, resulting in three parts: solid, fluid, and interface.

In the second phase the two users were able to work in opposite directions, duplicating and rotating the three parts until they met at the opposite side. The users rotated parts in opposite directions to create the full annulus, and ensure that no vanes in the annulus overlapped. A few parts and associated surfaces were combined by each user, repaired, and then combined with other parts. Because the users were doing the same process on the same part, this phase did not require

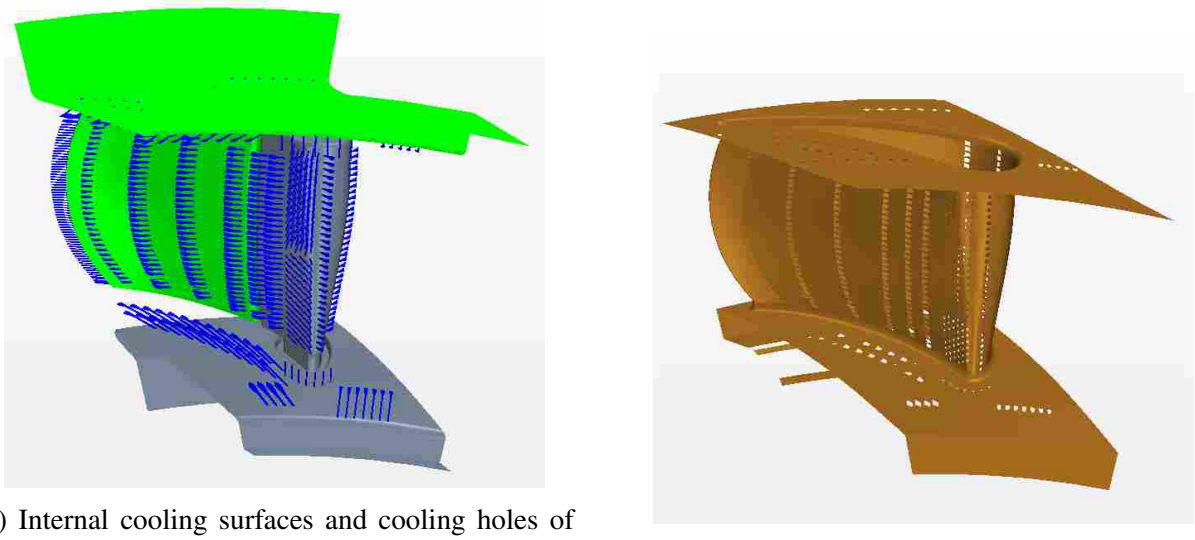
any planning. However, it did require communication due to the need to repair the surface. Surface Repair mode was used to seal the edges together of each newly created vane. The two users had to ensure that only one user was using the Surface Repair mode of STAR-CCM+ at a time. The two users alternated using the Surface Repair mode while duplicating, rotating, and combining the vanes. A user working faster than the other would eliminate bottlenecks by continuing to create new vanes until the full annulus was complete with three parts: solid, fluid, and interface. The interface part was duplicated and combined with the solid and fluid parts. Surface Repair mode was then alternated between the two users. User 1 sealed the interface edges to the solid, and user 2 sealed the interface edges to the fluid. This resulted in two closed and manifold parts for solid and fluid.

During the third phase user 1 set up the boundaries for the solid region, and defined the interface boundaries between both parts. User 2 specified the boundaries and set up boundary conditions for the fluid region. User 1 then created a report, monitor, and plot for temperature of the vane surface, and each user created a separate scalar scene: one for vane temperature and one for vane absolute total pressure (see figure 4.8).

Unique to this model was the size of the geometry. In previous models that had been completed for this research, the operations did not take much longer to execute by the server than it took for the users to select that operation. With this model there was significant waiting time by the two users where the computer was completing the execution due to limited transfer speed across the server. This occurred as several surfaces or parts were combined together, a part was duplicated or rotated, and as surface repair mode was being opened, executed, and closed. This was the same situation for both the single-user and multi-user set-up. The result is that less of the model set-up time was spent in user operations, and more in server transfer, and this resulted in a longer test, with less speed up resulting from collaboration. The benefit was that it gave the users extra time to communicate and plan throughout the process of setting up the model. This extra time occurred during the periods of bottle necks in the server transfer, and meant that the users did not have to stop working specifically to be able to communicate and plan.

A specific problem with this model was due to the difficulty of generating a conformal interface between the two regions. This was due to the fact that zipping edges during surface repair significantly changed the shape of the cells as a result of the gaps from the revolved parts.

The interface cells of the solid and fluid regions changed differently based on the surrounding cells, and this caused the two parts of the interface to be misaligned. This caused errors while merging the interfaces together, which resulted in no meshes being generated, and errors in the surface mesh, which resulted in no closed volume in which to generate the volume mesh. Initial tests were done on only the fluid region of this model while methods were explored for creating the conformal interface and fully generating a volume mesh. The problems with the mesh were fixed by creating additional surfaces from the interface. Because the cooling holes of the interface were so small, the mesh of the rest of the interface was pushed to the smallest mesh size defined. This caused a large cell count, which contributed to the errors in the mesh generation.



(a) Internal cooling surfaces and cooling holes of Interface

(b) External interface surface

Figure 4.9: Interface surfaces of the AFRL Film-Cooled Turbine Vane

The inside surfaces of the cooling holes were split within the 3D-CAD modeling mode. All hole surfaces became a single surface, and the remaining interface could be split into the three remaining non-contiguous surfaces because the only link between them was the cooling holes. This resulted in four surfaces (see figure 4.9) that together represented the entire interface. Figure 4.9a shows the surfaces of the cooling holes and two internal interfaces, and figure 4.9b shows the external interface. These four surfaces all formed one part. This same interface part was combined with both the solid and fluid parts, and a separate in-place interface was created for the four surfaces of each part. This allowed mesh size to be specified on each individual boundary of

the interface part, and resulted in a correct generation of a complete volume mesh with a conformal interface between the two regions. The model was solved and compared qualitatively to the AFRL documented values [16] to ensure that the set-up was correct.

CHAPTER 5. INTEGRATION OF CAD AND CFD USING STAR-NX

CD-adapco has created parametric connections for NX, Catia, SolidWorks, Inventor, Pro/Engineer, Creo, and SpaceClaim. Each of these CAD programs has an add-on that enables the integration of CAD and CFD. Because much of the effort to develop collaborative CAD capabilities at BYU has been focused on NX, the research in this thesis focuses on STAR-NX, the integration of NX and STAR-CCM+. This chapter shows the benefits of integrating CAD and CFD through STAR-NX, and details how the integration is improved in order to support multi-user collaboration.

5.1 STAR-NX

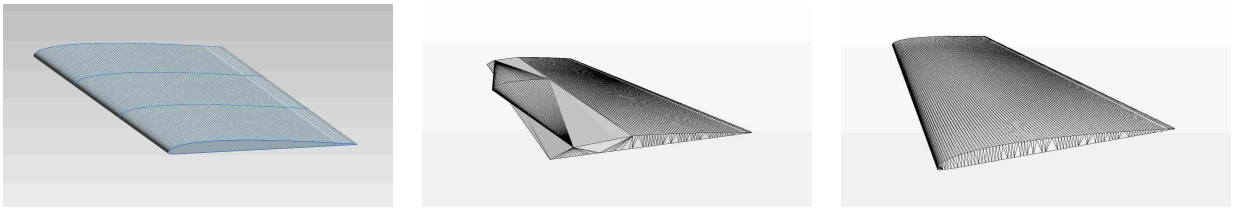
STAR-NX is added to NX, creating a toolbar and simulation tree for STAR-CCM+ within the NX environment. There are three specific benefits that result from the integration of CAD and CFD through STAR-NX:

- Direct geometry transfer between NX and STAR-CCM+
- Control of geometry parameters in both programs
- CFD set-up capabilities within NX

The STAR-NX toolbar is shown in figure 5.1. The STAR-NX operations described in this thesis are circled in green, and include Transfer CAE Model, Surface and Volume Mesh, Build and Run, Run Solver, Select Parameters, and Transfer Geometry.

Figure 5.1: The STAR-NX toolbar

As explained in chapter 1.2, a major problem present in the transfer of geometry from a CAD program to an analysis program is the use of neutral formats. STAR-NX presents a solution to this problem by allowing geometry data to be transferred at the click of a button. This is made possible because NX and STAR-CCM+ both use the Parasolid geometry kernel. Although the geometry is still represented in different ways in each program (NURBS surfaces for NX and tessellated surfaces for STAR-CCM+), the underlying data is the same as long as this data is not lost through a neutral file transfer. The result of this geometry transfer is that NX and STAR-CCM+ are both using the same model with the same data, and this reduces the amount of surface preparation that is necessary for a specific model.



(a) Airfoil geometry in NX before being exported

(b) Imported .IGES geometry in STAR-CCM+

(c) Geometry transfer via STAR-NX

Figure 5.2: Geometry transfer from NX to STAR-CCM+

The benefit of a direct geometry transfer can be seen in figure 5.2. The geometry of an airfoil was created in NX (shown in figure 5.2a), and transferred to STAR-CCM+ both through a neutral format and through the STAR-NX interface. The inconsistencies of a neutral file format are shown in this case when the airfoil model is transferred as a .IGES file (shown in figure 5.2b). In this transferred model the vertices are out of place on the leading edge of the airfoil, and the surfaces connect the wrong vertices. These surface errors are not present when the STAR-NX add-on is used for the transfer (shown in figure 5.2c), where the vertices and surfaces of the airfoil are identical to those as represented in NX. Even though some geometry preparation will likely be needed for more complex models, the STAR-CCM+ representation will be a better representation of the NX representation, requiring less geometry clean-up.

The ability to control geometry parameters further improves the geometric transfer via STAR-NX. STAR-NX creates a parametric connection between the NX part and the STAR-CCM+

model. Within NX, desired NX expressions can be selected as variables for the model. These variables can be selected within NX by selecting the "Select Parameter" button on the STAR-NX toolbar (see figure 5.1). If both programs are open, these parameters can be changed in either program, and updated to the other program. The geometry and selected parameters can be transferred to STAR-CCM+ without any CFD parameters by clicking the "Transfer Geometry" button (as shown in figure 5.1). In STAR-CCM+, the geometry can be updated by editing the CAD Client Model Parameter value and updating changes. A major benefit of this parametric connection is that multiple design iterations of a single model can be performed without requiring a model transfer each time. A CFD model that has been set up with boundary types and values can be updated with these specified parameters while maintaining the boundary properties of the previous iteration.

The STAR-NX toolbar and tree allow CFD operations including boundary definition, boundary value set-up, meshing, and solving to take place using the NX environment as the client for STAR-CCM+. Using the "Transfer CAE Model" button on the STAR-NX toolbar (see figure 5.1), all of these aspects of the model can be transferred to the model within STAR-CCM+. Each step of the set-up process that is done through the STAR-NX interface is done on a STAR-CCM+ simulation file that is created when the STAR-NX toolbar is used. The boundaries and mesh of the CFD set-up are displayed in NX as shown in figure 5.3, which shows a simplified car model after being meshed in NX. The boundaries types have been assigned in NX, where the surface of the model with red rectangles shows an inlet boundary, and the surface with blue circles shows the symmetry plane of the model. This model is described in chapter 5.4. The NX part file and STAR-CCM+ simulation files are linked together, but only the simulation file is changed as the CFD set-up within NX takes place. The "Build and Run" button opens a STAR-CCM+ client where the simulation is solved.

Figure 5.4 shows the architecture of STAR-NX, and the interaction of the NX client with the STAR-CCM+ server. STAR-NX is the link between NX and STAR-CCM+, allowing geometry parameters to be passed from STAR-CCM+ to NX, and geometry and CFD parameters from NX to STAR-CCM+.

There are three areas where STAR-NX lacks useful capabilities. First is the ability to only use solid models within NX. Sheet models can not be used for STAR-NX. Second, the CFD capabilities within STAR-NX are also limited, and the only volume meshing models that can be used

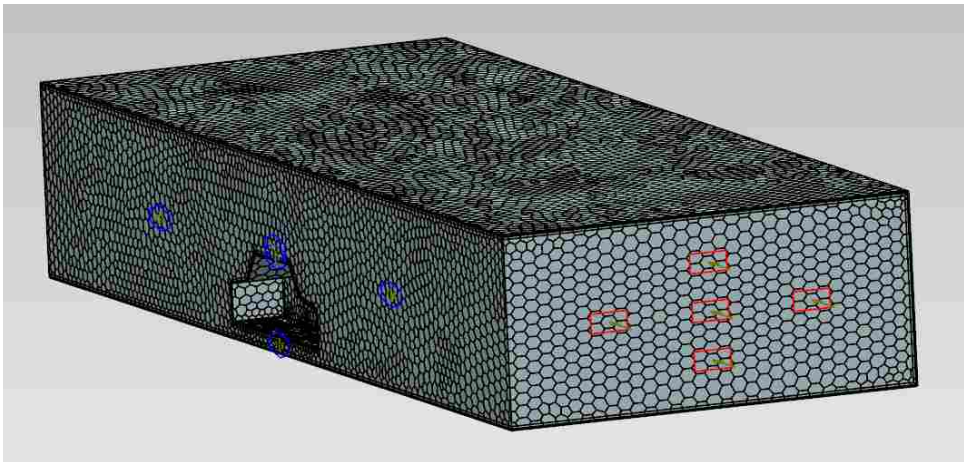


Figure 5.3: STAR-CCM+ set-up within NX using the STAR-NX toolbar. The model is of a simplified car in a wind tunnel.

are the prism layer mesher and polyhedral mesher [10]. Third, the CFD set-up is only maintained during a geometry parameter update if no surface preparation has been performed. If the surface repair mode and boolean operations have been used to fix errors in the geometry, these repairs will be undone when the model is updated. Any updates to parameters will be performed on the model that is shared between NX and STAR-CCM+, without any geometry preparation updates. This limits the effectiveness of the integration between NX and STAR-CCM+ that is given by STAR-NX, and for this reason this thesis evaluates STAR-NX when used with a multi-user version of NX (NXConnect) and multiple collaborating clients in STAR-CCM+

5.2 Collaboration with STAR-NX

The STAR-NX add-on was added to NXConnect by adding the STAR-NX palette and application to the custom directories of NXConnect. The same toolbar and tree for STAR-NX appeared on the NXConnect client, but with different capabilities. These capabilities were tested by two users working on the same part on two separate NXConnect clients. Similar to the interaction with a single user version of NX (see figure 5.4), STAR-NX was able to connect in the same way to a single client of NXConnect. The problem that was observed is that the CFD operations were not sent to the server, and as a result were not updated in the other NXConnect client. While geometry parameter changes made in the NX environment of one NXConnect client were updated to the server, other NXConnect client and STAR-CCM+ server, any changes pushed to the NXConnect

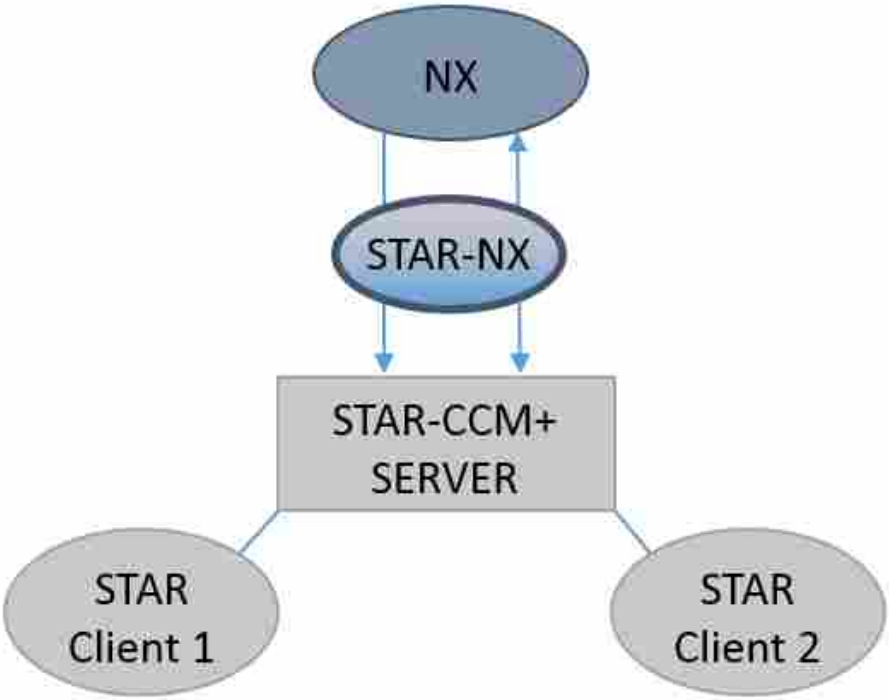


Figure 5.4: Architecture of Star-NX

model via model updates within STAR-CCM+ were not recognized by the server, and thus not updated in the other client. This architecture and interaction are shown in figure 5.5, where the link to the NXConnect server from Client 1 is not complete for all operations, particularly STAR-NX parameter updates. In this case the only NXConnect client that was updated was the client in which the STAR-NX toolbar button was clicked to transfer the geometry or CAE model. This was due to the fact that the indicator within the NX model that a parameter or feature has been updated is not incorporated into NXConnect. The NXConnect server did not recognize that a change had been made despite the changes being made to one of the clients.

The problem with the interaction of STAR-CCM+ and NXConnect can be seen in figure 5.5 in that the STAR-CCM+ server is not connected to the NXConnect server. STAR-NX was created to interface with an application rather than an architecture. The thick client - thin server architecture of NXConnect uses each client as an individual application that is connected to and kept updated through the NXConnect server. For this reason, when the STAR-NX toolbar and tree are used within NXConnect, the STAR-CCM+ server is created from and linked to an individual

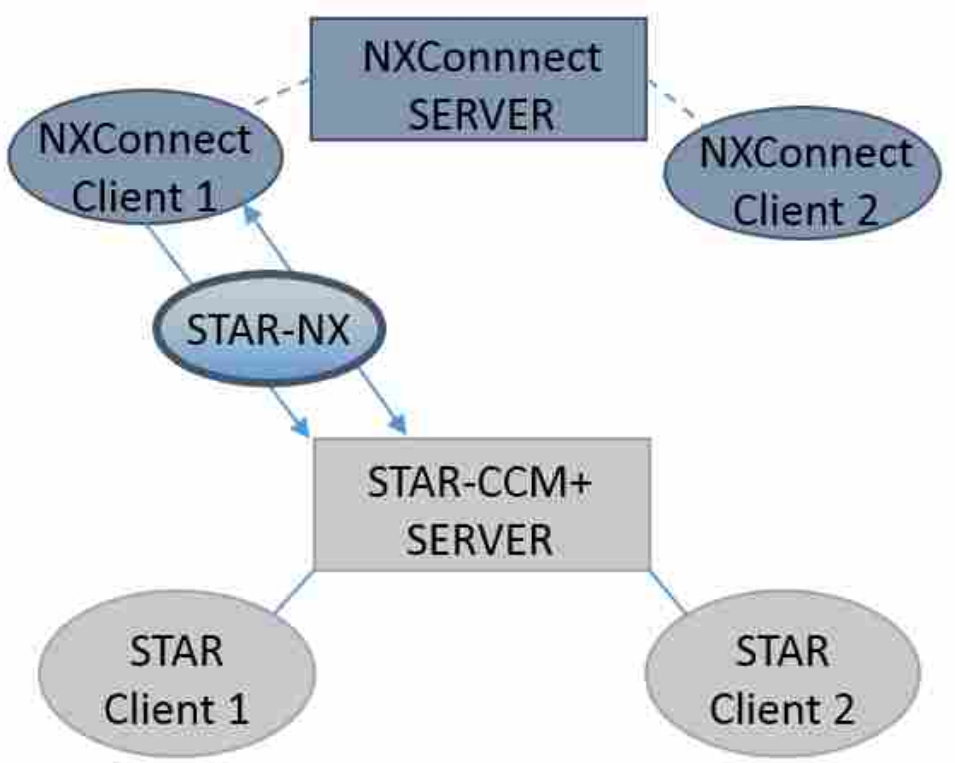


Figure 5.5: Previous method of STAR-NX interaction with NXConnect. Parameter changes made through STAR-CCM+ are not applied to all NXC clients.

NXConnect client. This problem would likely be avoided if NXConnected utilized a thin client - thick server architecture, resulting in all interaction through STAR-NX taking place from one server to the other. However the NXConnect architecture was necessary to take advantage of the functionality and API of NX. This relates to the CAD and analysis integration bottleneck described by Slotnick et al. [4]. STAR-NX was not created with a thick client - thin server or any multi-user architecture in mind for the CAD connection, and NXConnect was not created with STAR-NX in mind. Thus the integration of these multi-user programs has limitations and the NXConnect server does not receive updates through STAR-NX.

Several workarounds were found that enabled the geometry changes to be updated in both clients during the initial testing of STAR-NX with NXConnect. One method was to open the dialog box of the feature that needs to be updated. This required the dialog box of the sketch, extrusion, revolve, etc. containing the parametrized dimension to be opened and closed. Similarly the NX expression window could be opened and closed. What was seen from these workarounds was that

the connection between client and server in NXConnect still existed, but although workarounds existed, they were not convenient for the user. NXConnect has methods of recognizing when changes are made to the model and ensures that these changes are made to all clients connected to the same model. In the NX environment changes would be made through the feature dialog boxes and NX expression window, and these methods were satisfactory for these types of changes. However, the change to the model through a STAR-NX update does not trigger any of these methods, resulting in no changes being made to other clients. Because NXConnect is continually being developed at BYU, additional development was proposed to improve the geometry transfer of NXConnect, and use with STAR-NX.

5.3 Development of STAR-NX

The ability to combine the benefits of collaboration and integration are evident in situations where CAD users are collaborating on a model that requires multiple design iterations. One research objective was to make the necessary changes within NXConnect to enable this combination. A problem that limits the options for how these changes can be made is the lack of API for STAR-NX. Without the API no changes can be made to STAR-NX, and changes made to the STAR-CCM+ model via STAR-NX are not displayed in the output window. The proposed architecture for keeping NXConnect clients consistent is shown in figure 5.6. The Multi-User CAE tool (MUC) is a section of code added to NXConnect that triggers an update to the server and to all connected clients when a change is made to the model through STAR-NX. A benefit of NXConnect is that there is an existing method of pushing the changes to the server and to all other clients. The MUC is created as a way to identify changes through STAR-NX, and call the existing pushers of NXConnect to update these changes.

Creating a journal within NX of actions performed by the user is a method of seeing the specific manner in which the user can programmatically use these same actions. The created journal is one method of seeing the API that is revealed to the user, and how it is used. A proposed method for this research involved using “undo marks” from the NXOpen API as a method of recognizing within the NXConnect client that a parameter change has been pushed from the STAR-CCM+ client. A journal recorded with a parameter changed in this way revealed a unique undo mark, “Update Expression Data." The initial attempt at improving the integration of NXCon-

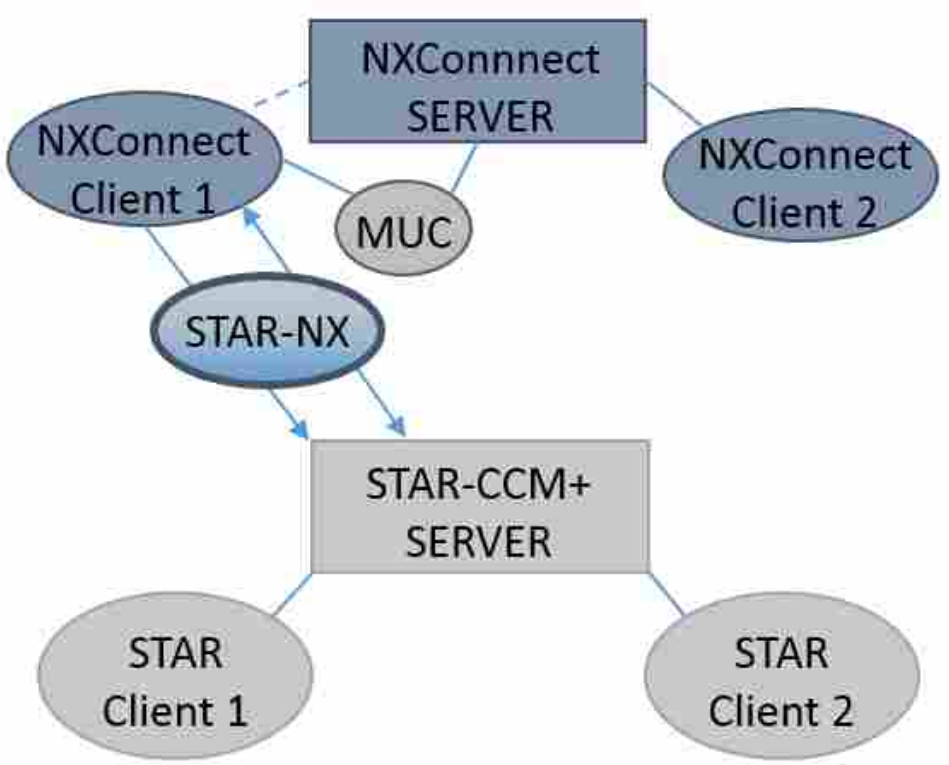


Figure 5.6: Proposed architecture for STAR-NX with NXConnect. Multi-User CAE Tool (MUC) links parameter changes in the NXC client to the server

nect and STAR-NX involved this undo mark, with the plan that when created it would trigger the NXConnect pusher. This method was unsuccessful and although the undo mark was being created as the part was updated through STAR-NX, NXConnect was unable to recognize this undo mark. This is because the undo mark was created and deleted while the STAR-CCM+ server had control of the process. The undo mark was already deleted by the time NXConnect regained control.

The second proposed method for creating the MUC was to monitor expressions in all clients for changes. NXConnect runs a loop every 0.1 second that monitors the undo marks created by any connected client. The added STAR-NX code uses this loop, and is run once every 50 times through the loop. Every five seconds the STAR-NX function compares the current values of the model expressions with the values from the previous time the function was run. The STAR-NX function is run less frequently than the NXConnect loop because updates to the model via STAR-NX occur less frequently, which allows time for the model to be rebuilt as changes are made to different features. The STAR-NX function monitors the number of expressions, and the name

and value of each. If any of the expressions are added, or have changed in either name or value then already existing NXConnect pusher will be called. The pushers update the expressions to the NXConnect server, which updates the changes all clients of NXConnect and STAR-CCM+. This method was successful in updating the geometry in all NXConnect clients. The STAR-NX code that was added to NXConnect is given in appendix B.

5.4 STAR-NX Testing

The updates to NXConnect and STAR-NX were tested on a model based off of the STAR-NX tutorial "External Flow Over a Car" [10]. The test model used a simplified car body as shown in figure 5.3. This test set the height, width, and length of the wind tunnel, and height and length of the car as variable parameters that would be tested. Each of these parameters were changed multiple times to ensure there were no bugs in the code, and these changes were made in both STAR-CCM+ and NXConnect. This testing was performed to ensure that the geometry of all clients was updated, and also to recognize any limitations of the MUC.

The model was created with sketches on six planes. A through curves solid was created through these planes to represent the body of the car, and the parameters for length and height of the car were chosen to be variable through the STAR-NX toolbar. Next a rectangular sketch was extruded to represent the wind tunnel, intersecting half of the the car. The intersecting surface of the wind tunnel was the symmetry plane for the model. The wind tunnel had variable parameters for length, height, and width (see figure 5.7). The Solid of the car was then subtracted from the solid of the wind tunnel to form a single fluid domain around the body of a car. In contrast to many of the CFD models used for this research, this CAD model was simple. The simplicity was to ensure that collaborative modelling capabilities of NXConnect and the supported operations of STAR-NX were not being tested, only the capability of the MUC to ensure that all geometric parameter changes were made to each client.

The use of a simplified model in the testing of the STAR-NX MUC was to focus on testing the ability for multiple users to collaboratively use the STAR-NX capabilities, rather than test the collaborative modeling capabilities of NXConnect, or the abilities of STAR-NX to create CFD models within NX with a parametric connection to the CAD model. NXConnect is in a state of continual testing and development to reduce any limitations, and the limitations of STAR-NX

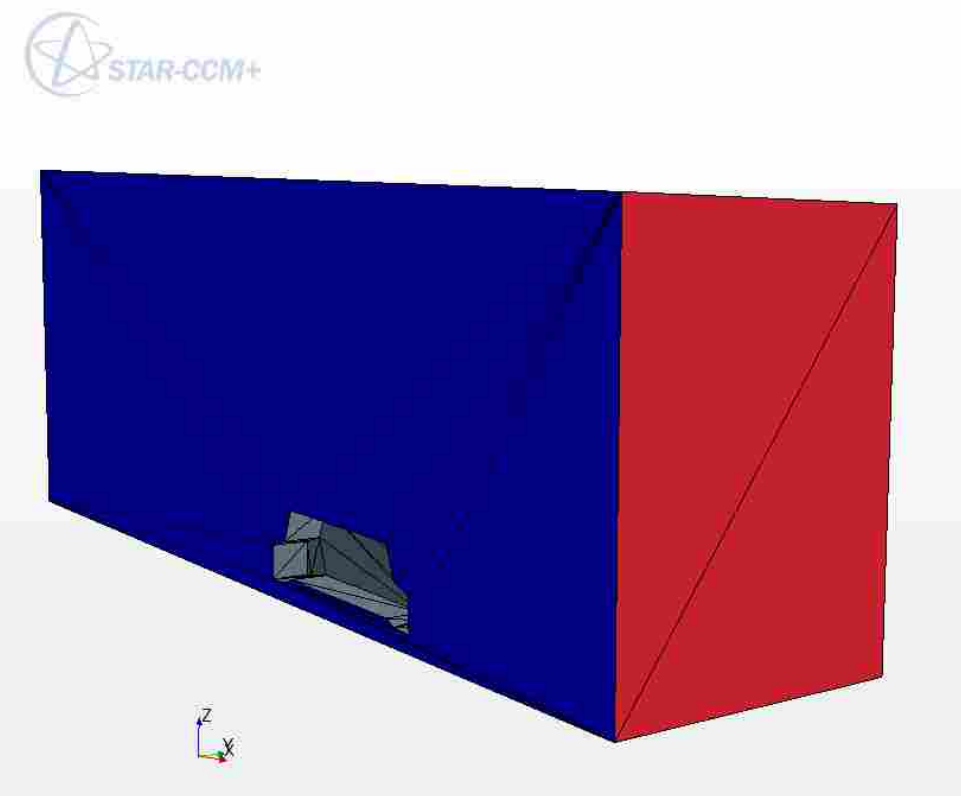


Figure 5.7: Model set-up of a car in an wind tunnel for testing the Multi-User CAE tool of NX-Connect with STAR-NX.

are well documented in the STAR-CCM+ User Guide [10]. What is tested here is the goal of allowing geometry parameter changes to all NXConnect and STAR-CCM+ clients when updated through STAR-NX, and to ensure that the CFD set-up performed in NXConnect is maintained as the geometry is updated.

Geometry preparation within CFD programs is necessary for complex models because these programs contain the capabilities for creating closed and manifold volumes. The solid boundaries of a CAD model define only the solid region of a model. On external flow models a wind tunnel can be created around the part with the solid part subtracted from the wind tunnel. This is a simple way to have a solid CAD model define a fluid CFD boundary but does not work for models with multiple regions or internal flow. There is no way within CAD programs to define an internal fluid flow domain. This must be done in pre-processing with a CFD program. This becomes a problem when using STAR-NX because any surface repair done to the model will be undone as the model parameters selected in STAR-NX are updated. The capabilities of the STAR-NX MUC

could not have been shown on a model with internal flow, which is another reason a simple model with external flow was used.

CHAPTER 6. RESULTS

Tests were performed to evaluate a collaborative CFD simulation set-up, and the multi-user addition to STAR-NX. The results of these tests and specific benefits for each model are described in this chapter.

6.1 Results of Multi-User Testing

The multi-user testing for STAR-CCM+ was completed on six models by two users, and compared to the model set-up of a single user. Each of these simulation set-ups were evaluated for the benefits of time savings, model improvement, and ability to see and correct mistakes in the model. There was also a discussion and evaluation of the overall impression of the users working in collaboration.

6.1.1 Time Savings of Multi-User STAR-CCM+

Each of the models tested with multi-user collaboration showed a decrease in set-up time. The percent reduction and speed up are shown in table 6.1, along with the amount of times each model was set-up by a single user and two combined users, and the average times for both individual and collaborative set-ups. The average individual and collaborative times are calculated from the recorded times in the fully completed individual and collaborative set-ups. Equation 6.1 is used to calculate % time reduction and speed up is calculated from equation 6.2.

$$\%\ Reduction = \frac{(Avg.\ Individual\ Time\ -\ Avg.\ Collab.\ Time)}{Avg.\ Individual\ Time} \tag{6.1}$$

$$Speed\ Up = \frac{Avg.\ Individual\ Time}{Avg.\ Collab.\ Time} \tag{6.2}$$

These results were only calculated for the operations in a set-up involving user time, and did not include iterative improvements to the model, or the process to figure out the correct set-up of the model.

Table 6.1: Time results of single user and multi-user tests

Model	Individual Set-Ups	Collab. Set-Ups	Avg. Individual Time (min.)	Avg. Collab. Time (min.)	% Time Reduction	Speed Up
Control Valve	4	2	10	7	30	1.43x
Multi-Region Heat Exchanger	4	2	17	11	35	1.55x
RAE2822 Airfoil	4	2	11	7	36	1.57x
Cooled Turbine	4	2	28.125	21.25	24.4	1.32x
Crescendo Case	4	3	90.25	58.7	35	1.53x
AFRL Film-Cooled Turbine	3	2	91	64	29	1.42x

The percent time reduction shown in table 6.1 is calculated from the average single user set-up times and two user set-up times. The RAE2822 Airfoil model had the highest time reduction at 36%, and both the Multi-Region Heat Exchanger model and Crescendo case model were close at 35% each, and the Cooled Turbine and AFRL Film-Cooled Turbine Vane models had less improvement at 24.4% and 29% time reduction respectively. The Control Valve model was the most simple set-up, and had a time reduction of 30%. The models that required the longest set-up time were the Crescendo case and AFRL Film-Cooled Turbine Vane.

Even on a complex simulation, especially the Crescendo Case model where there was lot of surface repair required, significant time savings were achieved by working collaboratively. The majority of the time savings can be attributed to the ability of the separate users to work on separate regions of the simulation. When the surfaces needed to be repaired one user could use boolean operations while the other used the Surface Repair mode. The main issue that decreased the time saving effectiveness of a multi-user set-up was having one user unable to do any work until the

other finished a certain operation. These bottlenecks are a real possibility if the steps for the set-up are not planned out. This requires collaboration between the users.

Speedup is another way to look at the benefits of collaboration, and is a way to understand possible trends in the scalability of the model for multiple users. This is shown in figure 6.1 with curves for both ideal and realistic speedups for more than two users. This plot is based on the Crescendo Case model, and the data points for one and two users are taken from the test results. The ideal curve assumes a linear change in speedup as number of users changes. The ideal curve is different than a speedup of $n\times$ because it includes the data point for the speedup of a two-user set-up, which was calculated using equation 6.2. However, it is more realistic to assume that the rate of change of the speedup will decrease as the number of collaborative users increases. The realistic approximation of speedup was extrapolated for up to five users using equation 6.3, which estimates the speed up for additional users beyond what was tested. This method of calculating realistic speedup estimates that the same percentage of decrease in speedup from $n\times$ that was seen from one user to two users would be experienced for each additional user. The realistic curve is only an estimate, and the actual values could be verified in model testing with more users. Using this method of calculating speedup (see equation 6.2), with n users, a speedup of $n\times$ is equal to a set-up time of $\frac{1}{n}$. This is ideal because no extra user time would be spent on a model. Although none of these tests resulted in a speedup of $n\times$, there is less time being spent on a simulation set-up, and this will lead to the model being ready for the next step of the engineering process sooner, resulting in less cost to the company creating the design.

$$s_{i+1} = s_i + (s_i - s_{i-1}) \left(\frac{s_{i-1}}{s_i} \right) \tag{6.3}$$

It was originally anticipated that a speedup of $n\times$ would occur for n users in collaboration as had occurred in the testing of NXConnect [3]. This would be expected if a single operation took the same amount of time regardless of the number of users performing the operation, and each step and operation of the simulation set-up could be overlapped. The benefit of a speedup of $n\times$ is that no extra man hours would be required for a simulation set-up. From the results in table 6.1 it can be seen that $2\times$ was not achieved by 2 users. There are two main reasons for this. The first is that user downtime occurred in areas where the workload of two users was not able to be

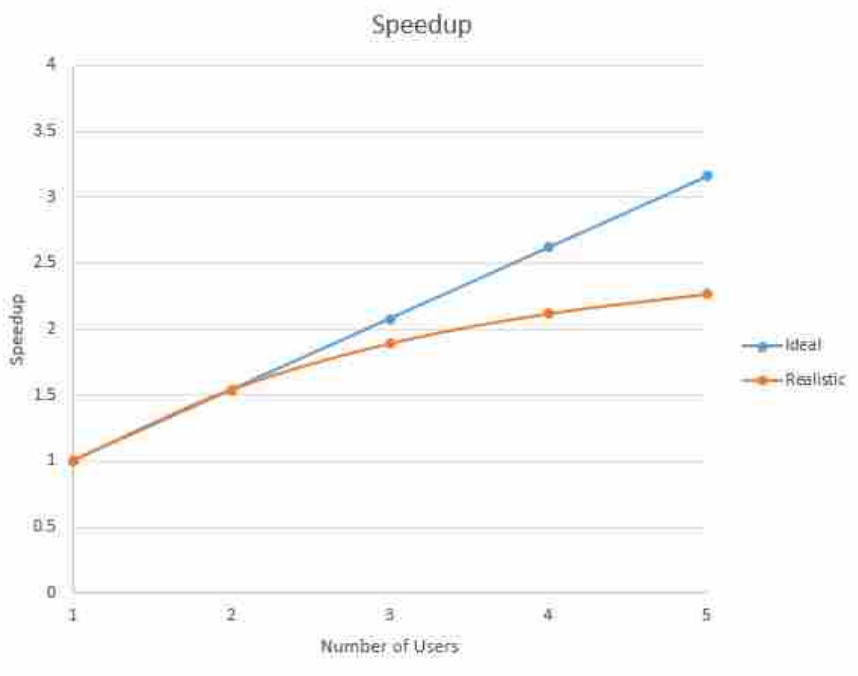


Figure 6.1: Speedup shown for the Crescendo Case model. The ideal speedup is scaled linearly from from the data for up to five users. The realistic speedup has a decrease in the rate of change of speedup as the number of users increases.

fully partitioned, and one user had to wait until an operation was finished by another user. The second is that even operations that could be performed synchronously sometimes took longer to be performed by multiple users. This is due to the work flow being changed in order to avoid user downtime that occurs in two situations: when processes cannot be completed in parallel, and when operations cannot be performed synchronously. Operations that cannot be completed in parallel are those that rely on data from a previous operation, for example, boundary properties cannot be set until after parts and surfaces have been turned into regions and boundaries. Processes that could not be performed in parallel were a problem in these tests because of the serial nature of the CFD process. The work flow for a single user is very serial, following the steps of geometry preparation, boundary definition, physics and mesh model selection, initial condition definition, and post processing in order. Planning was required to eliminate most of the downtime for a multi-user set-up that was caused by non-parallel processes. The changing of the order, particularly when two users completed the same operation synchronously slightly increased the total number of man hours spent on that operation. This can be seen for each model tested.

The steps taken for the Control Valve model are shown in figure 6.2. Areas that keep the speedup of this model from being 2× for two users are downtime caused by the serial nature of the CFD set-up process for the second user during the geometry preparation stage, and increased time required for geometry preparation, setting of boundary conditions, and post-processing.

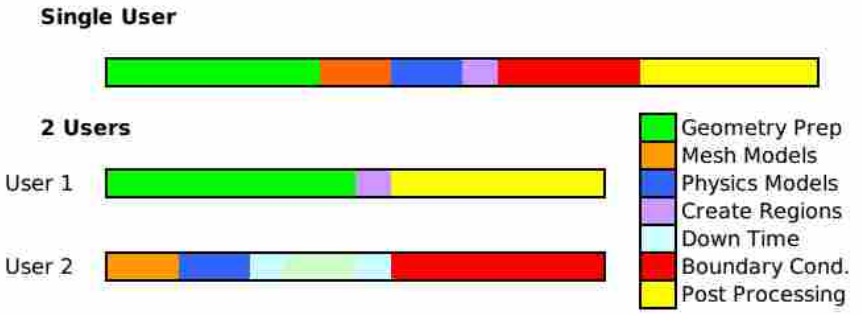


Figure 6.2: Operation breakdown for a single user and two users in the set-up of the Control Valve model.

The setting of boundary conditions in the Multi-Region Heat Exchanger model took more man hours when done synchronously for two users than for a single user. As shown in figure 6.3, downtime that was experienced by user 2 during the selection of meshing models and mesh set-up also contributed to the low speedup of this model.

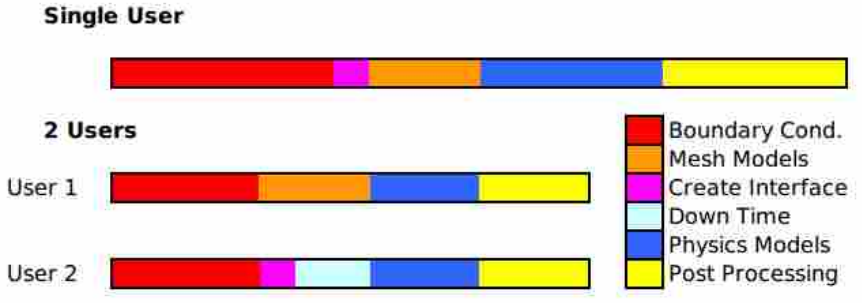


Figure 6.3: Operation breakdown for a single user and two users in the set-up of the Multi-Region Heat Exchanger model.

Downtime for user 2 and additional man hours were observed in the synchronous set-up of post-processing in the Transonic Airfoil model (shown in figure 6.4). User 2 waited for user 1 to finish the post-processing of the model. The increased number of man hours for post-processing

was a result of the separate reports being set-up individually rather than being copied with only a few parameters changed. This was done as a way to fill a greater amount of downtime that would have been experienced by user 1 if only user 2 had set-up post processing. In both cases the speedup would have been the same, but having different users work on each report and scene helped to ensure that the operations were performed correctly. The speedup is greatest in this model, but mainly because no geometry preparation was involved.

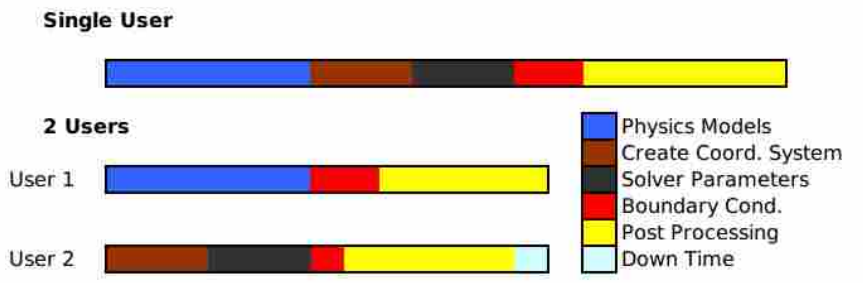


Figure 6.4: Operation breakdown for a single user and two users in the set-up of the Transonic Airfoil model.

As shown in figure 6.5, the time spent by user 1 in 3D-CAD Modeling mode resulted in downtime for user 2 during the set-up of the Cooled Turbine model. This was due both to the serial nature of the set-up and lack and synchronous capabilities of an operation. The lack of multi-user capabilities in 3D-CAD Modeling mode required user 2 to work on other operations as the geometry was prepared. The serial nature of the set-up allowed only the selection of meshing and physics models to be performed while user 1 prepared the geometry. Downtime occurred as soon as user 2 completed these operations and lasted until parts and regions could be completed by user 1. Extra time was also spent in post-processing for two users. This was due to the fact that the scenes created in this model were duplicated and only slightly edited to show different properties in a single user set-up. When two users created the scenes synchronously, these scenes were not duplicated, thus the number of man hours increased for post-processing in a two user set-up of this model.

Although most operations for geometry preparation in the Crescendo Case model were able to be performed synchronously, and although the model set-up was planned in a way to avoid the bottlenecks in these operations, more man hours were spent on geometry preparation in a two user

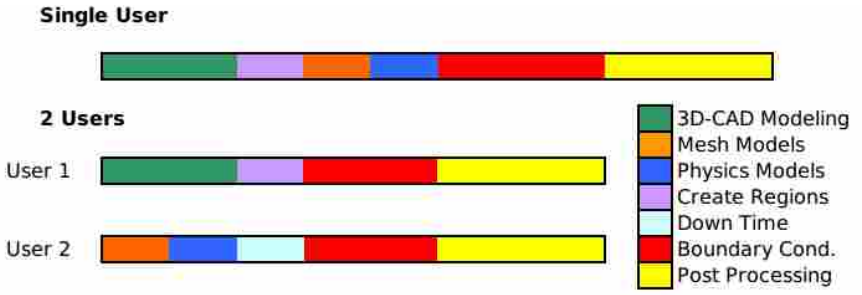


Figure 6.5: Operation breakdown for a single user and two users in the set-up of the Cooled Turbine model.

test. This was due to the necessity of changing the order of surface preparation operations as a way to avoid downtime. Downtime in the model set-up occurred as the interfaces and mesh size at those interfaces were selected on this model (see figure 6.6). User 2 had no other operations to perform until those interfaces were created.

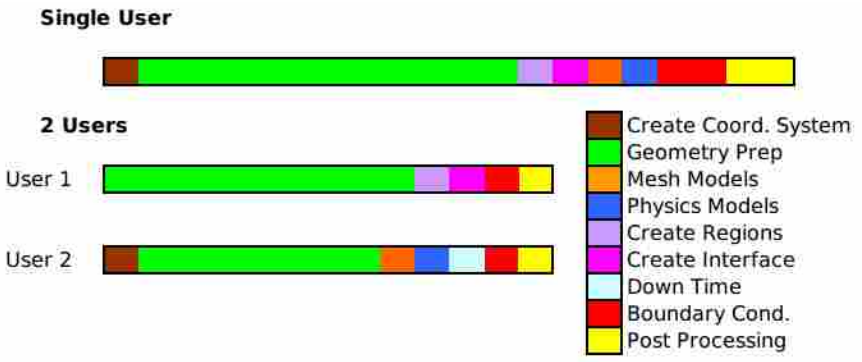


Figure 6.6: Operation breakdown for a single user and two users in the set-up of the Crescendo Case model.

Figure 6.7 shows how more man hours were spent on geometry preparation in a multi-user set-up of the AFRL Film-Cooled Turbine Vane model. This was due to the slow transfer speed across the network connecting both clients which increased the time that an operation would require. There were also issues with serial processes as one user would finish the geometry preparation before the other, and would have to wait until all the geometry parts were ready to be made into regions.

These time savings would be even greater if the testing included time required to refine mesh in areas, update physics parameters, or further repair the surface geometry, as would be done

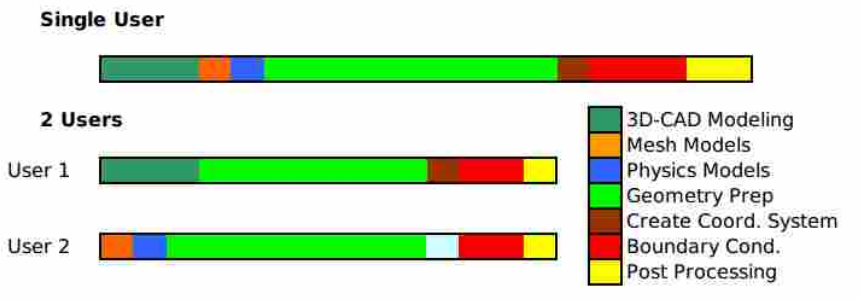


Figure 6.7: Operation breakdown for a single user and two users in the set-up of the AFRL Film-Cooled Turbine Vane model.

multiple times on complex CFD models, however, those variables were not included in the testing discussed here. Because the same two users participated in both the single-user and collaborative simulation set-ups, the effect of collaboration on planning could not be included in these results. Prior to the multi-user set-up, these users had already planned how to set-up the model for a single user, and examining the planning phase of partitioning the model between the two users would not have accurately represented the planning required for these simulations. Further testing with an increased pool of experienced users would allow that to be tested, and would likely show a positive effect of collaboration on planning.

Additionally, these time savings would be improved if the operations that presented limitations to collaboration were improved to allow synchronous execution. The speedup for these models would get closer to $2\times$, however it is unlikely that this speedup would ever actually be achieved. This is due to the same reasons already discussed: downtime caused by difficulty in partitioning the model, and the time spent moving into a new step of the set-up process that does not follow the normal sequence of a single-user set-up. The time required for multiple users to perform operations such as geometry preparation would be reduced by helping eliminate the intermittent bottlenecks caused by limited capabilities in the operations, however, bottlenecks that result from the need to complete a step before moving to the next will not be eliminated with the improvement of the collaborative capabilities. Even if all operations could be performed synchronously with both the same and different operations, there would still be bottlenecks caused by the single-user serial nature of the steps followed in a CFD model set-up. For this reason it is unlikely that a speedup of $n\times$ will ever be achieved.

6.1.2 Model Improvement

Model improvement is looked at in three ways: changes to a model that result in greater solution accuracy, the ability to further analyze the model as a result of time saved in model set-up, and the constant monitoring of the model that comes from having multiple users examining the model set-up. The results show the way in which the model can be improved to better accomplish the purposes of its design, such as changes to the model geometry. If each iteration of the model requires less user time, this process can be done more times, and the model can be improved.

The Surface Wrapper in STAR-CCM+ is often used on complex models as a way to decrease the amount of geometry preparation required and save time. However, on conjugate heat transfer models the surface wrapper decreases model accuracy, and prevents a conformal mesh from being generated on the interface between the fluid and solid regions. Because mesh generation cannot be executed by multiple users in collaboration, the benefits of complex models tested used Surface Repair and boolean operations rather than the Surface Wrapper for geometry preparation. The accuracy of a conjugate heat transfer model is greater with a conformal mesh. The Crescendo Case model was originally used to demonstrate the capabilities of the Surface Wrapper meshing model, and has previously been set-up that way. The model set-up for the testing of collaboration involved creating a conformal mesh, which increased the accuracy of heat transfer between regions. Because the operations for the Crescendo Case model set-up could be accomplished in collaboration, time was able to be saved while also producing a more accurate solution.

The complexity of CFD geometry preparation is increased by the steps of region and boundary definition, mesh set-up, assignment of physics models, and setting initial conditions. If any of these steps are missing or done incorrectly the simulation could fail to mesh or solve correctly, or could give an incorrect solution. In this case collaboration can help by giving users a way to double check the work of other users, and by requiring fewer steps that could be forgotten by each user. This requires that the users understand their role and tasks in the plan, and if an exact plan is not being followed, it requires good communication between each user to ensure that no steps are being missed.

Another benefit of collaboration is the training of users new to STAR-CCM+. In the case of all tests other than the AFRL Film-Cooled Turbine Vane model, user 2 began this research without any prior experience using STAR-CCM+. On complex models this gave the less experienced user

some exposure to geometry preparation without requiring them to do all the complicated aspects. User 1 was able to examine and validate the work of user 2 while each user still was able to accomplish their specific tasks. User 2 was able to also perform new and difficult tasks while knowing that any errors in the model would be corrected.

6.1.3 Catching and Repairing Errors in the Model

Having more users observing the model set-up proved beneficial to a multi-user set-up. This was particularly noticed on the AFRL Film-Cooled Turbine Vane model. On each of the set-ups completed by two users, errors were caught that might have been missed otherwise, or would have had to be fixed later, resulting in lost time. The process of duplicating and rotating the initial vane to form the full annulus was a simple task, but was complicated by the size of the model. The method that was planned due to simplicity was to first create all duplicates, and then to rotate the vanes together. By rotating all vanes together, a single rotation angle could be used for each new position. All except the original vane were selected and rotated the first time, and then each new rotation was done with one fewer vane, until the full annulus was created. The operations of duplicating and rotating took several seconds each and this is one area were difficulty arose. If the process of rotating the vanes was not completed when the dialog box to rotate parts was closed, the rotation would not occur. Because of the model size, the new parts created during the duplication process were not shown, as this would have slowed down the server transfer speed more. This made it more difficult to recognize those errors as they occurred, and during the first multi-user test both users had to repair errors where vanes were missing from the annulus. These errors were caught both times by the other user. Similarly, errors occurred when the coordinate system that was created at the center of the annulus was not selected, and the newly rotated part was rotated to an incorrect location.

On the second multi-user test of the AFRL Film-Cooled Turbine Vane model, the initial surface repair on one of the sections of vanes did not repair all errors. This went unnoticed by user 1 who was performing the operation, due to focusing only on the operation, and not to the results of that operation. This is something that can occur when users are hurrying to finish their tasks both in collaborative and single user set-ups. User 2 recognized this error and it was fixed, otherwise it would not have been noticed until the failure of the mesh generation.

New users to CFD can be overwhelmed by the steps of CFD and steps are sometimes missed. Surface preparation was one of the areas that presented difficulties for the less experienced user because the problems encountered were different each time. For a multi-user simulation these errors were recognized sooner with more users paying attention to the model, rather than being noticed after the mesh has been generated or the model has been solved. For these models, splitting up the model into parts and steps for two users at times resulted in more errors than a single user set-up, but these errors were recognized in each test, and the users were better able to work together to fix the errors.

6.1.4 User Impressions

Each model presented specific operations to a collaborative set-up that made them unique. This led to varied results in terms of the time savings (see table 6.1) and in the process and planning that was required for each. The set-up process for each model was described in chapter 4, but it is important to understand what was learned from specific models and how it contributed to the collaborative experience. The Crescendo case and Cooled Turbine models showed one of the greatest and least time savings. In the opinion of the users those two models respectively showed the greatest and least effectiveness of collaboration, but it isn't enough to deterministically assume that the models with greater time reduction will always be considered more effective for collaboration. It is more important to judge collaborative effectiveness based on user impressions, and to examine how the collaborative effectiveness of a model resulted in the observed time reductions.

The Control Valve model showed that despite being the most simple model used in these tests, the time reduction was not the greatest. The model set-up required several processes that depended on previous processes, and the resultant down time was never able to be made up with effective collaboration because the set-up was so easy. Compared to the other models in these tests, the users were less able to find a flow in collaboration, and this is primarily due to the simplicity of the model and lack of problem solving required.

The Multi-Region Heat Exchanger model supported collaboration due to the two regions it contained. Because each user was able to work in separate regions, there was never any down time caused by bottlenecks in the set-up. The 35% time reduction seems to contradict the hypothesis that complex models will see greater benefits from collaboration. The reality of this set-up is that it

is not truly representative of a typical CFD model, and the model was designed to not require any surface repair. The simplicity does not hold back collaboration as with the the Control Valve model, rather it was looked at as a sort of upper limit for time reduction for a full model set-up. A more complex model might see more benefit and time savings from finding a flow and solving problems in collaboration, but those would likely be offset by bottlenecks in times when collaboration is stalled. This was shown with more complex models in which collaboration was very effective, but the time reduction was similar.

The RAE2822 Airfoil model had the greatest time reduction, but only represented a small part of the simulation set-up. Because the model was imported as a meshed volume, no geometry preparation was required. This model was used to show that collaborative CFD could be beneficial on aerospace models.

The example with the least improvement resulting from collaboration was the Cooled Turbine example. A difference of this example is that the initial set-up of the geometry was done in the 3D CAD Modeling mode. The domains were set up in this mode from an imported parasolid file. The volumes were extracted without splitting and duplicating the surface and without using surface repair at all. While in 3D CAD Modeling mode only a single user was able to prepare the geometry. At this time only the continua set-up did not depend on the geometry that had not yet been created. The multi-user collaboration in this simulation was primarily done once the regions were created, and multiple users could set up the boundary and initial condition, reports, and scenes. Because the geometry preparation for this model was performed in 3D-CAD modeling mode, it became a major bottleneck in the collaborative process, while it should ideally be the area in which collaboration could have the greatest effectiveness.

Being the most complicated model, the Crescendo Case simulation highlighted the important aspects of collaboration, as well as highlighting where there can be hold-ups due to lack of planning. The bottlenecks can often be limited if the user is able to look ahead to other parts of the set-up that are not limited by the steps of the other user. Examples of these include setting up the mesh and physics continua, reports, and scenes. An important result of communication between the users was that Surface Repair mode was able to be used by both users at separate times. This was because the process of using surface repair often involved entering Surface Repair mode, performing the operation, and then closing Surface Repair mode to split surfaces. Because the user

frequently entered and left Surface Repair mode for this model, communication enabled the other user to enter and do operations required for that part. Effective collaboration in this model was a result of the ability of the two users to communicate and avoid bottlenecks by coordinating the timing of each operation, which is why the resultant time savings was one of the highest of all tests. This model is considered most effective because it also demonstrated a full set-up as previously defined in chapter 4: with model import, geometry preparation, boundary definition, and the post-processing set-up. Although this model had less time reduction than the RAE2822 Airfoil model, it was more representative of a full CFD set-up.

The AFRL film-cooled turbine vane model showed less time reduction in model set-up than the other complex conjugate heat transfer models. This was primarily a result of the amount of time required for the server to completely execute the operations. The server execution could not be sped up through collaboration, and at times the transfer speed across the server from client to client became a bottleneck. At times the surface repair portion of the model set-up took longer for two users than for a single user. This was due to the network transfer speed that was maxed out at times between the two clients. Despite the bottlenecks in transfer speed, the multi-user tests always had a time reduction over single-user tests. This time reduction could likely be improved with a higher network transfer capabilities. What was noticeable is the amount of communication that occurred during the set-up of this model. Less planning was initially required for this model because the user had the time during those bottlenecks to think about, and discuss the next steps. This is the kind of collaboration that improves the model set-up, and enables a quicker set-up in areas that are not limited by the server speed.

Overall what was noted by the users is that collaboration in CFD is effective and this effectiveness can be seen both quantitatively and qualitatively. These benefits are desirable because of how much time and work is spent on the set-up of CFD models [4].

Each model set-up and the results provided the users with impressions about how collaboration could be improved, as well as a framework of requirements for an effective collaborative CFD program. Any CFD program using collaboration must contain the following capabilities:

1. Allow collaboration in all methods of geometry preparation

2. Allow pre-processing and post-processing operations to be performed synchronously with other operations

3. Allow pre-processing and post-processing operations to be performed synchronously with the same operation

4. User ability to control their own scene

5. Ability for each scene to show only the parts of the model desired by the user

STAR-CCM+ fulfilled all of these requirements, making it an effective tool for collaboration. There are limitations to the collaborative capabilities and collaborative user experience of STAR-CCM+ and these are identified and documented in chapter 6.1.6. There are also aspects of the CFD set-up process that make collaboration difficult. The main example of this is the inherent parallelism in some CFD steps. This difficulties can be limited through training and better understanding of collaboration, as occurred with the users conducting multi-user tests with STAR-CCM+. While fixing and finding answers to these limitations would increase the effectiveness of collaboration, the five necessary capabilities allow for multi-user collaboration in CFD for STAR-CCM+ and show that it can be effective. All CFD programs will likely have specific best practices, but these five requirements are essential for collaboration.

6.1.5 Best Practices

During the process of working through each of these simulations multiple times there were many tricks and work-arounds learned to avoid confusion between the two users. These are easy to do but would be difficult to learn and would eliminate confusion that, while not negating the capabilities of the multi-user architecture, hindered the user's ability to set up the simulation similarly to a single user set-up in terms of user interface. Most user interface issues take place in the viewing of scenes. Two aspects of viewing scenes (separate scenes with individual control, and the ability to specify what parts of the model are shown on each scene) were identified as requirements for a collaborative CFD program. STAR-CCM+ has these capabilities, but the unique requirements for how to use them require an explanation of the best practices of STAR-CCM+.

The separate clients are not differentiated in the scenes therefore one user will often be forced to see what the other user wants to see. The scenes in STAR-CCM+ allow surfaces or parts to be viewed and interacted with. Curves, surfaces, and volumes can be selected and edited in many ways similar to how they are done in the simulation tree, and when they are selected they are highlighted and labeled with the name of the surface and associated part. When multiple users are viewing the same scene the last part to be selected in either the scene or the tree will be the one highlighted and labeled on the scene of all users. A method of avoiding this confusion is to turn off labeling and highlighting, but this limits the interaction with the scene. The way to fully utilize the highlight function and to avoid confusion is to have the users working in separate scenes. If each user only has their own scene open this confusion will not happen. When a new scene is created in STAR-CCM+, it is created and shown on all clients. It is important for the users who are not working on this scene to close it on their client, rather than just switching back to their desired screen.

However, this can also be an important tool in collaboration. Any user viewing the working scene of another can see exactly what that user can other than the cursor. This would help the user when there is a question about that part of the model, and to avoid errors.

Star-CCM+ also has the capability of the user selecting specific surfaces to view while hiding the others. The issue with this occurs if multiple users are working in separate scenes and on separate parts with all other parts hidden. When a user splits a surface there are new surfaces created, and these will appear in the scene of all other users because those specific surfaces have not been hidden. This could cause an issue in that the part could be deleted or unnecessarily edited by the user who was not working on that specific part. This can be avoided by specifying only the part that each user is working on in the part displayer of the scene, and deselecting all others. This way any surface that is created from another part will only be associated with that part, and not the scenes of the parts being worked on by other users.

The most important practice for a collaborative CFD set-up is communication. This is not specific to any operation capability or to STAR-CCM+. These tests have shown that collaboration in CFD is effective, and the client-server architecture of STAR-CCM+ allowed this to be tested. The limitations and best practices described in this chapter will not apply to all programs or all models, but had to be considered in the planning for these models. Planning and communication

allow for a relative plan, that can be changed as needed based on any limitations of CFD operations, and on any model.

6.1.6 Limitations of STAR-CCM+ Multi-User Capabilities

The operations that were identified as having the greatest proposed benefit to collaborative time savings are mesh generation, 3D-CAD Modeling mode viewing, and surface repair. They were reported to CD-adapco and recommended to have multi-user capabilities enabled.

Mesh generation is generally performed after the domain boundaries and initial conditions are set up and before the post-processing is prepared. Mesh generation is the only operation in STAR-CCM+ that can not be performed synchronously with any other operations. Problems with this limitation are avoided because meshing is mainly considered computational time of the simulation set-up rather than user time. The model is often sent off to a more high powered computer where the mesh is generated after much of or all the set-up work of the model has been completed. At this point the user is able to work on set-ups of different models. However, the advantage would be seen with a complex model of multiple stages. In this case separate users could mesh separate parts simultaneously to be able to accurately see the mesh parameters that need to be used for that part. In simulations that do not require a conformal mesh between regions, this capability would result in greater ability to analyze and edit specific parts of the mesh that need to be improved.

Additionally improvements can be made to the mesh generation process that enables the simultaneous use of other non-meshed based operations. This capability would enable the users to set up boundary conditions, pre-processing reports and solvers, and scenes while the mesh is being generated. The user time involved in setting up these parts of the simulation would remain the same, but part of the computational time of the mesh generation would be used elsewhere.

The Cooled Turbine and AFRL Turbine models which required use of the 3D-CAD Modeling mode highlighted the limitations of collaboration within this mode. Although multiple users were able to enter either the same or different 3D CAD models within STAR-CCM+ simultaneously, only one scene could exist for all clients. This resulted in each user seeing the highlighted parts of other users, and requiring the same part orientation to be used for all clients even if it was not the same part being shown. If users attempted to select different surfaces or parts while in this mode it became confusing and difficult. A major improvement could be made by allowing the

creation of multiple scenes within the 3D-CAD Modeling mode. This would allow users to work independent of the other users view simultaneously while preparing the geometry.

Surface repair can often be a part of the simulation set-up that is most intensive for user time. A clear example of this is creating boundaries to fill holes for inlets and outlets. In a more simple operation on a well-defined edge, the hole can be filled using surface repair by double clicking any cell edge of a continuous edge and selecting "fill hole". This is slightly more difficult when using Boolean operations. The hole can be filled by creating part curves on the surface, one of which must be the curve of the fill boundary. The part curves must be split by contiguity, and the name of the desire curve noted. Now the hole can be filled in the parts based operations by selecting the part, surface, and desired curve. The resulting surfaces cannot be used to create a closed domain without using the surface repair node.

A great improvement would be the ability for multiple users to simultaneously enter into Surface Repair mode on separate parts. This would go along with the partitioning of the model for each user and enable users to create surfaces for inlets and outlets where it is needed in that domain, and to do the necessary repair to ensure that the regions are closed and manifold. The resultant time savings from this addition would be great because of the amount of time that is spent repairing surfaces compared to setting up the rest of the simulation. This would result in less downtime for a user waiting for the other to finish surface repair, and eliminate the need to extensively plan the order of boolean operations as to ensure that each part and surface to be combined are closed an manifold.

These changes would increase the effectiveness of collaboration within STAR-CCM+ by allowing more work to be done where there are currently existing bottlenecks of the simulation set-up, and by making the partitioning of the multi-user work flow, and model views more intuitive with fewer required work-arounds.

6.2 STAR-NX Results

The STAR-NX model of the car in wind tunnel was tested with two users working in both NXConnect and STAR-CCM+. The results demonstrated are the capabilities and limitations of the multi-user client developed for NXConnect.

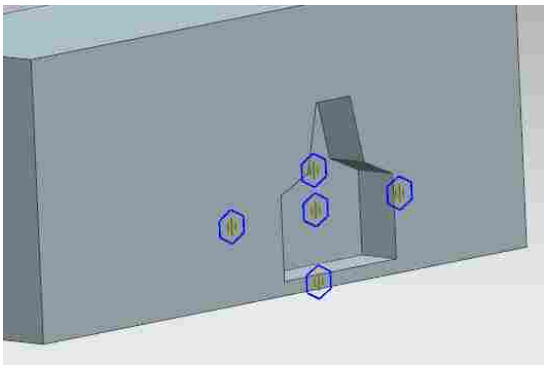
6.2.1 STAR-NX Capabilities

The testing of STAR-NX showed that the multi-user client was capable of updating the geometry on each NXConnect client. As was the goal of the Multi-User CAE tool, geometry changes can be updated through STAR-NX in both NXConnect and STAR-CCM+. In both cases the geometry will be updated on all clients. Figure 6.8 shows the original geometry in two NXConnect clients (figures 6.8a and 6.8b) and one client of STAR-CCM+ (figure 6.8c). Both NXConnect clients and the STAR-CCM+ client have the same geometry of the wind tunnel and car. STAR-NX was used to updated the wind tunnel height and length and the results are shown in figure 6.9. The geometry is updated on both NXConnect clients (figures 6.9a and 6.9b) and in the STAR-CCM+ client (figure 6.9c). All clients in both programs show the correct updated geometry.

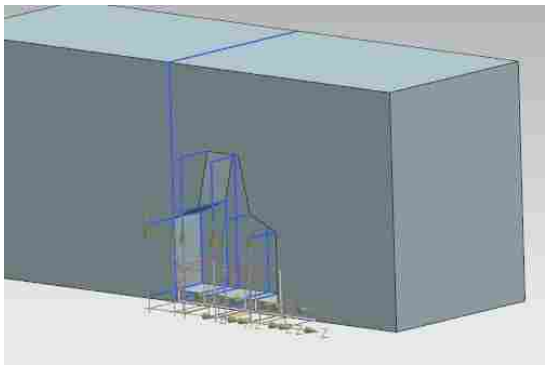
6.2.2 STAR-NX Limitations

One limitation of STAR-NX is with multiple clients of STAR-CCM+. If multiple users are working on a model set-up in STAR-CCM+ then each user would ideally be able to make parameter changes to the model. In the same way with a multi-user CFD set-up not using STAR-NX, each parameter change is immediately updated to the properties window of all other clients. However, only the client originally created through STAR-NX (client 1) is able to push the updates to NXConnect. Nothing will happen when the other clients attempt to push the changes.

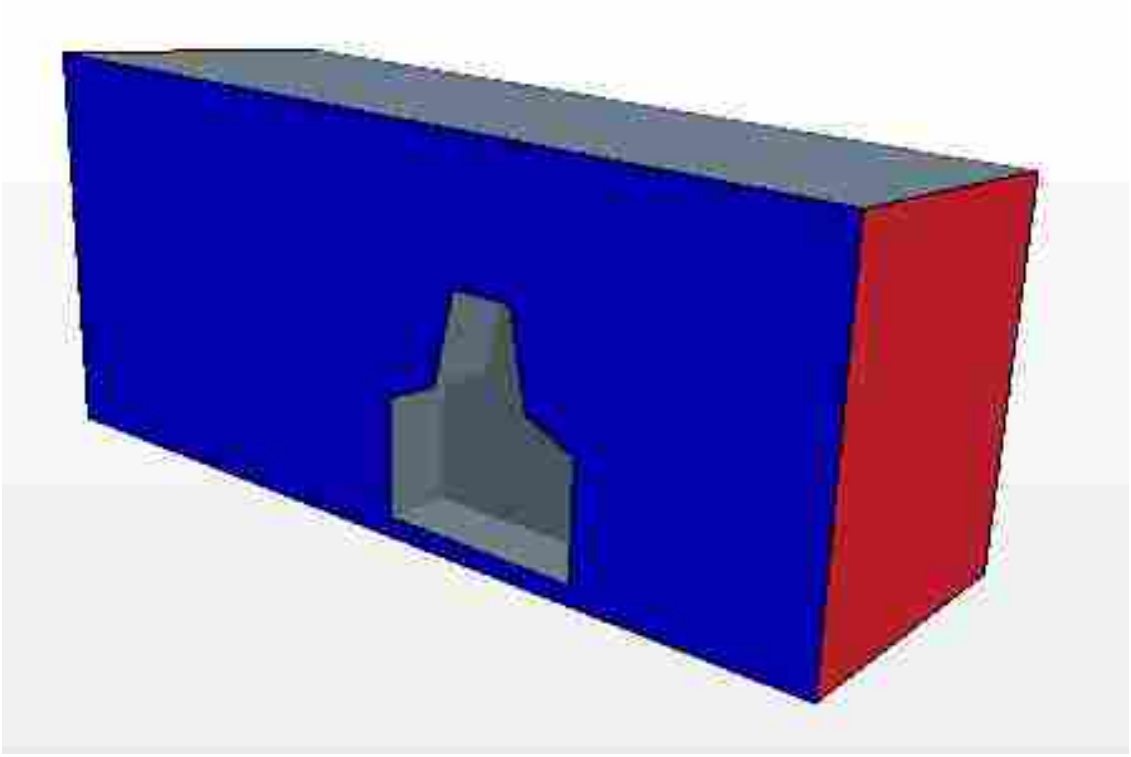
Another limitation is that NXConnect does not support a multi-user CFD set-up within the NX environment. This is due to the lack of API for STAR-NX, as it is not revealed by CD-adapco. The effect is that each NXConnect client cannot connect to a single STAR-CCM+ server. Although the geometry will be consistent for all clients of both programs, the CFD set-up will be different. Each client of NXConnect will have a different server of STAR-CCM+ that will be created when CFD set-up is done on that client. This limitation can be a problem if multiple users are trying to collaboratively set up a CFD simulation through NXConnect. However, this can be worked around by transferring CFD data through a single NXConnect client, and doing the collaborative CFD set-up with multiple STAR-CCM+ clients. This method has the benefit of having more CFD set-up capabilities in STAR-CCM+ than STAR-NX.



(a) NXC Client 1 Geometry

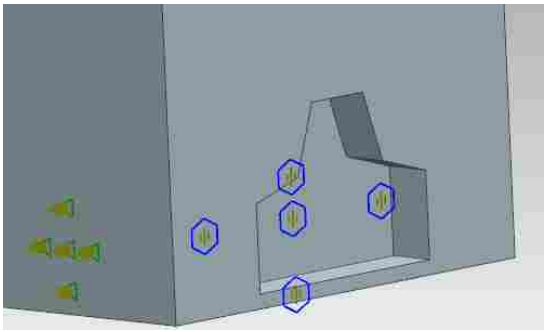


(b) NXC Client 2 Geometry

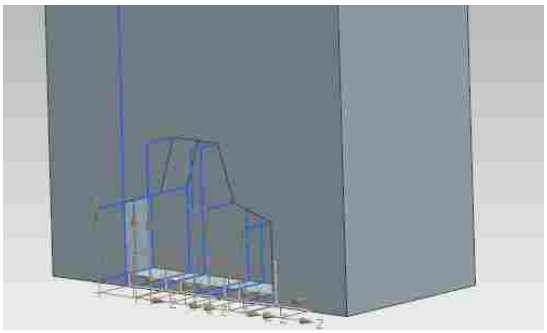


(c) STAR-CCM+ Geometry

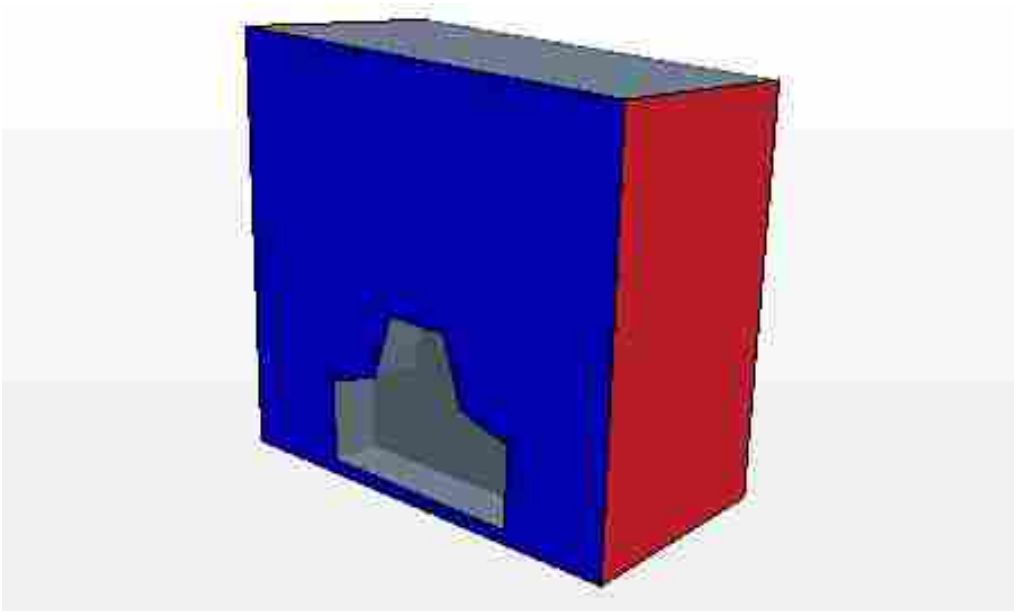
Figure 6.8: Original model of car in wind tunnel



(a) NXC Client 1 Geometry



(b) NXC Client 2 Geometry



(c) STAR-CCM+ Geometry

Figure 6.9: Changes made to model via STAR-NX

CHAPTER 7. CONCLUSION

Using STAR-CCM+, multi-user collaboration in CFD was demonstrated and evaluated for the first time. Six models were used for testing the collaborative benefits of a multi-user set-up. The model set-up involved preparing geometry for meshing, defining boundary types and properties, specifying physics and meshing models, and generating post-processing reports and visualizations. The set-ups for each model were performed multiple times by two users individually and two users in collaboration. The benefits of collaboration were demonstrated and evaluated using these six models. The models had varying degrees of set-up complexity to test all capabilities and show the types of models that are best suited for collaboration. The integration of multi-user CAD and CFD was improved and tested with STAR-NX.

This evaluation of collaboration in CFD and integration of CAD and CFD achieved three key things for the first time. First, the benefits of collaboration were shown that would benefit any model. Second, STAR-CCM+ was tested as the only CFD program currently supporting collaboration, and the results of these tests were reported to CD-adapco as a way to improve the program. Third, the integration of CAD and CFD was enhanced with the ability to link collaborative engineering programs.

7.1 Importance of Collaboration

The benefits of collaboration have been seen in engineering [1], but the engineering software to fully realize these benefits does not exist. Collaborative CAE software has been developed at BYU to improve collaboration [2]. The difficulty in CFD of preparing geometry prior to meshing as discussed by Slotnick et al. [4] is identified as an area that can be improved through multi-user collaboration and the integration of CAD and CFD. Collaboration on CFD models had not been tested previously due to the lack of programs that enable collaboration. The client-server architecture and collaborative capabilities in STAR-CCM+ allowed testing of collaboration in CFD.

7.2 Using Collaboration in CFD

The testing of these CFD models has revealed that the set-up of CFD models can be improved through multi-user collaboration. It has also shown that there are types of models that would benefit most from these improvements. The analysis of collaboration in CFD included documenting the observed benefits, the impression of the users who participated in this testing, and identifying the types of models that would receive the greatest benefit from collaboration. The collaborative capabilities of STAR-CCM+ allowed for the testing of these models, but any CFD program with similar capabilities would show the same benefits. Collaboration in any CFD program will similarly reduce time spent on simulation set-up, improve models, and help catch errors in the model set-up.

A good collaborative CFD code was defined as one that allows for collaboration in all set-up operations, and user control of their individual scene. According to these requirements, STAR-CCM+ is a good and effective CFD program for multi-user collaboration. Despite its limitations and the learning curve that must be climbed to fully understand the work-arounds to these limitations, STAR-CCM+ has all these necessary capabilities. The development of another collaborative CFD code must include these capabilities.

7.2.1 Observed Benefits of Collaboration in CFD

Three benefits were observed for a collaborative CFD set-up: time savings, model improvement, and the ability to catch errors due to more users viewing the model. Time savings between 24.4% and 36% were observed. The time savings were best observed on complex conjugate heat transfer models like the Crescendo Case model, which required a lot of geometry preparation and problem solving to avoid bottlenecks. Model improvement was seen on models where time savings would normally be prioritized over solution accuracy. The ability to prepare geometry with the Surface Repair mode and boolean operations in the Crescendo Case tests allowed the creation of a conformal, continuous mesh across the interface of multiple boundaries, resulting in more accurate heat transfer solutions across boundaries. The ability to recognize and fix errors in the model set-up were seen in the set-up of the AFRL Film-Cooled Turbine Vane model. The complexity of this model resulted in unexpected errors during surface repair. However, because both users were

observing the geometry to ensure that it was ready for meshing, these errors were noticed, and the users were able to work together to fix them.

7.2.2 User Impressions

Two main observations were made during the model testing. A flexible plan allowed users to creatively work around bottlenecks that occurred during a collaborative set-up, and communication was essential to altering that plan as needed. The Crescendo Case model showed this best as bottlenecks decreased and more time was saved in each of the three plans made for model set-up. The users were only able to estimate the amount of time that each operation would take, and at times this resulted in user downtime which was observed differently in each simulation set-up. The use of a flexible plan and communication are recommended practices for collaboration in CFD and will allow the users to find solutions to these bottlenecks. The methods of communication in these tests showed how the benefits of collaboration could still be realized when the collaborating users are geographically far apart. Communicating though Google Chat is a way that users in different areas could communicate, and required effective communication, as things must be explained clearly and succinctly but without the benefit of being able to directly point to any part of the model.

7.2.3 Ideal Models for Collaborative CFD

It is clear from the multi-user testing that some models and simulations lend themselves better to collaboration than others. Complex models such as the Cooled Turbine, Crescendo Case, and AFRL Film-Cooled Turbine showed these benefits most. The amount of geometry preparation required allowed the collaborating users to reduce time in areas where the most time was generally spent. Conjugate heat transfer models will benefit from collaboration due to the existence of multiple regions, which would result in greater freedom in the model set-up for each of the users. Large models will also benefit from collaboration. A model of a jet engine with multiple parts would allow users to work on separate areas of the engine. Geometry preparation could be performed by multiple users on these separate areas, and closed and manifold volumes could be created more quickly.

A pattern observed during the testing of these models is that specific steps in the simulation set-up process were more greatly benefited by collaboration. Geometry preparation and the boundary set-up of separate regions was most easily partitioned to steps for multiple users. As such, any model requiring geometry preparation and containing multiple regions will benefit most from collaboration. The steps to set up the boundaries of a single region, and the use of 3D-CAD Modeling mode were least benefited by collaboration, and models containing these these as a significant part of the set-up process would have reduced collaborative benefits.

7.3 STAR-CCM+ for Collaborative CFD

The ability to test the benefits of collaboration in CFD was possible because of the client-server architecture of STAR-CCM+. STAR-CCM+ is the only CFD program that has these capabilities, and until other programs are created to enable collaboration it will be the only CFD program where these benefits can be observed. Testing of these six models showed the capabilities and limitations of the operations in STAR-CCM+, which were previously unknown to even CD-adapco. The testing on these models using STAR-CCM+ proved the collaboration in CFD is capable, and that the benefits of collaboration are clear. The collaborative capabilities of STAR-CCM+ allowed a full CFD simulation to be set up with multiple users, and with few exceptions, nearly all operations could be completed synchronously by multiple users.

7.3.1 Limitations of STAR-CCM+ Collaborative Capabilities

The operations with collaborative limitations include the ability to mesh with multiple users at the same time, use Surface Repair mode simultaneously by multiple-users, and work in multiple scenes within the 3D-CAD Modeling mode. The benefits of a multi-user set-up are shown despite these limitations, but they must be taken into account when preparing how users can work together for a set-up that includes these operations. The collaborative process would be more effective if the operations containing these limitations were improved to have greater synchronous capabilities resulting in greater time reduction.

7.3.2 Best Practices

Specific to the limitations of STAR-CCM+ the best practice was to use boolean operations and Surface Repair mode together during geometry preparation. These limitations influenced the process of collaborative model set-up as methods were implemented to avoid bottlenecks. The use of different scenes for each user prevents confusion caused by the highlighting of parts selected by the other user. Separate scenes give the users control of what they want to see. Bottlenecks caused by the use of 3D-CAD Modeling mode can be decreased when it is used synchronously with operations to set up the meshing models and physics models.

7.3.3 Recommendations

The limitations of collaborative capabilities were documented and reported to CD-adapco in addition to the operations that would have the greatest impact from the improvement of collaborative capabilities. Mesh generation could be improved by allowing other non mesh based operations to be performed during mesh generation. Surface Repair mode could be improved by allowing multiple instances to be opened synchronously on different parts. The ability to open multiple scenes within 3D-CAD Modeling mode would enable multiple users to collaborate on 3D-CAD models but control their own visualization scene. These improvements would increase the amount of time saved by multiple users in a CFD simulation set-up even more than already shown.

7.4 STAR-NX

Integrated CAD and CFD is enabled with STAR-NX. STAR-NX creates a parametric connection between NX and STAR-CCM+, enabling the model to be transferred directly from NX to STAR-CCM+ without the need of a neutral file transfer. It also allows the set-up of CFD properties and operations to occur within NX. Through this integration the time consuming process of geometry preparation and all other aspects of the CFD simulation set-up are not required for each iteration of the design cycle, thus reducing model set-up time.

NXConnect did not support all benefits of STAR-NX. The geometry changes that were made through STAR-NX did not update on all NXConnect clients. The STAR-NX Multi-User CAE tool (MUC) was created to solve this problem.

7.4.1 Creation of Multi-User CAE Tool

The MUC was added to the NXConnect code. Through this addition, all NX expressions are monitored and compared to previous values. If there is a change in the expressions, the MUC will catch that and call existing NXConnect functions that ensure the geometry is updated on the NXConnect server and all clients.

7.4.2 Model Testing of Multi-User CAE Tool

The MUC was tested on a model of a simplified car in a wind tunnel, specifically to show that the model geometry would update correctly. The CAD model was created by three users in NXConnect and the model was transferred to STAR-CCM+ for all three users. The variable parameters of car height and length, and wind tunnel length, width, and height were updated through STAR-NX. The success of the MUC was demonstrated as all clients of both programs were updated with the correct geometry.

7.4.3 Future Work With STAR-NX

The research in this thesis focused on developing the MUC and testing to ensure that it works as described. Future work in the testing of the STAR-NX MUC would show the time improvements that occur in a model that has multiple geometry changes, and the time saved by not requiring geometry preparation for each of these iterations.

REFERENCES

[1] Bercovitz, J., and Feldman, M., 2011. "The mechanisms of collaboration in inventive teams: Composition, social networks, and geography". *Research Policy,* ***40***(1), pp. 81–93. 1, 5, 37, 75

[2] Red, E., French, D., Jensen, G., Sillito Walker, S., and Madsen, P., 2013. "Emerging design methods and tools in collaborative product development". *Journal of Computing and Information Science in Engineering,* ***13***(3), pp. 031991–1–031001–14. 1, 6, 8, 75

[3] Briggs, J. C., Hepworth, A. I., Stone, B. R., Coburn, J. Q., Jensen, C. G., and Red, E., 2015. "A fundamental integration of multi-user cax methods for synchronous design and analysis". *Journal of Computing and Information Science in Engineering*. 2, 7, 57

[4] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D. "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences". 2, 8, 12, 13, 49, 67, 75

[5] Godo, M., 2013. 3D CAD to Meshing - Why This is Critical for Production Level CAE. CD Adapco Webinar, December. 2, 8

[6] Bidarra, R., van den Berg, E., and Bronsvoort, W. F., 2002. "A collaborative feature modelling system". *Journal of Computing and Informational Science in Engineering,* ***2***(3), pp. 192–198. 5, 7, 11, 12

[7] Red, E., Jensen, G., French, D., and Weerakoon, P., 2011. "Multi-user architectures for computer-aided engineering collaboration". *17th International Conference on Concurrent Enterprising*, pp. 1–10. 6, 8, 16, 26

[8] Borrmann, A., Wenisch, P., van Treeck, C., and Rank, E., 2006. "Collaborative computational steering: principles and application in HVAC layout". *Integrated Computer-Aided Engineering*, pp. 1–16. 8

[9] Chandra, S., Lee, A., Gorrell, S., and Jensen, C. G., 2011. "CFD Analysis of PACE Formula-1 Car". *Computer-Aided Design and Applications,* ***8***, pp. 1–14. 8

[10] CD-adapco. *Star-CCM+ User Guide*, 8.8.6 ed. http://stevedocs.cd-adapco.com/ViewDocs/authdocs/starccmplus_latest_en/index.html. 11, 12, 16, 47, 52, 53

[11] Hughes, T., Cottrell, J., and Bazilevs, Y., 2005. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". *Computer Methods in Applied Mechanics and Engineering,* ***194***, pp. 4135–4195. 12

[12] Gujarathi, G., and Ma, Y. "Parametric CAD/CAE integration using a common data model". *Journal of Manufacturing Systems,* ***30***, pp. 118–132. 13

[13] Siemens Industry Software, 2012. *NX 8 for Simulation*. http://www.plm.automation.siemens.com/en_gb/products/nx/nx8/simulation/index.shtml. 13

[14] Cannon, L., Nysetvold, T., Phelps, G., Winn, J., and Jensen, C. G., 2012. "How Can NX Advanced Simulation Support Multi-User Design?". *Computer-Aided Design and Applications,* ***2***, pp. 21–32. 13

[15] Ni, R. H., Humber, W., Fan, G., Johnson, P. D., Downs, J., Clark, J. P., and Koch, P. J., 2011. "Conjugate Heat Transfer Analysis of a Film-Cooled Turbine Vane". *Proceedings of ASME Turbo Expo 2011*(GT2011-45920). 38

[16] Ni, R. H., Humber, W., Fan, G., Clark, J. P., Anthony, R. J., and Johnson, C. J. J., 2013. "Comparison of Predictions From Conjugate Heat Transfer Analysis of a Film-Cooled Turbine Vane to Experimental Data". *Proceedings of ASME Turbo Expo 2013*(GT2013-94716). 43

APPENDIX A. COLLABORATION PLANS FOR MULTI-USER TESTING

For the complex models used in the multi-user testing of STAR-CCM+ and STAR-NX, plans were developed that the two users could follow. The plans included in this appendix are for the Crescendo Case Model described in chapter 4.6, the AFRL Film-Cooled Turbine Vane described in chapter 4.7, and the car model tested with STAR-NX described in chapter 5.4.

A.1 Crescendo Case

The multi-user test for the Crescendo Case model was performed using the following steps.

A.1.1 Crescendo Case Surface Repair

User 1

1. Import Eurocopter model
2. Combine all parts into on
3. Combine all part surfaces
4. Name surfaces for solid, fluid, inlet, etc
5. Create Geometry -> New Shape Part -> Block to cover the gaps in the housing where the hinges are. Intersect the block with the housing part
6. Imprint surfaces of the top of housing and the rest of the housing
7. Intersect all intersecting surfaces and delete the intersected surfaces
8. Delete the inside surfaces of the holes in the housing, and fill the holes

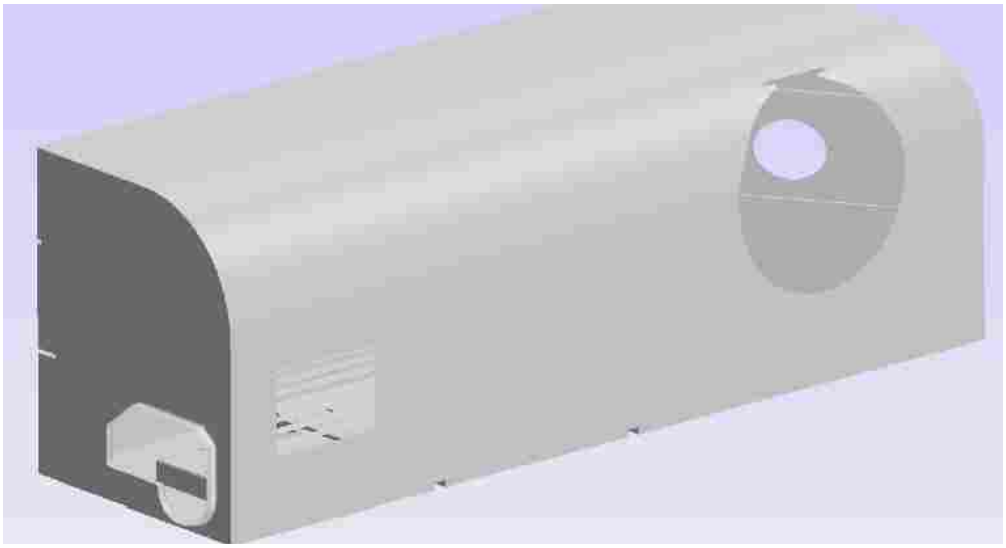


Figure A.1: Internal housing surface of Crescendo Case model

9. Split housing by non-contiguous surfaces into external housing and internal housing (see figure A.1)
10. Create surface for exhaust cowl
11. Fill outlet holes in external and internal housing, and create new parts from these surfaces
12. Fill inside surface of vent holes, and intersect with the vent. Create surfaces for vent outlet, external vent, and internal vent
13. Create surface for inlet of housing

User 2

1. Create Cartesian coordinate system with origin on the corner of the housing as shown in figure A.2
2. Split by patch to create the four surfaces of the exhaust shown in figure A.3. Create a part for each of these surfaces. Name the original part "Housing"
3. Perform Boolean operations –> Merge
4. Split cowl into interior and exterior parts by intersecting with the surfaces that were created to fill the housing outlet holes (see figure A.4)

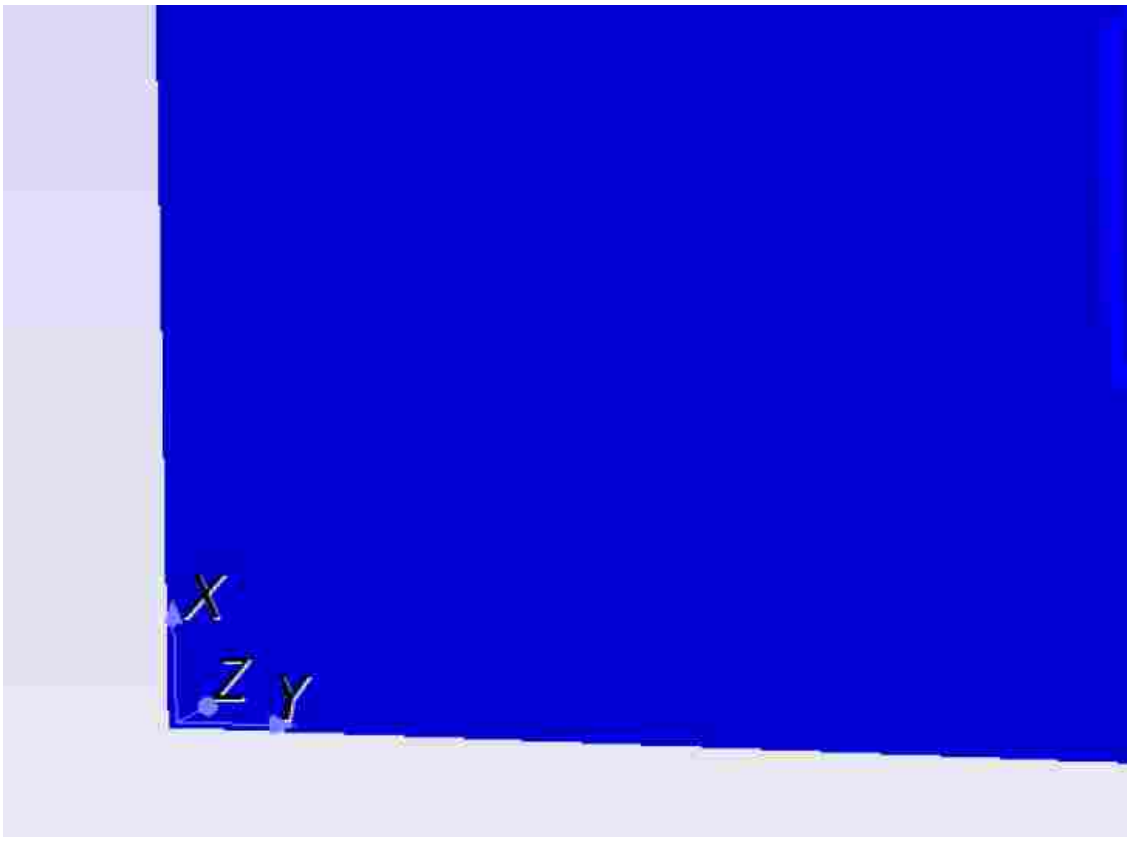


Figure A.2: Created cartesian coordinate system

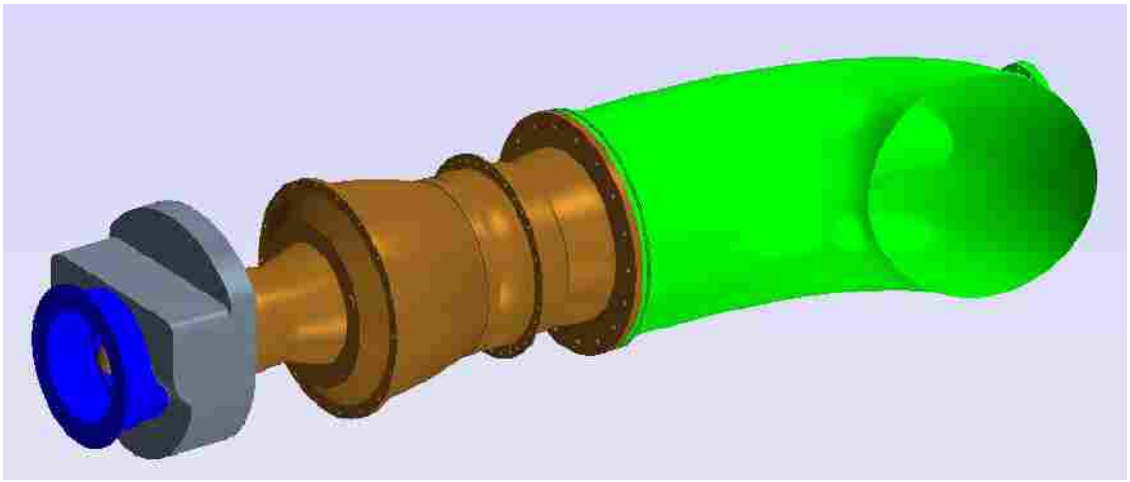


Figure A.3: Exhaust surface of Crescendo Case model

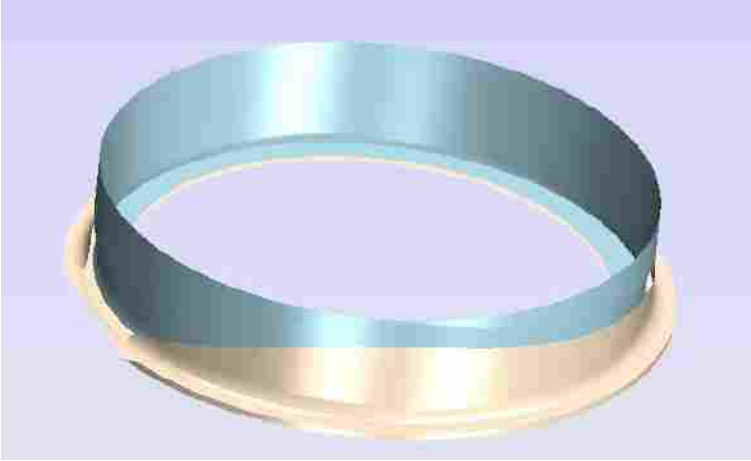


Figure A.4: Conjugate Heat Transfer Simulation (Cooling flow within the turbine blade cools the solid blade)

5. Delete bolts from exhaust part

6. Combine all exhaust parts into a single part, and create surfaces for the inlet and outlet of the exhaust

7. Create feature curves on the edges of the inlet and outlet. Use these feature curves to split the exhaust part by part curves. This will split the exhaust into two parts

A.1.2 Creating Interfaces and Regions

User 1

1. Create 3 parts, one of the solid, one of the inlet and outlet boundaries of the internal flow, and one of the inlet and outlet boundaries of the external flow

2. Duplicate the boundaries of the interfaces, and combine them with the fluid parts. To create a separate part for Interior flow, exterior flow, and solid (see figure A.5). Open surface Repair mode and zip the edges for each of these parts

3. Assign parts to regions

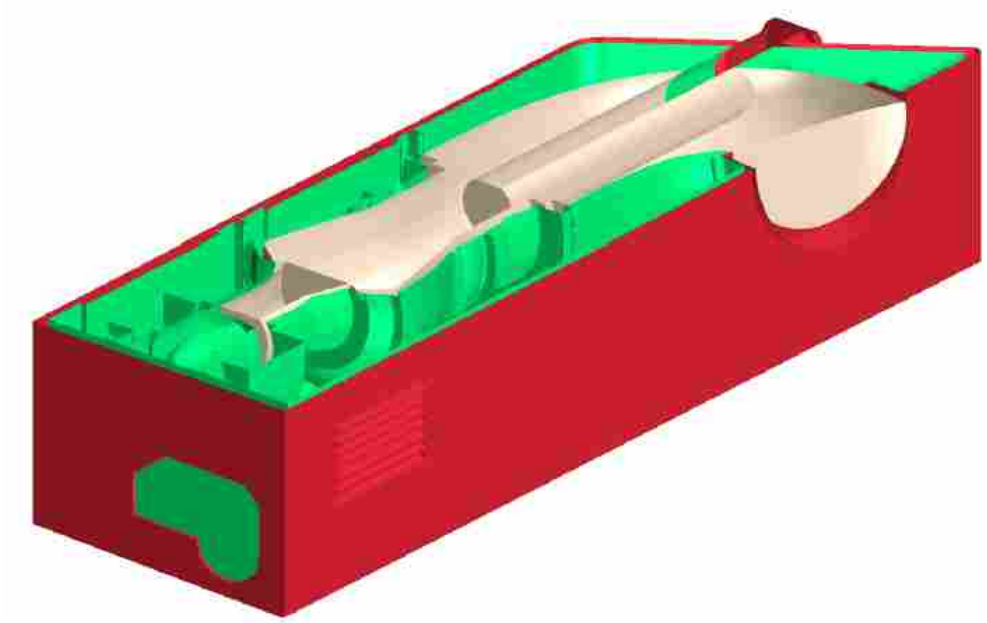


Figure A.5: Three closed and manifold regions of Crescendo Case model

4. Create interfaces for internal flow, and exterior flow. These interfaces are created for boundaries that have the same name. Edit mesh conditions –> Custom Surface Size –> Min: 5% of base, Max: 25% of base

5. Assign boundary types and conditions for internal flow (see figure A.6)

Interior Flow	v, m/s	T, deg. C	p, Pa
Velocity Inlet	5	600	
Pressure Outlet		50	0
Exterior Flow			
Velocity Inlet	2	50	
Pressure Outlet		50	0

Figure A.6: Crescendo case model boundary conditions

User 2

1. Create continua for the mesh, solid region, and two fluid regions.

   (a) Exterior Flow

i. Models: 3D, Steady, Gas, Coupled Flow, Constant Density, Turbulent, K-Epsilon, Coupled Energy, Cell Quality Remediation

ii. Initial Conditions

A. Velocity: 2 m/s in z-direction of created coordinate system

B. Static Temperature: 50 deg. C

(b) Interior Flow

i. Models: 3D, Steady, Gas, Coupled Flow, Constant Density, Turbulent, K-Epsilon, Coupled Energy, Cell Quality Remediation

ii. Initial Conditions

A. Velocity: 5 m/s in z-direction of created coordinate system

B. Static Temperature: 600 deg. C

(c) Solid

i. Models: 3D, Steady, Solid, Coupled Solid Energy, Constant Density

ii. Initial Conditions

A. Static Temperature: 100 deg. C

(d) Mesh

i. Models: Surface Remesher, Polyhedral Mesher, Prism Layer Mesher, Embedded Thin Mesher

ii. Reference Values

A. Base Size: 3 cm

B. Number of Prism Layers: 3

C. Prism Layer Thickness: 20% of base

D. Thin Mesher Layers: 2

E. Thin Solid Thickness: 10% of base

2. Assign boundary types and conditions for solid and exterior flow, using previous figure

A.1.3 Post-Processing

User 1

1. Create report, monitor, and plot for heat flux at inlet and outlet of interior flow region
2. Create new scalar scene with 2 displayers:
   (a) Fluid Temp.
      i. Parts: Fluid derived surface, inlet and outlet of both fluid regions
      ii. Scalar Field:
         A. Function: Temperature
         B. Units: C
      iii. Contour Style: Smooth Filled
      iv. Color Bar:
         A. Color Map: Cool-Warm
         B. Levels: 200
   (b) Solid Temp.
      i. Parts: Solid region, Solid derived surface
      ii. Scalar Field:
         A. Function: Temperature
         B. Units: C
      iii. Contour Style: Smooth Filled
      iv. Color Bar:
         A. Color Map: Blue-Red
         B. Levels: 200
3. Display both Color bars in scene

User 2

1. Create derived part: plane section of all fluid regions at (using Laboratory coordinate system):

   Origin: [0.8357693611146675, -0.08530677741238347, -0.12554570182617114] m,m,m

   Normal: [0.104493050637671, -0.9941591044408834, -0.02699773001075806] m,m,m

   (a) Name the derived surface “Fluid”

2. Duplicate the Fluid derived surface, change name to “Solid”, and change input parts to the solid region

3. Edit clip plane 1 properties in the newly created Scalar Scene (Scalar Scene 1: Attributes: Clip Planes):

   Origin: [0.8357693611146675, -0.08530577731238347, -0.12554570182617114] m,m,m

   Normal: [0.10449305063767188, -0.9941591044408834, -0.02699773001075806] m,m,m

4. Enable clip plane 1

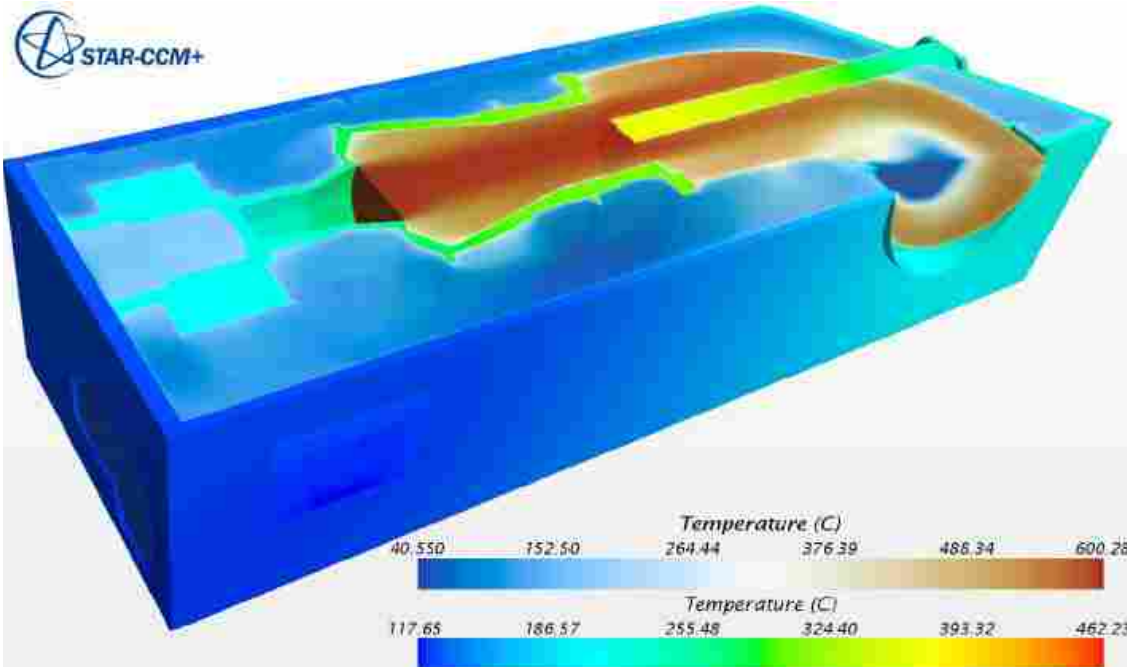


Figure A.7: Conjugate Heat Transfer Simulation (Cooling flow within the turbine blade cools the solid blade)

A.2 AFRL Film-Cooled Turbine Vane

The plan for multi-user collaboration in the set-up of the AFRL Film-Cooled Turbine Vane model is described here.

A.2.1 3D CAD Modeler

User 1

1. Open 3D CAD
2. Import turbine.x_t file
3. Imprint both parts
4. Name surfaces for solid, fluid, inlet, etc.
5. Split surfaces for Interfaces: Cooling1, Cooling2, Cooling3
6. Name surface of all cooling holes
7. Close 3-D CAD
8. Create geometry parts from CAD parts

User 2

1. Create Cartesian coordinate system.

   The easiest way to do this is to create a new Cartesian coordinate system [tools -> coordinate system -> Laboratory -> Local Coordinate System (right click -> New Cartesian)] and then copy and paste the following lines into the property window). Rename "Cartesian 1" to "Rotation"

   [0.0, 0.056213902532657976, 0.9984187484027175] X-Axis

   [0.0, -0.9984187484027175, 0.056213902532657976] Y-Axis

   [1 0 0] Z-Axis

   [-0.02112196944653, -3.594220300405E-7, -1.2470793988432E-7] m,m,m origin

2. Create a new part for the Fluid Region. Geometry -> 3D CAD Model –> Fluid. Use Medium tessellation

3. Select the surface “default” and split surface by patch. Select all the surfaces in figure A.8, and split them with the name “solid”

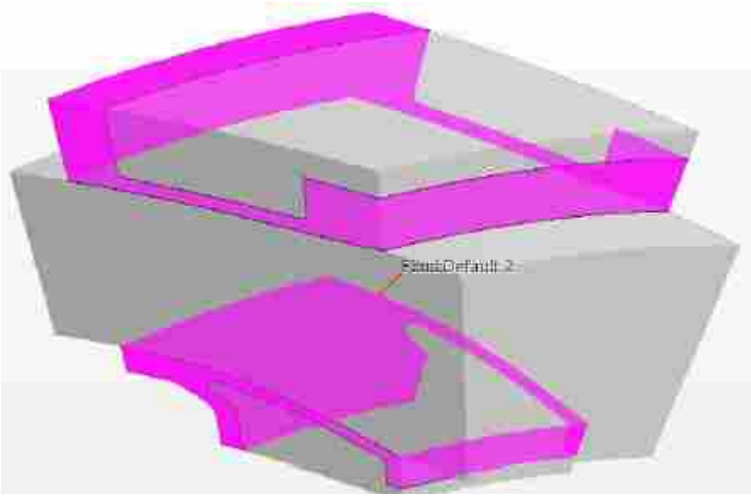


Figure A.8: These parts represent boundaries for the solid region that are not part of the fluid region

4. Delete the new created surface “solid” (shown as “Default 2” in the image above)

5. Split the surface by patch to create the remaining surfaces shown in figure A.9: Cooling–ID, Cooling-OD, Inlet, Outlet, Periodic1, Periodic2, and Fluid

6. Delete both periodic surfaces

7. Create new mesh continuum (surface remesher, polyhedral mesher, prism layer mesher)

   (a) 4 prism layers: 25% of base

   (b) 0.005 m base size

   (c) Surface growth rate of 1.25

8. Create new Physics Continuum for Fluid (3D, steady, gas, coupled flow, constant density, turbulent, k-epsilon, coupled energy). Use the following Initial Conditions:

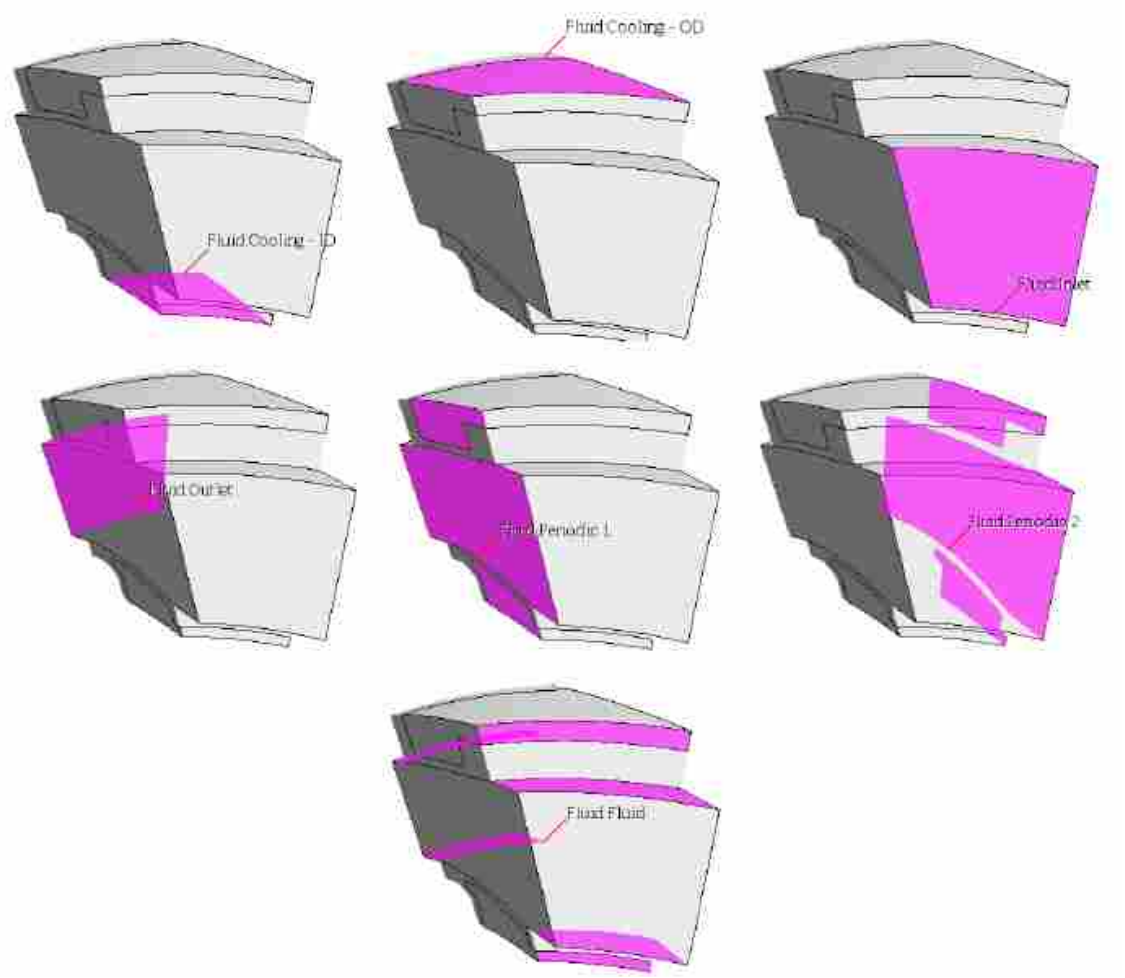


Figure A.9: AFRL Cooled-Turbine surfaces created in 3D CAD Modeling Mode

(a) P = 65.2336 psi

(b) Static Temp = 788.781 R

9. Create new Physics Continuum for Solid (3D, steady, solid, coupled solid energy, constant density). Use the following Initial Conditions:

   (a) T = 550 R

10. Check if User 1 is finished with 3D CAD. If not, proceed to the next step, rotating only the fluid region

A.2.2 Geometry Preparation

User 1

1. Rename and do final surface prep on Solid
2. Create composite of parts: fluid, solid, and interface
3. Begin duplication and rotation of parts (Right click on body tab in tree –> duplicate/transform –> rotate) the part 11 or 12 times about the Z-Axis of the "Rotation" Coordinate System (make sure coordinate system is set to the created "Rotation" CCS) by an angle of:

   -15.6521739 degrees (type "-15.6521730 deg" to set the rotation in degrees)

   It will be easiest if you do this on the last rotated part each time so the rotation will be the same angle each time. This will create about half of the annulus for the row of blades. User 1 will be creating the other half. Communicate to know when this is done

User 2

1. Rename and prep fluid part surface
2. Duplicate composite. This includes the solid part and interface part, and the fluid part if it has not yet been duplicated and rotated
3. Begin duplication and rotation of parts from composite (Right click on body tab in tree –> duplicate/transform –> rotate) the part 11 or 12 times about the Z-Axis of the "Rotation" Coordinate System (make sure coordinate system is set to the created "Rotation" CCS) by an angle of:

   15.6521739 degrees (type "15.6521730 deg" to set the rotation in degrees)

   It will be easiest if you do this on the last rotated part each time so the rotation will be the same angle each time. This will create about half of the annulus for the row of blades. User 1 will be creating the other half. Communicate to know when this is done.

A.2.3 Combine Parts

User 1

1. Right click on the scene and select “Hide all parts” and close geometry scene 1

2. Combine composites, move solid regions from composite, then duplicate the composite containing only the interface. Add all solid regions and combine 5 of the solid parts and the related surfaces within that body. After combining the first time move on to the next step.

User 2

1. Right click on the scene and select “Hide all parts” and close geometry scene 2

2. Remove fluid parts from composite, and add them to the new Composite 2. Combine 5 fluid and interface parts into a single fluid part and then combine all related surfaces within the newly combined body. Move to next Step.

A.2.4 Surface Repair

Both Users

1. Re-open geometry scene

2. Drag one of the newly created into the scene, “add to geometry scene -> geometry”

3. Right click on the same part -> Repair surface (coordinate to make sure that only one user is in surface repair mode at a time)

4. As outlined in figure A.10, select Manage Threshold –> Leave “free edges”, “non-manifold edges”, and “non-manifold vertices”. Click on the box with the number of free edges and select “zip edges” (second figure). Make sure that each threshold has a count of 0. If not repeat this step

5. Go back to the combine parts step and repeat both steps with the next 5 parts

6. Combine the 5 newly combined parts for each domain (User 1: Solid and User 2: Fluid) and repeat surface repair steps. This will result in one solid and one fluid part that have been repaired.

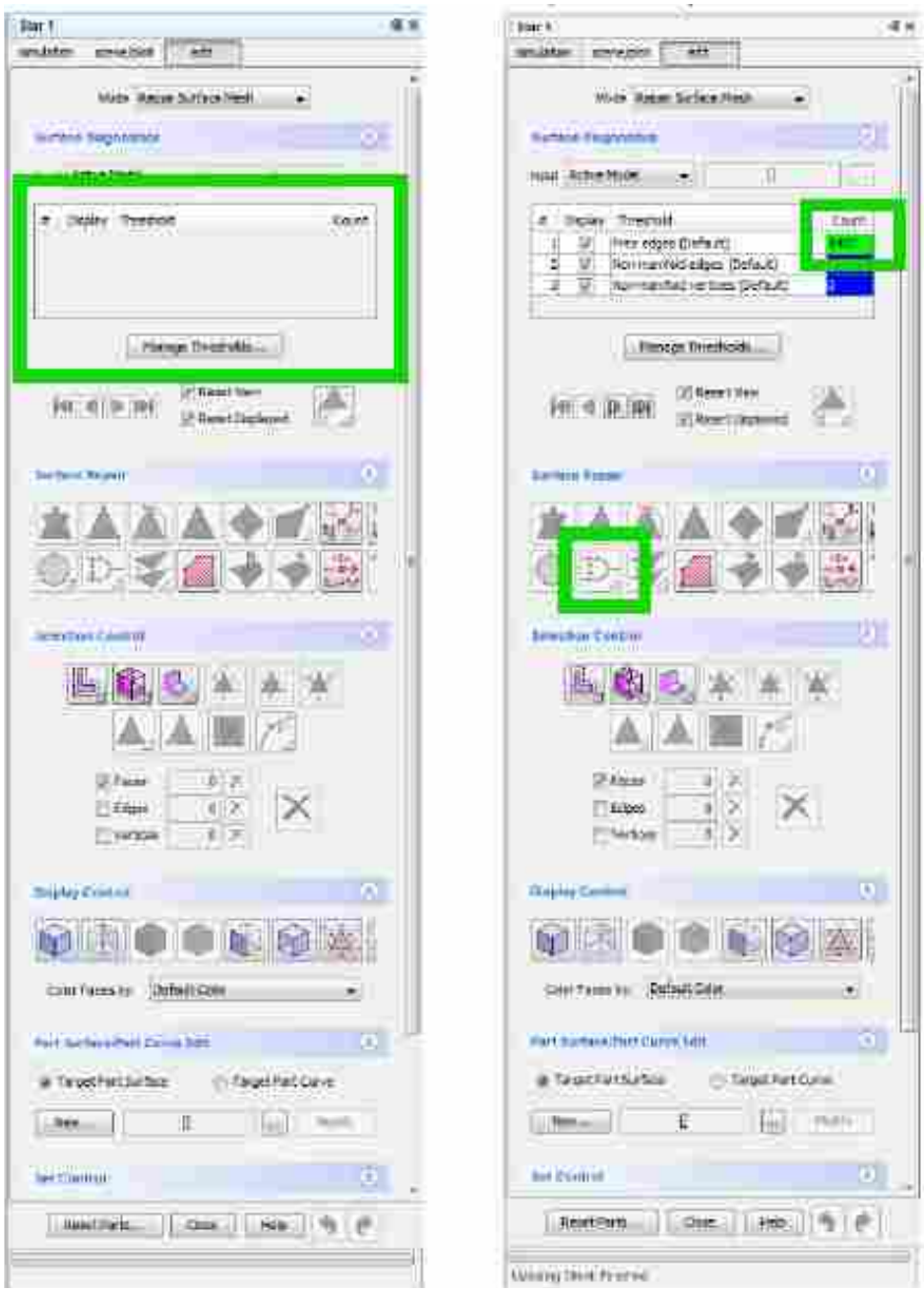


Figure A.10: Method of zipping surfaces in Surface Repair Mode

A.2.5 Region Set-Up

User 1

1. Assign Solid part to region. Select one boundary for each part surface and one region per part.

2. Create interfaces and specify mesh refinement for cooling holes, and other cooling interfaces.

User 2

1. Assign Fluid part to region. Select one boundary for each part surface and one region per part.

2. Set boundary types:

   (a) Cooling ID and Cooling OD: Stagnation Inlet

   (b) Inlet: Velocity Inlet

   (c) Outlet: Pressure Outlet

3. Set boundary conditions as shown in figure A.11: (type psi and r for units in STAR-CCM+)

Cooling Flow		Tt	Pt			
	ID	435.9 R	68.49 psia			
	OD	439.1 R	66.86 psia			
Core Flow		Tt	T	Pt	P	V
	Inlet	790.8 R	788.781 R	65.82 psia	65.2336 psia	110 ft/s
	outlet	790.8			38.67 psia	

Figure A.11: Boundary and initial conditions for AFRL Cooled-Turbine model

A.2.6 Post-Processing

User 1

1. Create surface average report. In the properties window, select Pressure as the scalar field function, and Regions –> Fluid –> Solid Interface as the part. Right click on the report and create monitor and plot.

2. Create a scalar scene. Rename it as “Pressure on Blades”. Set the following Properties:

   (a) Displayers -> Scalar 1 -> Parts: Solid Interface

   (b) Displayers -> Scalar 1 -> Scalar Field -> Function: Pressure

   (c) Displayers -> Scalar 1 -> Scalar Field -> Units: psi

   (d) Displayers -> Scalar 1 -> Color Bar -> Levels: 150

   (e) Displayers -> Scalar 1 : Contour Style: Filled Smooth

   User 2

1. Create scalar scene. Rename it as “Blade Temperature”. Set the following properties;

   (a) Displayers -> Scalar 1 -> Interface: Cooling1, Cooling2, Cooling3, Cooling Holes

   (b) Displayers -> Scalar 1 -> Scalar Field -> Function: Temperature

   (c) Displayers -> Scalar 1 -> Scalar Field -> Units: R

   (d) Displayers -> Scalar 1 -> Color Bar -> Levels: 150

   (e) Displayers -> Scalar 1 -> Contour Style: Filled Smooth

2. Save simulation

A.3 STAR-NX Testing

The testing of the STAR-NX Multi-User CAE Tool involved two users who modeled a simplified car in a wind tunnel, and completed a CFD set-up using STAR-NX.

A.3.1 NXConnect Modeling

User 1

1. Create sketch on ZY plane. Dimension as shown in figure A.12: (Create the sketch so that dimensions are used rather than constraints. This will help when creating expressions)

2. Repeat this step on all new datums. Reopen sketch on the plane of CL*0.035 and change 1.5 to 2.222. Rename the 2.222 expression “CH”

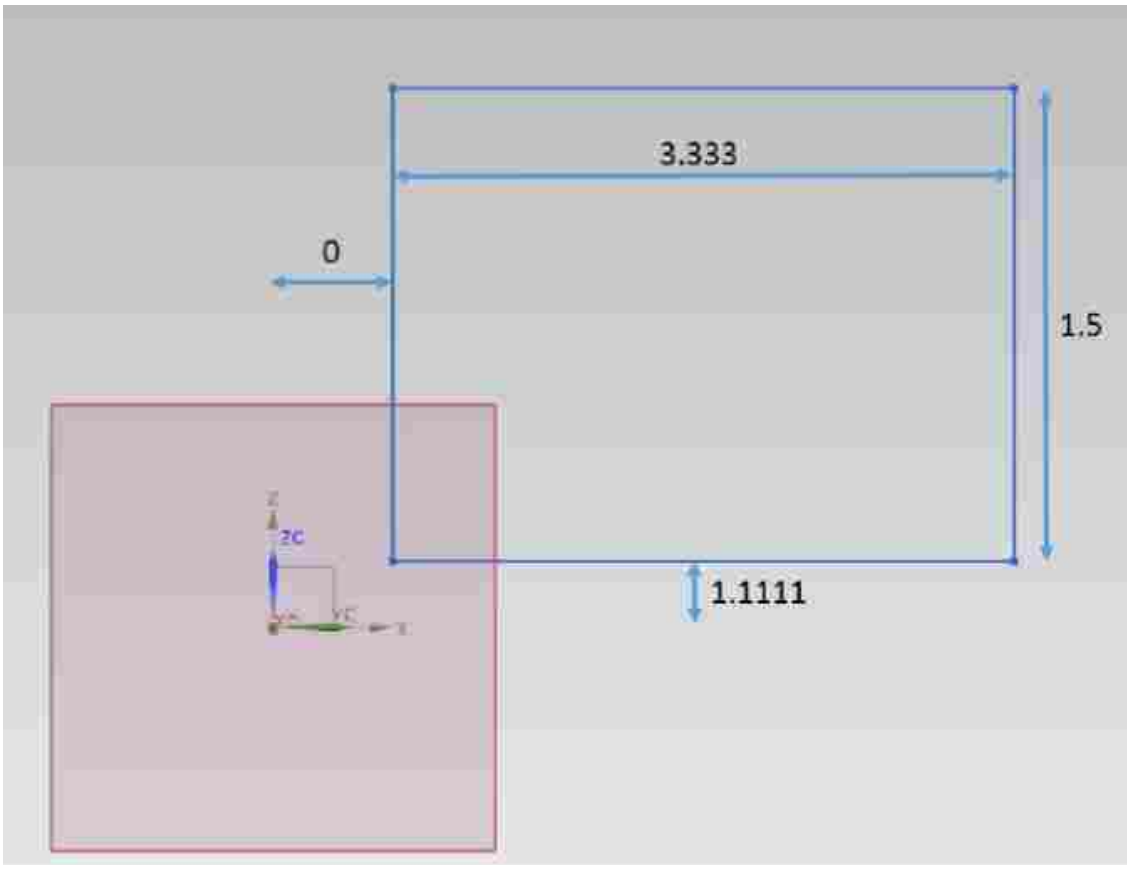


Figure A.12: STAR-NX car cross section sketch

3. Edit all sketches. Change the 1.5 value to the following values: (this is not editing the expressions, only the sketch)

   (a) X = 0 and CL -> CH*0.5

   (b) X = 0.25*CL and 0.75*CL -> CH*0.6

   (c) X = 0.65*CL -> CH

4. Unite all sections of the car

5. Assign boundaries to Inlet, Outlet, Symmetry Plane, and WT Wall

6. Transfer CAE Model using STAR-NX toolbar

User 2

1. Create a datum plane offset 5.555 in. from the ZY plane

2. Go to tools –> expressions and change change the name of that expression to “CL” (Car Length)

3. Create four more datum planes. These will be offset 0.25*CL, 0.35*CL, 0.65*CL, and 0.75*CL

4. Create a sketch on the original ZY plane. Dimension as shown in figure A.13: (Create the sketch so that dimensions are used rather than constraints. This will help when creating expressions)

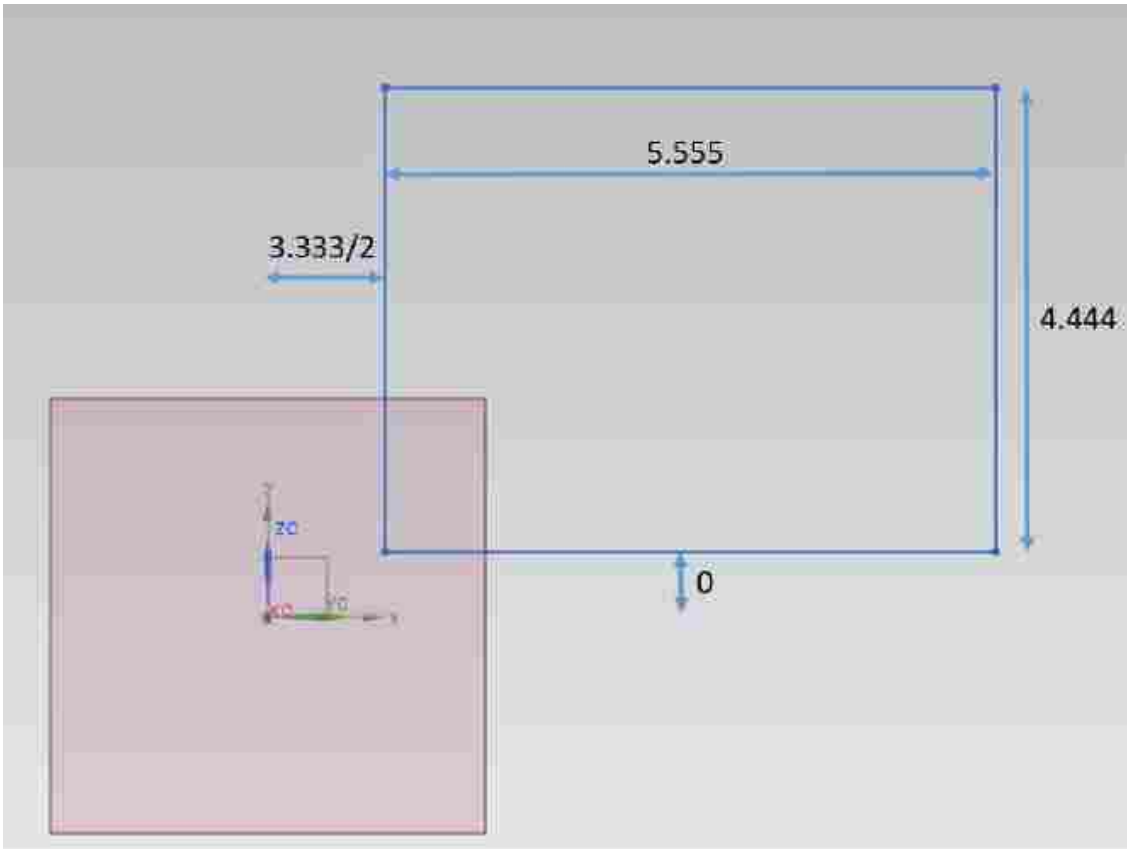


Figure A.13: STAR-NX wind tunnel cross section sketch

5. Extrude the sketch: -1.111 to 6.666

6. Go to tools – expressions and change the names of the expressions with the following values:

   (a) 5.555 -> “wtW” (Wind Tunnel Width)

(b) 4.444 -> “wtH” (Wind tunnel Height)

(c) -1.111 -> “wtB” (Wind Tunnel Back Length)

(d) 6.666 -> “wtF” (Wind tunnel Front Length)

7. Begin creating “Through Curves” between sketches on the newly created datum planes (2 sketches at a time)

8. Subtract Volume of Car from the Wind tunnel

A.3.2 STAR-CCM+ Parameter Changes

User 1

1. Enable collaboration in STAR-CCM+

2. Give host and port number of simulation to user 2

3. Create mesh scene 1 –> Right click on scene and change representation to geometry

4. Close mesh scene 2 when it is created

5. Begin changing geometry parameters in geometry –> CAD Client model, and updating geometry

User 2

1. Open STAR-CCM+, and select File -> Connect to Server. Input the host and port number given by user 1.

2. Close mesh scene 1 and create a new mesh scene.

3. Work with user 1 in updating the parameters of the CAD client model.

4. Ensure that parameter changes are updated to all clients of all programs.

APPENDIX B. STAR-NX CODE

The STAR-NX Code was added to NXConnect. The sections of code shown here show the loop on which STAR-NX is run, the update expression data, the method in which the NXConnect expressions are compared, and how the changes are updated.[1]

Code B.1: Section of the Code that does this

```
private int counter = 1;
private static int runTheStarNXExpressionCheckThisManyTimes = 50;
internal override void CheckForStarNXExpressionChanges()
{
    if (counter >= runTheStarNXExpressionCheckThisManyTimes)
    {
        // Push Expressions Function
        NXConnectController.Dispatch(NXConnectController.Command.PushChangedExpressionsOnTimer,
        DataCaptureModule.NewestMark);
        if (NXConnectApp.Parts.WorkPart.Features.ToArray().Length > 0)
        {
            NXOpen.Features.Feature latestFeature = NXConnectApp.Parts.WorkPart.Features.ToArray().Last();
            latestFeature.MakeCurrentFeature();
        }
        counter = 0;
    }
    counter++;
}
```

Code B.2: STAR-NX Update Expression Undo Marks

```
else if (mark.Name.ToLower() == "update expression data")
{
    // A parameter has been updated!
    // Go find which expression has been updated and send it to the NXConnect server
    NXConnectApp.Parts.WorkPart.Views.WorkView.Fit();
    NXConnectApp.Parts.BaseWorkInfo.Expressions.PushAllExpressions();
}
```

[1] Help with the development of STAR-NX was given by David French, Jonathan Sadler, and Scot Wilcox.

Code B.3: NXConnect Expressions are compared

```
// STAR NX: METHOD OF PUSHING CHANGES
public static IEnumerable<ExpressionData> previousExpressions = null;
public void SyncChangedExpressionsOnTimer(NXOpen.Session.UndoMarkId undoPoint)
{
    // There are currently no old Expressions to compare against
    List<ExpressionCollection.ExpressionData> prevExpressions = null;
    if (prevExpressions == null)
    {
        prevExpressions = new List<ExpressionData>();
    }
    // Create a list of all the current Expressions
    List<ExpressionCollection.ExpressionData> currentExpressions =
                (from exp in NXConnectApp.Parts.WorkPart.Expressions.ToArray()
                where exp.IsAlive()
                select new ExpressionCollection.ExpressionData
                {
                    originalName = exp.Name,
                    originalRHS = exp.RightHandSide,
                    expression = exp
                }).ToList();
    // Expression info for the Current Expressions
    List<ExpressionInfo> currentExpressionInfo = new List<ExpressionInfo>();
    foreach (var expr in currentExpressions)
    {
        //update the ExpressionInfo
        var info = this[expr.originalName];
        if (info != null)
        {
            info.updateInfo(expr.expression);
            currentExpressionInfo.Add(info);
        }
    }
    // If Another list already exists, compare values
    if (prevExpressions.Count > 0)
    {
        //====================================
        // Values of Expressions are compared
        //====================================
        // Updated Features
        List<NXOpen.Features.Feature> changedFeatures =
        (from ex in prevExpressions
        where ex.prevExpressions.IsAlive() &&
        (ex.prevExpressions.RightHandSide != ex.originalRHS ||
        ex.prevExpressions.Name != ex.originalName)
```

```
        from feature in ex.prevExpressions.GetUsingFeatures()
        select feature).Distinct().ToList();
        // Updated Expressions
        var changedExpressions = from exp in prevExpressions
        where exp.expression.IsAlive() &&
        (exp.originalName != exp.expression.Name ||
        exp.originalRHS != exp.expression.RightHandSide)
        select exp;
        // Create Expression info for changed expressions
        List<ExpressionInfo> changedExpressionInfo = new List<ExpressionInfo>();
        foreach (var exp in changedExpressions)
        {
            var info = this[exp.originalName];
            if (info != null)
            {
                info.updateInfo(exp.expression);
                changedExpressionInfo.Add(info);
            }
        }
        List<ExpressionInfo> prevExpressionInfos = currentExpressionInfo;
        ExpressionInfo.Pusher.PusherList changedExpressionsPusher = GetPushers(changedExpressionInfo);
        PushUtils.PushHelper pushHelper = new PushUtils.PushHelper
        (DataCaptureModule.NewestMark, changedExpressionsPusher);
        pushHelper.Push();
        SyncChangedExpressions(previousExpressions, DataCaptureModule.NewestMark);
    }
    // Current expressions are relabled as previousExpressions for the next time through the loop
    prevExpressions = currentExpressions;
}
```

Code B.4: NXConnect controller

```
//  STAR NX is Run When NXConnect is in Modeling Mode
public static NXOpen.MenuBar.MenuBarManager.CallbackStatus PushChangedExpressionsOnTimer...
(NXOpen.Session.UndoMarkId current)
{
    if (topState == States.StateType.Modeling)
    NXConnectApp.Parts.BaseWorkInfo.Expressions.SyncChangedExpressionsOnTimer(current);
    return NXOpen.MenuBar.MenuBarManager.CallbackStatus.Continue;
}
```