

Using the TetrUSS CFD Suite in Undergraduate Research

**CDR Robert Niewoehner, USN
ENS Joshua Filbey, USNR
United States Naval Academy**

Abstract

With the growth in computational power and the availability of maturing software, Computational Fluid Dynamics (CFD) is crossing the threshold from a specialized discipline to a widely accessible tool. Specifically, the difficulty of the enabling mathematics and the challenge of mastering the available codes has heretofore restricted substantial application of CFD to graduate studies, or simplistic problems for undergraduates. Codes now available from commercial, academic and government sources seek to improve the accessibility of CFD and its utility to a wide range of applications.

NASA advertises the TetrUSS CFD system, developed and maintained by NASA Langley Research Center (LaRC), as "CFD for the non-expert user." Modules of the TetrUSS system provide graphical interfaces for the development of unstructured grids about surface definitions imported from common CAD tools, and the solution of a viscous flow within the grided volume. While in use by many expert CFD users, NASA's expressed intent is to equip non-expert users interested in CFD's product rather than its process.

This paper chronicles the "non-expert" experience of an advanced undergraduate researcher and his faculty advisor in applying these tools to a complex, full-configuration aircraft for the purpose of analyzing a flight dynamics problem. Comments are provided on the adequacy of the available training, the ease of use of the ensemble of modules, the requisite academic preparation, and the quality of the results. Furthermore, the paper discusses both the present limitations for use in undergraduate settings, as well as viable applications.

I. Background

Early developmental flight tests of the F/A-18 E/F Super Hornet by the principal author revealed unpredictable lateral handling in the Power Approach (PA) configuration at angles-of-attack between 12 and 15 degrees, a phenomena later named "PA wing drop." As this was a critical flight condition for shipboard catapult takeoff, the problem had to be fixed decisively prior to the initial carrier-based testing. Prior to serious investigation, and hoping for a quick fix, a list of a dozen easy software variations to the flight control program were proposed, coded and slated for flight. Testing revealed that the closure of an aerodynamic vent at the wing root's leading edge solved the problem without any adverse impact. The change was burned into all subsequent flight control software loads; and the development program put the problem behind them, pressing forward to attack the myriad other challenges unearthed daily by flight test. The

mechanics of neither the cause nor solution were understood, but in a climate awash in unsolved problems and programmatic cost and schedule pressures, solved problems draw little interest.

Successful completion of the program three years later provided time for reflection on the course of events. Among other unanswered questions, engineers and pilots alike wondered “What caused the PA wing-drop in the first place, and why was closure of the vent effective?” In parallel with a wind tunnel study performed by a Naval Air Systems Command (NAVAIR) researcher, the Aerospace Engineering Department at the U.S. Naval Academy was tasked to perform a computational fluid dynamics study of the flow fields both before and after the fix.

A parallel research effort was intended to capture the benefits of each of the two test methods. A computational study would provide pressure distribution and off-body flow data that could only otherwise be gathered from an elaborate and expensive wind tunnel entry. The complexity of the geometry, with slotted flaps and slots, would necessitate several million nodes, with a consequent computational cost of dozens of CPU hours, per flight condition, per configuration. A ten-day wind tunnel entry, on the other hand, could achieve thousands of data points over a wide range of configurations and flight conditions. Furthermore, agreement between the two at various check cases would serve as a persuasive validation of both.

Convinced by NAVAIR computational aerodynamicists that the software tools would be accessible to a bright young undergraduate, a junior-year Aerospace Engineer was recruited to dedicate himself half-time senior year to the project, under the aegis of the Academy’s Trident Scholar program. Four weeks of the student’s summer were scheduled to provide instruction in the use of the software, under the direct supervision of computational scientists at the Naval Air Warfare Center- Aircraft Division facility at Patuxent River, Maryland.

NAVAIR project sponsors selected NASA Langley’s TetrUSS software suite to perform the study. The complexity of the aircraft geometry suggested an unstructured-grid CFD approach, while the maturity of the TetrUSS suite and its interface were deemed appropriate for an advanced undergraduate student. The availability of TetrUSS training at NASA LaRC factored into the choice.

II. TetrUSS

The Tetrahedral Unstructured Software System (TetrUSS) is a suite of programs designed and maintained by NASA-LaRC for the solution of internal and external fluid flow problems using unstructured computational grids. TetrUSS documentation explains, “The goal is to provide a validated capability to non-CFD expert users for performing rapid aerodynamic analysis and design of complex configurations. Capabilities include rapid grid generation and inviscid flow analysis, and an emerging Navier-Stokes viscous flow analysis. The system consists of a loosely coupled set of software for geometry setup, grid generation, flow solution, and analysis. In the [below] figure, the primary TetrUSS components are respectively GridTool, VGRIDns, USM3D, and VIGPLOT.”

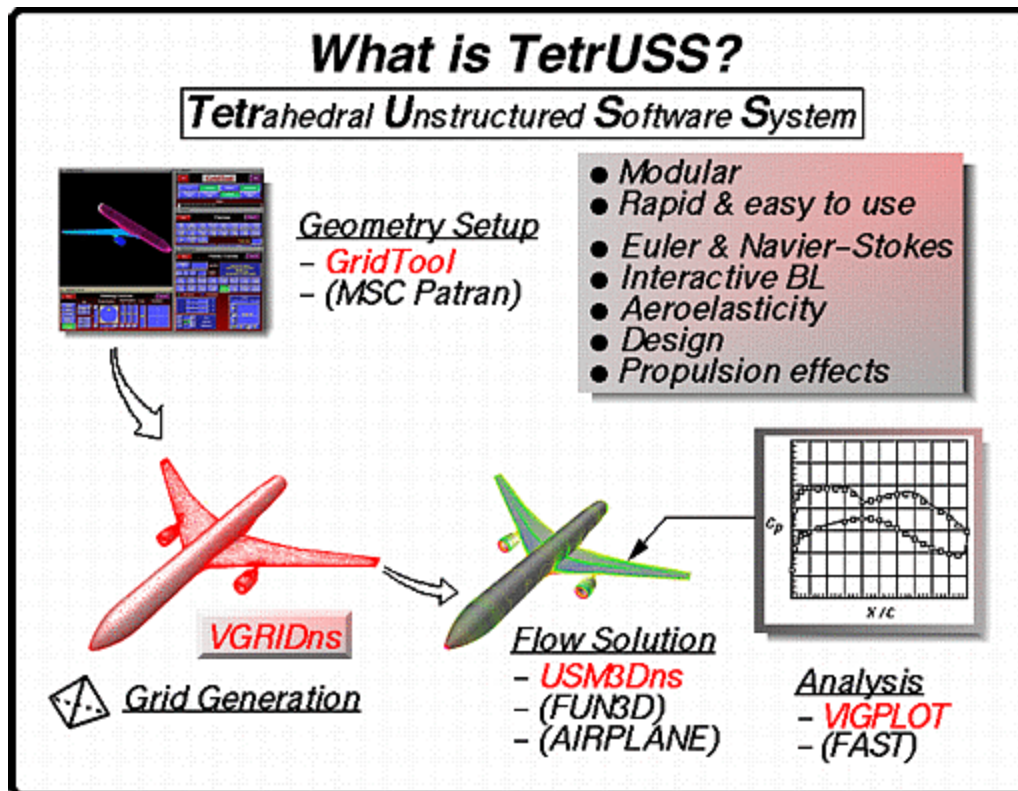


Figure 1. TetrUSS Overview

An overview of the TetrUSS suite may be found at the LaRC website, as well as an extensive TetrUSS bibliography documenting details of the underlying algorithms.¹

As depicted in Figure 1, drawn from NASA-LaRC's documentation, the solution process comprises four major tasks, though the process is not serial as suggested. The TetrUSS suite does accommodate the use of tools outside the suite, but our experience has been restricted to the TetrUSS components alone.

The four major TetrUSS components correspond roughly with the four primary tasks: GridTool defines the problem geometry and boundary conditions, VGRID creates the unstructured grid within the flow volume, USM3D performs the flow solution, and VIGPLOT provides for graphical visualization and analysis of the solution.

The sections below describe both the function of the modules and their supporting utility programs, with Figure 2 providing a map of the interactions between modules. Particulars of our use of TetrUSS will be addressed in subsequent sections. The authors used the most current release of the software.²

GridTool

The GridTool program module of the TetrUSS suite defines virtually all the problems details including geometry, boundary conditions, and the parametrics upon which the grid density will be based.

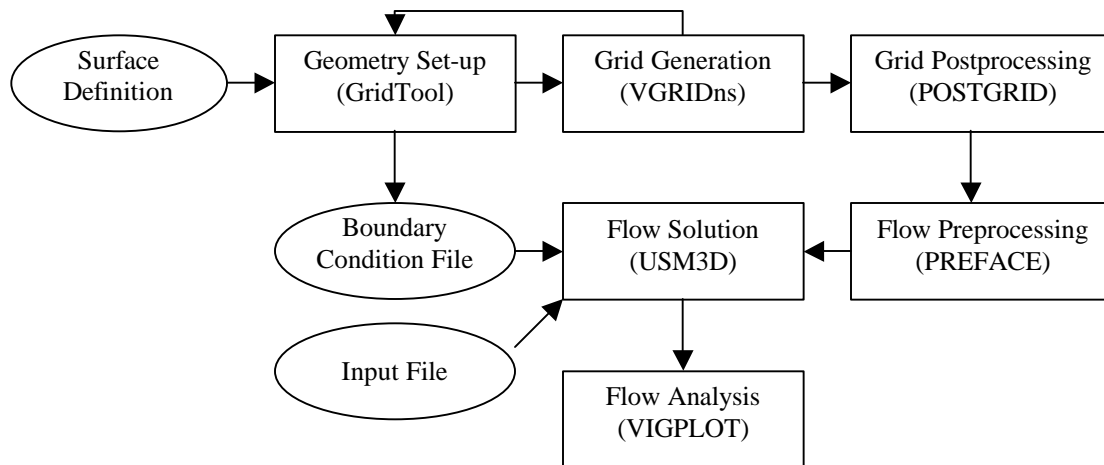


Figure 2. TetrUSS process.

GridTool starts with a high-definition CAD geometry, accepting any of several standard CAD file formats. While a watertight model is preferred (i.e.- no leaks or gaps), GridTool incorporates a modest capability for closing gaps in the model to achieve the watertight character necessary for grid generation. The surface geometry is traced out and formatted for the flow-solver's use. This process involves outlining the surfaces with points and curves to model the shapes and curves of the aircraft as accurately as possible.

Next, the outlined geometry is broken into patches constructed from these curves. These patches construct the starting point for the volume grid, and model the shape of the aircraft and the other boundaries of the computational domain. These patches and the boundary conditions to be imposed on them are the only representation of the model passed to the VGRID.

The patches must satisfy specific criteria in order for the grid to generate successfully. Any curved patch must be three or four-sided, and both as regular in shape and as planar as possible. Planar patches may have any number of sides. High aspect ratio patches or patches with extremely acute angles are also avoided as they render irregular triangles when the grid generation begins. The surface grid generator uses cubic splines to match curved surfaces and so does not model the sharp surface curves perfectly unless the highly curved areas of the surface are broken into appropriate patches. Patching is very laborious but the success of entire grid generation process depends upon the care with which it is done.

Flow properties at each patch are defined as part of the patching process. A number of options exist for boundary definition. For our external flow problem we used five types of boundaries: viscous no-slip patches for the aircraft surfaces, flow through boundaries for the free-stream boundaries, reflection plane for the aircraft plane-of-symmetry, engine inlet and engine exhaust. In the case of the engine inlet and exhaust, the engine flow parameters are specified within the patch definition (the flow solver will balance mass flow rates). Additional patch boundary condition options include rotor and propeller disks, though unnecessary for this application.

Once the aircraft and boundaries are patched, background sources are added to the model within GridTool. The grid generation software, VGRID, uses an advancing front algorithm, solving Laplace's equation, working outward from the aircraft surface. The sources determine the density of the grid's tetrahedra. Complex geometric areas and those areas with large flow gradients, such as the area surrounding our slotted flap, require a smaller, denser grid than those areas such as the middle of the wing where shallow flow gradients might be expected. As grid density determines the numerical convergence properties as well as accuracy, the creation of the sources is a vital phase in the process. While the placement, size, and orientation of sources is heuristic, VGRID automatically determines source strength.

GridTool is consequently more than the translation of a CAD surface into a CFD-ready model. With the exception of those flow properties specified in the input file for the flow solver, GridTool provides the environment in which all of the geometry and grid properties are defined.

VGRIDns

Generation of a surface grid is the next step towards problem solution, performed using the VGRID software module. The surface grid is a mesh of triangles on all the surfaces forming the boundaries of the computational domain. When the volume grid is generated, the triangles of the surface grid become the base of the first layer of tetrahedra. The principal task involves loading the geometry and sources from GridTool into the VGRID database and generating a surface mesh on each patch based on the defined sources. Each patch is then examined to evaluate the mesh generated, ensuring there are no irregular triangles. The program automatically assesses the surface mesh quality, with poorly meshed patches identified to the operator.

If bad meshes are found during this process, three things that can be done. The first choice requires rotating the patch within VGRID so that the grid generation begins on a different side. The surface grid generation algorithm starts from one side of the patch and works outward so the side from which the front advances directly determines the final result. If this does not correct the problem, the second method involves altering the actual shape of the patch. This method entails going back into GridTool, and either splitting the patch or rearranging the patch geometry to yield a more regularly shaped patch. The final method of correction entails changing the nearby source sizes and intensities in order to change the resulting surface grid density. In a problem with 300+ patches, several dozen required reorientation or splitting, while only a few required source adjustments. As can be seen, however, the flow from GridTool to VGRID is not serial, but entails frequent iteration to/from GridTool, adjusting the input file to ensure a satisfactory resultant surface grid.

After each patch is meshed satisfactorily, the model goes back into GridTool, and stretching is applied. The stretching process involves stretching the grid in various places where no large gradients are expected in order to reduce cell count. In particular, gentle gradients in the span-wise direction along wings and flaps permit significant stretch in the span-wise direction. The stretching process, in this case, leaves the chord-wise grid density unchanged.

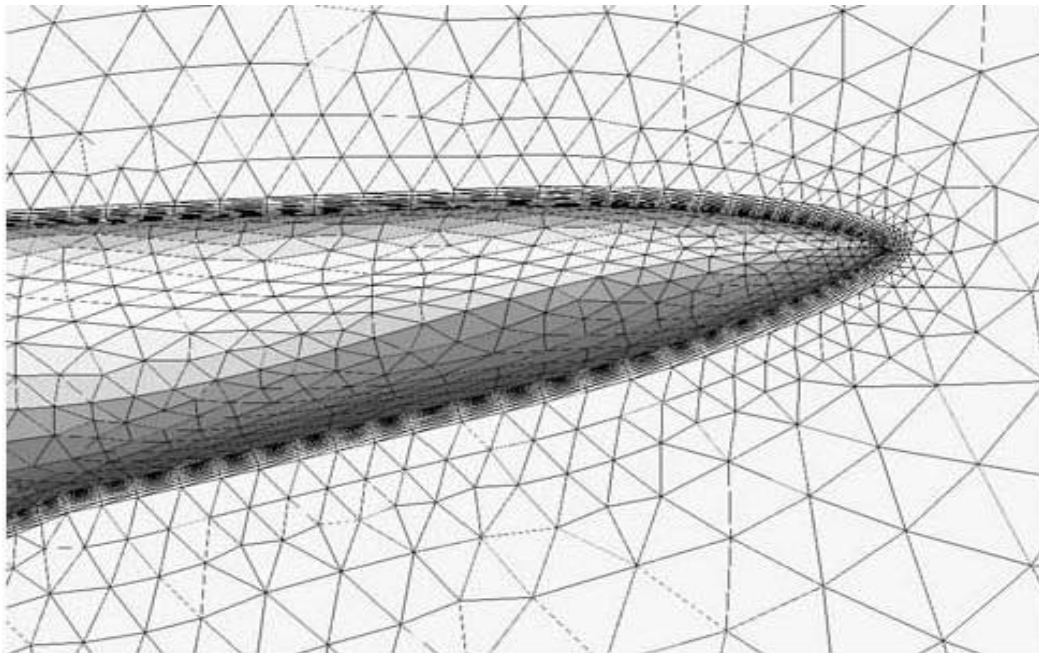


Figure 3. Viscous Grid on Reflection Plane.

During the stretching process, the viscous boundary layer is also defined, which tells the grid generation software where viscous flows are present, and how large to make the viscous portion of the grid. The viscous grid generation algorithm uses an advancing layer method to generate the viscous layer, based on the following properties: the initial thickness of the layers, the number of layers, and the thickness of each following layer. A Blasius calculation for a flat plate determines the thickness of the boundary layer; and the number obtained is then used to determine the thickness of the initial layer, and the number of requisite layers. Once the viscous grid is generated, the geometry is taken back into VGRIDns, and the surface mesh is regenerated; this time accounting for the stretching and viscous effects. In Figure 3, a close-up of the nose of the F/A-18E, depicts this viscous layer, and how it differs from the inviscid portion of the grid. Figure 4 depicts the grid generation process to this point, outlining the major steps necessary to create a surface mesh.

The program calculates surface normal vectors as the final step prior to generation of the volume grid. These vectors determine the grid's growth properties and the direction of layer growth for each patch. If these vectors conflict, e.g., pointing into another surface, the grid generation process aborts and the problem must be resolved. Problems here typically occur from poor source selection or highly concave geometries. Once resolved in GridTool, the entire surface and volume generation process is repeated to reflect the changes made to the model.

When a final volume grid is generated, a small utility program, POSTGRID, analyzes the entire grid and identifies trouble spots where sharp grid variations may cause problems for the flow solver. Portions of the grid may then be scooped out and rebuilt in the problem areas to try to obtain a smoother mesh. Once this process is completed, the volume grid is taken back into GridTool again so that the surface front can be projected onto the original CAD surface

geometry. This step corrects for node locations off of the true surface boundary caused by the patches' spline approximation of the airplane's curvature.

USM3D

With the grid generation complete, the problem can at last be handed over to USM3D for solution. In order to run properly, the flow solver requires the grid file as well as an input file listing all the initial conditions governing the problem, such as free-stream pressure, temperature, and Mach number. The input file also includes engine specific data used by the flow solver to model the engine exhaust and the mass flow rate into the inlet.

USM3D's output includes both the state vectors for each cell locations, and the integrated 3-axis force and moment coefficients on the model, or any selected subset of model components.

VIGPLOT

VIGPLOT is the TetrUSS application designed to specifically manipulate the USM3D output files for post-run visualization and analysis. Multiple common options are available via a graphical interface, and exportable in several formats.

III. The "Non-Experts"

As this review intends to assess NASA's "non-expert" claims, establishing the team's "non-expert" credentials is appropriate.

The principle researcher was a senior Aerospace Engineering student in the top one percent of his graduating class at the United States Naval Academy. His formal coverage of fluids had included an introductory course, two semesters of Aerodynamics, one semester of Gas Dynamics, and a lab course. His mathematical preparation included three semesters of Calculus, Ordinary Differential Equations, and an introduction to Numerical Methods. His only preparatory exposure to Partial Differential Equations had occurred in the Aerodynamics sequence, and he did not have the prior benefit of our elective offering in computational fluids. His selection to the Trident research program had included a formal proposal and defense.

The student's training in the use of the TetrUSS suite consumed a dedicated four-week summer internship. The first week was spent at Patuxent River, under the guidance of computational aerodynamicists with the Naval Air Systems Command (NAVAIR), and devoted to basic familiarity with the TetrUSS components and file formats. The second week was spent at NASA-LaRC taking their TetrUSS course. Weeks three and four were again at Patuxent River, assembling the pieces of the CAD model.

The faculty advisor's research and experience was in flight test, and flight dynamics and control, with no prior experience in either advising student research or computational fluids. In order to be conversant with the problem process, the advisor likewise attended the NASA-LaRC training. The student continued to have ready access to NAVAIR scientists, as well as TetrUSS designers and technical support personnel at NASA-LaRC.

IV. Assessment

The following assessment is formatted as questions most likely to be posed by faculty members considering use of such a tool.

Was the TetrUSS suite an appropriate match for this specific technical problem?

The problem's originally intended to solve the viscous flow about a half-body model which included leading edge flaps and a slotted trailing edge flaps at approach airspeeds. The suite has been previously used by NAVAIR to solve the flow about the same airplane at transonic speeds with no high-lift devices (a two-million-cell problem). The additional geometrical complexities of flaps and vents motivated the selection of an unstructured grid here as well, and were expected to swell the problem size to three million cells.

As of publication time, seven months into the program, a viscous grid still eluded us. The slotted flap, and multiple sharp interior angles had proven difficult to grid. Consequentially, first the slot and then viscosity were abandoned in order to meet graduation milestones. While our NAVAIR mentor has successfully generated the viscous grid for a flaps-up model of the same airplane, the flight condition had been transonic rather than low speed. It appears that the additional thickness of the slow-speed boundary layer was responsible for conflicts in highly concave areas of the model as the advancing front built out from the surface. The slot and the inlet lip provided the most challenging regions.

The inviscid calculations have been very fruitful and have identically matched the wind tunnel trends we sought to analyze. Furthermore, the output feature that enabled panel-by-panel analysis of the force and moment contributions provided the perfect tool for our sensitivity analysis, permitting localization of the airplane feature most sensitive to variation in flight condition. On the basis of these wind tunnel affirmed results, the suite must be considered a successful match with the problem.

The display and quantitative output capabilities of the TetrUSS suite were very powerful, and provided the full range of analytical options necessary and common to most commonly used CFD products. Figure 4 below depicts our two test configurations as mirrors of one another, with the static pressures in a plane perpendicular to the longitudinal axis.

What computational resources were required?

All of the model preparation was performed on SGI O2 workstations, with the grid built on a SGI Origin 2000, and requiring 15 minutes. The later was chosen purely for the greater installed memory. The inviscid grid was 1.2 million cells, and the flow solution required less than 12 hours (clock-time) running parallel on eight Origin 2000 processors, requiring 1.6Gbytes of memory. The identical grid was run on a T90 Cray in less than an hour. Unfortunately the T90 module supporting engine data was unstable, and the engine-off results (free flow through) did not correlate well with the wind tunnel results. We then chose to rely on the local parallel machine for the balance of the student's work. The T90 will again be attempted once a viscous grid is achieved due to the growth in cell count and requisite memory.

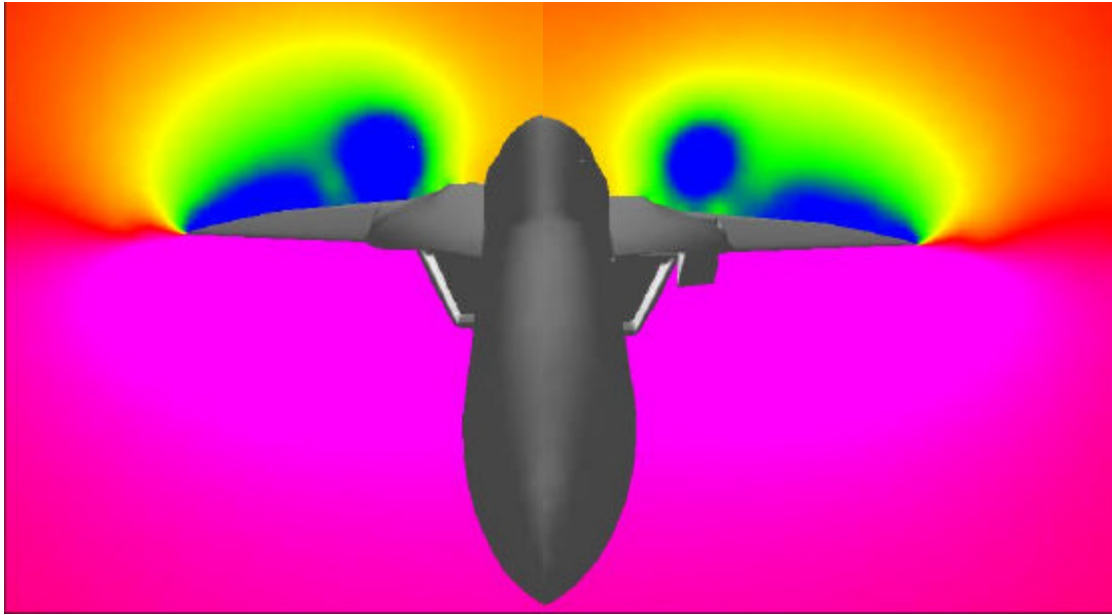


Figure 4. Representative VIGPLOT Output.

Was the student adequately prepared academically for a major research effort using these tools?

Yes, given the quality of the final product, the student's personal capabilities and his academic preparation were sufficient. Clearly a broader mathematical base both in analytical and computational partial differential equations would have been highly desirable, though they proved unnecessary. Once formally exposed to computational methods in graduate school, mastery will surely come easily due to this experience.

Was the internship worthwhile or required?

The dedicated four weeks of internship time must be considered mandatory, including the training class for both student and advisor. The dedicated internship provided a rapid spool-up that would have been very difficult to achieve with a part-time researcher during the academic year. Watching over the shoulders of researchers who use the TetrUSS suite daily in their professional lives gave the student his first insight into both the process and its various challenges. I would not attempt a similar effort without repeating this step.

Is the software adequately supported by NASA-LaRC?

NASA-LaRC's technical support for the TetrUSS suite is a phenomenal benefit. At the expense of travel alone, a weeklong class in the use of the software is conducted monthly at LaRC. This class is very well done, with each class member working through the various modules on either a personally provided model, or a tutorial model. It does take fully 4-5 days to get to the point of solving for the flow, but by the end of the week, each attendee will have manipulated all the critical features of each module. Furthermore, the week on site provides personal contact with TetrUSS's designers and support staff, so that as questions arise downstream, you're posing your questions to people you've met. The technical support quality on-site and subsequently by phone has consistently been superb. Technical support has not been restricted to mechanics of the software suite, but help was always readily available for problem specific questions both from the technical support staff, and the TetrUSS designers.

Was the training class required?

The training is mandatory in order to make any sense or progress with TetrUSS. The software does have a very mature graphical interface; and the training manuals are very good, but the complexity of the system could be only otherwise be mastered by an inordinate amount of time spent under the tutelage of someone who had themselves already mastered the suite. Use of TetrUSS should not be attempted without dedicated instruction from either LaRC or a seasoned TetrUSS veteran.

Furthermore, in our context, the faculty advisor had no prior familiarity with the software. It would have been very difficult to supervise the project without having also attended the training. Proficiency in the software requires almost daily interaction, and so a faculty member who attends the training and then takes a supervisory role will not long remember how to perform most of the functions. Supervision does require, however, an understanding of the process upon which the system is based, and the difficulties a student is likely to encounter. Therefore, the training should likewise be considered necessary for a faculty member supervising students depending upon TetrUSS for their results.

What specific difficulties were encountered?

The patching process whereby a CAD model is transformed into spline-defined patches is very time-consuming. The process is not mechanical and requires some judgment. Some guidance is provided both in the manuals and course-work, but mostly the student learned by attempting to create surface grids, identifying trouble spots, and then re-patching the troublesome locations. The process is hardly serial as suggested in the overview, and requires daily iteration between the GridTool and VGRID modules until the final grid is resolved. Several man-months were devoted to this phase alone, and currently continue in pursuit of an inviscid grid.

The patches provide primarily the boundary definition; the sourcing process establishes the grid attributes. Selection of source positions, sizes and orientations is based largely on very heuristic rules-of-thumb. For a complex geometry such as ours, coaching from a TetrUSS veteran was necessary, and we've had our grid reviewed frequently by both NAVAIR and NASA personnel intermittently through its construction. Though grid quality is very subjective, it improved dramatically with the release of GridTool version 3.2b, which featured automatic selection of source intensities. This change mid-year resulted in visually obvious improvements in the smoothness of the grid, and a welcome reduction in cell-count.

As of the time of submission, a viscous grid still eludes us. Specifically, several highly concave regions in the vicinity of the inlet are requiring considerable tailoring of the boundary layer grids. This is a region for which a transonic grid was easily calculated, but for which the additional thickness at slow speed is problematic. The solution will assuredly come at the expense of increased cell count. The set-backs here we have regarded as symptomatic of the complexity of the geometry rather than a weakness of the TetrUSS suite. Fortunately, our parallel wind tunnel effort has affirmed the validity of the inviscid results for the particular question we seek to answer with this application.

Would this software be a suitable fit for other advanced undergraduate or graduate students?

Both the student and faculty advisor agree that the software has indeed achieved its stated goal of placing full viscous CFD in the hands of the “non-expert” user as conceived by NASA-LaRC. Mastery of the software and the complexity of its use were well within the expectations of the scope of effort required to pursue so substantial a project. Indeed our most important observation regarding TetrUSS is that CFD study of complex aerodynamic configurations is now truly viable at the advanced undergraduate level, due to both the maturity of the software and the power of the hardware. Five years ago, a configuration such as ours could only have been tackled by the most expert of researchers, and then only with the patience of Job. No one would have dreamed at that point of handing a half time undergraduate such a problem.

Achieving the goal of “non-expert” use should not however be confused with pervasive academic use, something clearly beyond the designers’ intent. The complexity of the software and the effort required for mastery prohibit use for lower scale projects, such as student design projects or course-work assignments, unless some team member has come by mastery in another project context. Casual use by undergraduates or master’s students remains out of reach and the TetrUSS suite should not be construed as opening the door for viscous CFD for every student. Projects of a scope similar to our own probably constitute the minimum necessary to warrant the significant investment in student and faculty training time.

Should educators be concerned about “turn the crank” software?

A philosophical issue also came to light during the project. The Naval Academy’s Trident program has recently seen increased reliance upon sophisticated software packages among the program’s students. The program’s steering committee has raised concerns about undergraduates using such packages where the process is a black box into which data is poured and results extracted. Specifically, are we contributing to their education if they’re turning the crank on software into which they have no insight regarding its inner mechanics? This is a legitimate general concern, though probably not applicable to the TetrUSS suite. Successful project execution, in this case, required a level of user interaction that compelled the student to independently pursue an understanding of many internal mechanical issues. Examples included understanding the selection of turbulence models, selection of CFL number, selection of implicit vs. explicit routines, source influence on solutions to the Laplace equation, etc. Grappling with such details ensured that the faculty’s broader pedagogical interests were satisfied.

V. Conclusions

NASA has achieved the goal of making CFD accessible to the non-expert user. Specifically, TetrUSS makes full Navier-Stokes CFD solutions of complex geometries feasible for the undergraduate researcher approaching a large-scale project. A crucial piece of the suite’s suitability included NASA-LaRC’s superb technical support.

TetrUSS does not make CFD accessible to the casual academic user due to the significant necessary training time. Proficiency in the software requires consistent use throughout a working week. Some parts of the process are ‘art’ rather than ‘science’ and the non-expert user requires access to an expert in the art.

VI. Recommendations

This team recommends that TetrUSS suite be considered for undergraduate research students or masters' thesis students working projects of sufficient scope such that the return justifies the considerable investment necessary for mastery of the software. Training in the software for both student and advisor should be considered mandatory, and ready access to a TetrUSS veteran is highly desirable.

BIBLIOGRAPHY/ENDNOTES

1. <http://ad-www.larc.nasa.gov/tsab/tetruss> (overview),
2. <http://ad-www.larc.nasa.gov/tsab/tetruss/pubs.html> (TetrUSS bibliography).
3. The following versions of the TetrUSS suite were used Gridtool 3.2b, VGRIDns 3.3, PostGrid 3.3, and usm3d 5.1.3.

CDR ROBERT NIEWOEHNER, USN

Commander Niewoehner is Chairman of Aerospace Engineering at the United States Naval Academy (USNA). He received a B.S. in Physics from USNA, a M.S.E.E. from the Johns Hopkins University, and a Ph.D. in Aerospace Engineering from the Naval Postgraduate School. From 1995 to 1998 he was the Navy's Chief Test Pilot for the development of the F/A-18 E/F Super Hornet. CDR Niewoehner has over 2600 flight hours in 28 different airplanes.

ENS JOSHUA FILBEY

Ensign Filbey is a 2000 Aerospace Engineering graduate of USNA. During his senior year, ENS Filbey was a Naval Academy Trident Scholar and devoted approximately half of his academic time to the research discussed by this paper. ENS Filbey is currently awaiting the start of flight school, on the path to becoming a Naval Aviator.