



Visualization Techniques of a CFD++ Data Set of a Spinning Smart Munition

by Richard C. Angelini and Jubaraj Sahu

ARL-TR-3380

December 2004

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

DESTRUCTION NOTICE—Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5066

ARL-TR-3380**December 2004**

Visualization Techniques of a CFD++ Data Set of a Spinning Smart Munition

Richard C. Angelini

Computational and Information Sciences Directorate, ARL

Jubaraj Sahu

Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) December 2004		2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Visualization Techniques of a CFD++ Data Set of a Spinning Smart Munition				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Richard C. Angelini and Jubaraj Sahu (both of ARL)				5d. PROJECT NUMBER 4U03CC	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory Weapons and Materials Research Directorate Aberdeen Proving Ground, MD 21005-5066				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-3380	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Computational fluid dynamics (CFD)++ is a modern state-of-the-art CFD code that is well validated and routinely used at the U.S. Army Research Laboratory for accurate predictions of projectile aerodynamics. As part of a Department of Defense high performance computing Challenge Project, CFD++ code was used to perform unsteady (time-accurate) high performance computing numerical simulations in a new area of aerodynamic research on synthetic jets to control subsonic projectiles. Efficient visualization of the enormous amounts of data that are regularly produced by these physics-based unsteady numerical simulations is challenging because of the complexity of the grids, the size of the data sets involved, and the physical properties of the phenomena to be visualized. In this report, we describe techniques for visualizing the complex fluid flow in and around a pulsating jet diaphragm.					
15. SUBJECT TERMS CFD++; computational fluid dynamics; Ensign; scientific visualization					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UL	18. NUMBER OF PAGES 30	19a. NAME OF RESPONSIBLE PERSON Richard C. Angelini
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-278-6266

Standard Form 298 (Rev. 8/98)

Prescribed by ANSI Std. Z39.18

Contents

List of Figures	iv
Acknowledgments	v
1. Introduction	1
2. Background	3
2.1 CFD++	3
2.2 Scientific Visualization	3
3. Analysis and Visualization	4
4. Methodology	7
4.1 Preparing Data for Visualization	8
4.2 Clip Plane	9
4.3 Vector Arrows	10
4.4 Streamlines	12
4.5 Path Lines	14
4.6 Passive Scalar	15
5. Conclusion	16
6. References	17
Appendix A. Metacomp Environment Variables	19
Appendix B. Sample Ensight Command File for Creating an Object-Centered Clip Plan	21
Distribution List	24

List of Figures

Figure 1. Schematic of synthetic jet and its location on the projectile.	1
Figure 2. Computed velocity magnitude contours.	5
Figure 3. Instantaneous particle traces in the wake.	5
Figure 4. Lift force attributable to the jet versus angle of attack.	6
Figure 5. Sample shell script used to convert CFD++ output to Ensight binary format	6
Figure 6. Ensight case file.	9
Figure 7. Clip plane colored with scalar value pressure.	11
Figure 8. Clip plane colored with scalar value turbulence.	11
Figure 9. Vector arrows in base flow region of projectile (arrows colored by velocity).	12
Figure 10. Streamline animation during “pulse” emission.	13
Figure 11. Streamline animation when the jet is not pulsing.	13
Figure 12. Path lines indicating those particles emitted during the first (red) cycle of the projectile and during the second (blue) cycle.	14
Figure 13. Blue isosurface representing the passive scalar calculated for the pulsating synthetic jet.	16

Acknowledgments

The authors thank Dr. Ralph Noack, ARL MSRC PET, for consultation on the use of a passive scalar to assist in the representation of the jet thruster dynamics. This work was supported in part by a grant of high-performance computing time from the U.S. Department of Defense High Performance Computing Modernization program at the ARL Major Shared Resource Center, Aberdeen Proving Ground, MD.

INTENTIONALLY LEFT BLANK

1. Introduction

The U.S. Army is seeking a new generation of autonomous, course-correcting, gun-launched projectiles for infantry Soldiers. Because of small projectile diameter (20 to 40 mm), maneuvers by canards and fins seem very unlikely. An alternate and newly evolving technology is the micro-adaptive flow control through synthetic jets. These very tiny (on the order of 0.3 mm) synthetic micro-jet actuators have been shown to successfully modify subsonic flow characteristics. The synthetic jets (fluid being pumped in and out of the jet cavity at a high frequency on the order of 1000 Hz) are control devices (see figure 1) with zero net mass flux and are intended to produce the desired control of the flow field through momentum effects. Many parameters such as jet location, jet velocity, and jet actuator frequency can affect the flow control phenomenon. Until now, the physics of this phenomenon have not been well understood and advanced numerical predictive capabilities or high fidelity computational fluid dynamics (CFD) design tools have not been developed or applied to three-dimensional (3-D) simulation of these unsteady jets. The current research effort is aimed at advancing the state-of-the-art in CFD and advanced flow visualization to accurately predict and provide a crucial understanding of the complex flow physics associated the unsteady aerodynamics of this new class of tiny synthetic micro-jets to control modern projectile configurations.

We control the trajectory of a 40-mm spinning projectile by altering the pressure distribution on the projectile (figure 1) through forced asymmetric flow separation. Unsteady or time-accurate CFD modeling capabilities were used to assist in the design of the projectile shape, the placement of the synthetic actuators and the prediction of the aerodynamic force and moments for these actuator configurations.

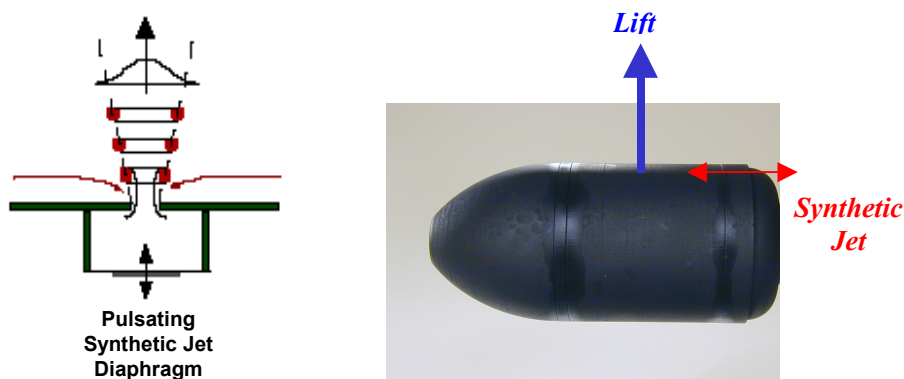


Figure 1. Schematic of synthetic jet and its location on the projectile.

Visualization of this complex problem is essential for allowing scientists to inspect and analyze the voluminous amounts of results from the computation, make conclusions about the phenomena being studied, and share those results with their peers. On a smaller, less complex data set,

routine images and animations would be obtained with any number of techniques and software packages. Clip planes created in and around an object that does not have changing coordinates or an object that does not have characteristics of pitch and yaw motion is a common practice and easily achieved by many applications. Streamlines calculated on a steady state problem are also easily obtained since it is simply a matter of placing the particle emitters and calculating the traces.

However, from a visualization standpoint, there is very little about the time-dependent numerical simulation of this 40-mm autonomous, gun-launched projectile that is routine. There are many challenges for visualization of this computational model and these challenges are highlighted through this report. The projectile is spinning and the center of rotation displays significant pitch and yaw, making a routine visualization such as an axi-symmetric clip plane very difficult to achieve. To create the desired result, the clip plane must be oriented by calculation along the centerline of the projectile rather than along a common XYZ or IJK plane. In order for us to understand the interaction of the synthetic jet with the free stream in the base region of the projectile, particle emitters need to be “glued” to the slot jet diaphragm as the projectile spins. The coordinates for these emitter points are changing over time, but it is essential to the integrity of the path line computation that these points represent the same release node at each time step. Additionally, it has proved to be very difficult to visually represent the effects of the pulsating synthetic jet in and around the diaphragm since the fluid flow is easily lost in the free stream flow around the projectile. Finally, while it is possible to investigate an individual time step from this unsteady calculation, it is currently impractical to attempt to interactively examine the entire complex phenomenon as it marches through time. The nature of the input/output (I/O) requirements to move between time steps and calculations required to visually represent the results requires the extensive use of batch processing to create animations. These animations allow for the evaluation of trends in the computation over time, rather than our having to look at a static snapshot of the simulation. Therefore, an overall evaluation of the entire simulation is provided.

To address the aforementioned challenges associated with visualization of the 40-mm synthetic jet munition, this report focuses on the specific methods used to visualize the complex fluid flow in and around a pulsating jet diaphragm. We start with an efficient scripting methodology for converting a large data set to a format compatible for visualization. Then, techniques for developing an axi-symmetric clip plane and vector arrows for the complex time-dependent motion of a small diameter munition are presented. Finally, we focus on visually representing the fluid flow associated with the jet pulse in the synthetic jet diaphragm through the use of calculated path lines and isosurface generation of passive scalar values on this very large, time-dependent computation.

2. Background

2.1 CFD++

A commercially available code, CFD++ (1,2), is used for the time-accurate unsteady CFD simulations. The basic numerical framework in the code contains unified grid (structured and unstructured grids), unified physics (incompressible and compressible Navier-Stokes equations with turbulence modeling), and unified computing (implementation on massively parallel computers based on the distributed memory message-passing model via native message-passing libraries or message-passing interface) features. The complete set of 3-D time-dependent Navier-Stokes equations is solved in a time-accurate manner for simulations of unsteady synthetic jet interaction flow field on the M203 grenade-launched projectile with spin. The 3-D time-dependent Reynolds-averaged Navier-Stokes (RANS) equations are solved via the finite volume method (1):

$$\frac{\partial}{\partial t} \int_V \mathbf{W} dV + \oint [\mathbf{F} - \mathbf{G}] \cdot d\mathbf{A} = \int_V \mathbf{H} dV \quad (1)$$

in which $\mathbf{W}$ is the vector of conservative variables, $\mathbf{F}$ and $\mathbf{G}$ are the inviscid and viscous flux vectors, respectively, $\mathbf{H}$ is the vector of source terms, V is the cell volume, and A is the surface area of the cell face.

Production-level calculations from CFD++ regularly produce hundreds of gigabytes (or even terabytes) of results. An efficient method of visually analyzing these results is needed for an in-depth understanding of the complex flow interactions. Better understanding of these interactions will result in more timely design of systems important to the Army.

2.2 Scientific Visualization

Scientific visualization is a technique used to demonstrate the results of these complex computations. Numerous visualization packages are available, both commercial and open source, which provide general purpose visualization of computational fluid dynamics data. The Flow Analysis Software Toolkit (FAST) from the National Aeronautics and Space Administration/Ames was predominant in the early years of interactive desktop visualization because it was freely available and it provided one of the first interactive desktop interfaces for visualizing the results of CFD computations. Commercial packages such as Enight¹, and Fieldview² evolved as early, commercially viable, general purpose visualization tools. Tecplot³ progressed from a two-

¹Enight® is a registered trademark of Computational Engineering, Inc.

²Fieldview™ is a trademark of Intelligent Light.

³Tecplot™ is a trademark of Amtec Engineering.

dimensional plotting package to a substantial 3-D graphics package. A new paradigm of visualization tools based on open source packages such as VTK (Visualization Toolkit) is given in ParaView⁴, is a free, fully functional, extendible visualization tool that is being developed by numerous commercial, university, and government partners. Each of these packages provides unique functionality such as client/server processing or parallel processor support and can be appropriate to use, depending on the source of the input data or the results that are expected to be generated.

3. Analysis and Visualization

Time-accurate unsteady CFD computations were performed to predict and characterize the unsteady nature of the synthetic jet interaction flow field produced on the M203 grenade-launched projectile for various yaw and spin rates for fully viscous turbulent flow conditions. A higher order hybrid RANS-LES (large eddy simulation) (2) approach was used for accurate numerical simulations of unsteady flows associated with synthetic jets. This approach computes the large eddies present in the turbulent flow structure and allows the simulation to capture with high fidelity additional flow structures associated the synthetic jet interactions in a time-dependent fashion. Modeling of azimuthally placed synthetic micro-jets required a very detailed grid with highly specialized boundary conditions for the jet activation and the use of advanced hybrid LES approach permitting local resolution of the unsteady turbulent flow with high fidelity. The addition of yaw and spin while the projectile is subjected to the pulsating micro-jets rendered predicting forces and moments a major challenge. The DoD High Performance Computing Modernization Office (HPCMO) selected this research as a grand challenge project and provided the massive computational resources required by these unsteady time-accurate simulations (3,4). This project has required extensive use of HPC assets, and advanced visualization played a critical role in providing fundamental understanding of the fluid dynamic processes involved. The unsteady time-dependent high fidelity simulations required more than 300 gigabytes (GB) of data and more than 1 terabyte (TB) of I/O for advanced scientific visualization, pushing the limit of the capabilities of the CFD visualization software and requiring additional new advanced capabilities. This visualization work represents one of the largest data sets, requiring possibly the largest use of HPC resources at the U.S. Army Research Laboratory (ARL) Major Shared Resource Center (MSRC).

This capability has provided fundamental understanding of fluid dynamics mechanisms associated with the interaction of the unsteady synthetic jets and the projectile flow fields. Many flow field solutions resulting from the simulation of multiple spin cycles and, thus, a large number of synthetic jet operations, were saved at regular intermittent time intervals to produce animations to

⁴ParaView® is a registered trademark of Kitware, Inc.

gain insight into the physical phenomenon resulting from the synthetic jet interactions. The unsteady jets were discovered to break up the shear layer coming over the step in front of the base of the projectile. The flow separation point is delayed on the curved base surface near the jet. It is this insight that was found to substantially alter the flow field (making it asymmetric and highly unsteady) near the jet and in the wake region, which in turn, produced the required forces and moments even at 0 degree angle of attack (level flight). Time-accurate velocity magnitude contours (figure 2) confirm the unsteady wake flow fields arising from the interaction of the synthetic jet with the incoming free stream flow at Mach = 0.24. Figure 3 shows the particles emanating from the jet and interacting with the wake flow, making it highly unsteady. More importantly, the disintegration of the shear layer is clearly evidenced by the particles clustered in regions of flow gradients or vorticity. Verification of this conclusion is provided by the excellent agreement between the predicted (solid line) and measured (solid symbols) values of the net lift force attributable to the jet (figure 4). The net lift force (F_y) was determined from the actual time histories of the highly unsteady lift force (see figure 5 for an example) resulting from the jet interaction (jet is on and off during spin cycle) at 0 degree angle of attack and computed with the new hybrid RANS-LES turbulence approach. The computed lift force along with other aerodynamic forces and moments, directly resulting from the pulsating jet, can produce the desired course correction to divert the projectile from its flight path. The results showed the potential of CFD to provide insight into the jet interaction flow fields and provided guidance as to the locations and sizes of the jets to generate the maximum control authority required to maneuver a spinning munition to its target with precision.

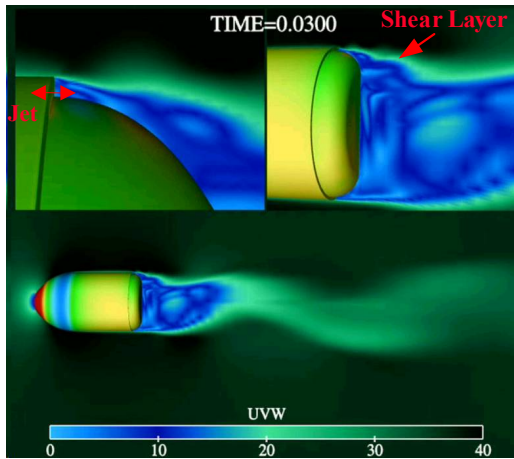


Figure 2. Computed velocity magnitude contours.

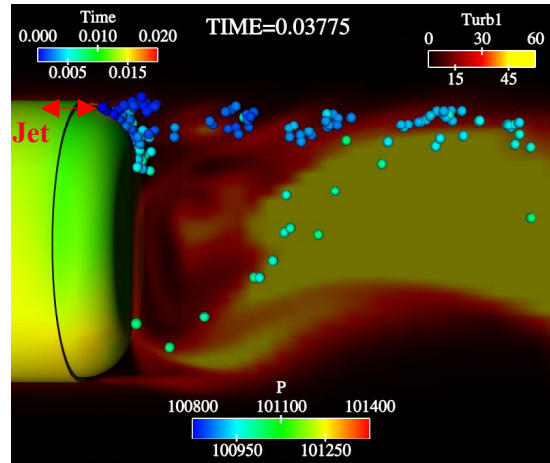


Figure 3. Instantaneous particle traces in the wake.

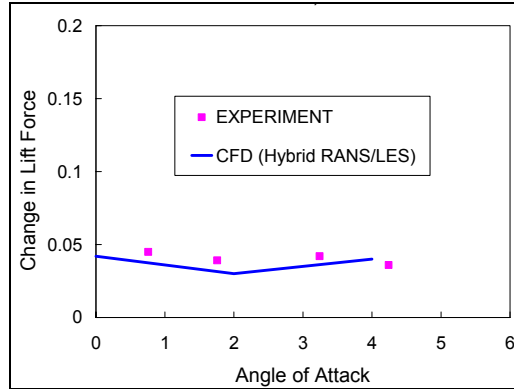


Figure 4. Lift force attributable to the jet versus angle of attack.

```
#!/bin/csh -f

set sahu_dir=/usr/var/tmp/sahu/cfd_rbd/40mm/channel_jet/body/sixdof_fs_lns_jl/coup
set output_dir=/usr/var/tmp/angel/JetOn
setenv MCFD_LICEXT 32849676

@ filenum=12241
set max_file=14040

cd $output_dir

while (1)
    echo $sahu_dir/pltosout.bin.$filenum
    # create symbolic links to necessary files in user directory
    foreach k ($sahu_dir/*.bin)
        ln -s $k $k:t
    end
    foreach k ($sahu_dir/*.inp)
        ln -s $k $k:t
    end
    rm pltosout.bin
    ln -s $sahu_dir/pltosout.bin.$filenum pltosout.bin
    genplif ensightb << EOF
1
1
EOF

    mv ensightb.geo    ensightb.geo.$filenum
    mv ensightb.P      ensightb.P.$filenum
    mv ensightb.R      ensightb.R.$filenum
    mv ensightb.T      ensightb.T.$filenum
    mv ensightb.Turb1   ensightb.Turb1.$filenum
    mv ensightb.Turb2   ensightb.Turb2.$filenum
    mv ensightb.U       ensightb.U.$filenum
    mv ensightb.UVW     ensightb.UVW.$filenum
    mv ensightb.V       ensightb.V.$filenum
    mv ensightb.W       ensightb.W.$filenum
    mv ensightb.case    ensightb.case.$filenum
    rm *.bin *.inp

    @ filenum+=1
    # are we finished yet?
    If ($filenum > 14040) then
        exit 0
    endif
end
```

Figure 5. Sample shell script used to convert CFD++ output to Enight binary format.

These time-accurate unsteady simulations (even with a single synthetic jet) require a large amount of resources. The unsteady CFD modeling technique required about 600 time steps to resolve a full spin cycle (67 Hz). The unsteady synthetic jet operates at a high frequency of 1000 Hz. Time-accurate CFD modeling of each jet cycle required more than 40 time steps. The actual computing time for one full spin cycle of the projectile was about 50 hours with 16 processors (i.e., 800 processor hours) on a SGI (formerly Silicon Graphics, Inc.), Origin 3000 or an IBM SP P3 system for a mesh size about 4 million grid points. Multiple spin cycles and, thus, a large number of synthetic jet operations were required to reach the desired periodic time-accurate unsteady result. Some cases were run for as many as 60 spin cycles requiring more than 48,000 processor hours of computer time for each case. These large-scale time-accurate aerodynamic computations required in total more than 500,000 processor hours on the SGI Origin 3000, IBM SP P3, and IBM SP P4 systems at the ARL MSRC, primarily with the CFD++ flow solver.

4. Methodology

For the purpose of this particular data set, we chose to use the Ensign visualization package. Ensign is a robust, fully functional, general purpose visualization package that is used to support the visual analysis of large, time-dependent data sets. Because of its architecture, Ensign works very well in a high performance computing environment. It uses a client-server model, where the client resides on a local graphics workstation and the server runs on a distributed system where the data are situated. With this methodology, large amounts of computed data do not need to be downloaded to a local workstation to be analyzed. The Ensign server is multi-threaded, therefore performing compute-intensive operations such as clip planes, isosurfaces, and particle traces in a timely manner. Only polygonal information is passed over the network between the server and client, therefore improving the efficiency of data analysis and evaluation. Ensign has the functionality and flexibility to be able to perform the complex visualizations required to accurately represent this data set.

It is not the intended purpose of this document to provide a complete tutorial about the use of the Ensign visualization package. However, for illustrative purposes, a description of some of the advanced, unique capabilities of Ensign and how they were applied to this problem is presented.

We describe a scripting technique for converting the CFD++ data to an efficient format accepted by Ensign. Then, a methodology for creating an axi-symmetric clip plane on a pitching and yawing munition is described, followed by the creation of vector arrows to show the dynamics of the flow field around the 40-mm munition. We then focus on the techniques that visually represent the interaction of the synthetic jet with the free stream flow around the projectile, including streamline and path line generation, and the use of passive scalars to isolate the effects of the jet thruster from the free stream.

4.1 Preparing Data for Visualization

The data files created by the CFD++ computation need to be converted to a format recognized by the Enight visualization software. A utility called “genplif” is available as part of CFD++ to perform data conversion. Genplif converts to the Enight format in both ASCII⁵ (command line parameter “ensight”) and binary format (command line parameter “ensightb”). The binary format is preferred since the I/O time to write and read the binary data is significantly faster than the ASCII format. The command to create an Enight binary data set from an existing CFD++ datafile is

```
prompt> genplif ensightb
```

Genplif is an interactive program that provides the user with a number of options for creating a visualization file. For the purposes of this visualization effort, the following technique was used to accelerate the process:

```
prompt> genplif ensightb << EOF
1
1
EOF
```

(NOTE: Before “genplif” is run, a number of environment variables need to be set; see appendix A for details.)

This standard Unix⁶ technique allows for two input strings to be entered before the interactive prompt requests input. Genplif is an interactive program, but in practice, genplif will be run in a batch mode to process multiple time steps of data.

The output from “genplif” is an Enight case file that contains metadata to describe the geometry and solution data. Genplif also creates a binary Enight geometry file and separate files for each vector and scalar field contained within the CFD++ data file. The 40-mm synthetic jet projectile data set is actually a time-dependent computation, with thousands of time steps. Therefore, an efficient technique for converting all the time steps was established (see figure 5).

Once all the data have been converted from CFD++ to Enight binary format, a master Enight case file was manually created to describe the entire time-dependent data set. The contents of this case file looks like figure 6.

A complete description of the Enight data formats, including case files, is presented in the Enight on-line documentation.

⁵American standard code for information interchange.

⁶Unix™ is a trademark of Bell Laboratories.

```

FORMAT
type: ensight gold
GEOMETRY
model: new.geo.***** change_coords_only
VARIABLE
scalar per node: Turb1 new.Turb1.*****
scalar per node: U new.U.*****
scalar per node: V new.V.*****
scalar per node: W new.W.*****

TIME

time set: 1
number of steps: 26
filename start number: 12494
filename increment: 1
time values:
0.5122883943
0.51233894575
0.5123894972
0.51244004865
0.5124906001
0.51254115155
0.512591703
0.51264225445
0.5126928059
0.51274335735
0.5127939088
0.51284446025
0.5128950117
0.51294556315
0.5129961146
0.51304666605
0.5130972175
0.51314776895
0.5131983204
0.51324887185
0.5132994233
0.51334997475
0.5134005262
0.51345107765
0.5135016291
0.51355218055

```

Figure 6. Ensign case file.

4.2 Clip Plane

The initial visualization developed from this complex 40-mm projectile data set was a basic flow field animation. A clip plane visualization of the data set over time provides a visual check to ensure that the data conversion process was successful and that the computation produced anticipated results.

For this first visualization, a clip plane was established down the centerline of the projectile. However, since the projectile had both horizontal and vertical motion (pitch, yaw) as it moved down the flight path, a computed position for the clip plane was required to maintain orientation along the centerline of the projectile.

Enight does provide a technique for maintaining this position in the moving mesh through the use of the following technique:

1. Select nodes at the tip of the projectile and the center of the base of the projectile. These nodes are used to position the clip plane and establish the orientation.
2. Create x, y, and z variables that corresponded to the node on the front of the projectile.
3. Create x, y, and z variables that corresponded to the node on the base of the projectile.
4. Create an average_x/y/z from these variables. This value becomes the clip origin.
5. Create a normal_x/y/z using the average in step 4 and the x,y,z variables in step 2. This will be the normal direction for the clip plane.
6. Create a clip plane and revise the plane tool origin/normal with the variables using the Enight command: “clip: update_to_newtoollocation” command.
7. Create a “while loop” over the available time steps. Inside the while loop, the time step is revised, the plane tool is adjusted, and finally the clip plane is revised. This technique effectively revises the clip part twice—once when the time is changed and once more when the plane tool location is revised, based on the orientation of the projectile.

Through this technique, each time the time step was revised, the clip plane remained centered on the projectile part as it moved through the mesh (see figures 7 and 8). The clip plane was then colored with a scalar or vector value, such as pressure, vorticity, turbulence, or velocity to represent the complex flow within the computational model. The complete command language script for this sequence is presented in appendix B.

4.3 Vector Arrows

The second visualization was performed to specifically evaluate the vector value velocity (typically represented by UVW). Vector arrows show the magnitude and direction of a vector variable within the mesh. In this case, vector arrows were randomly dispersed (seeded) throughout the computational mesh and visually inspected to determine if the results were plausible (figure 8).

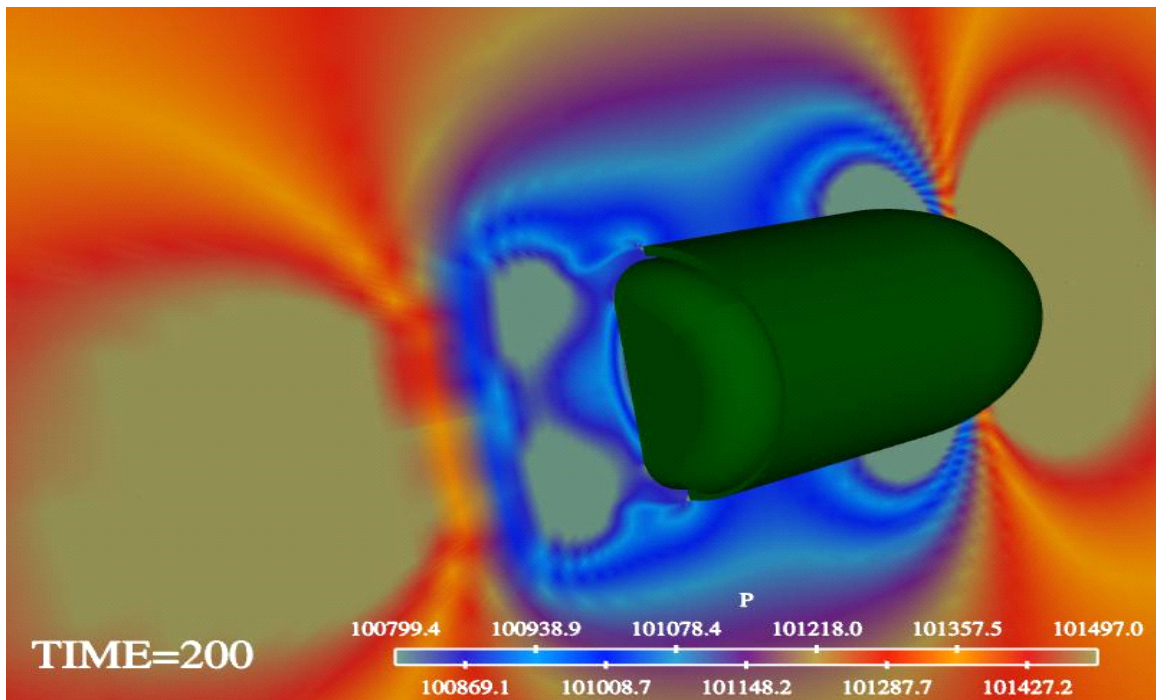


Figure 7. Clip plane colored with scalar value pressure.

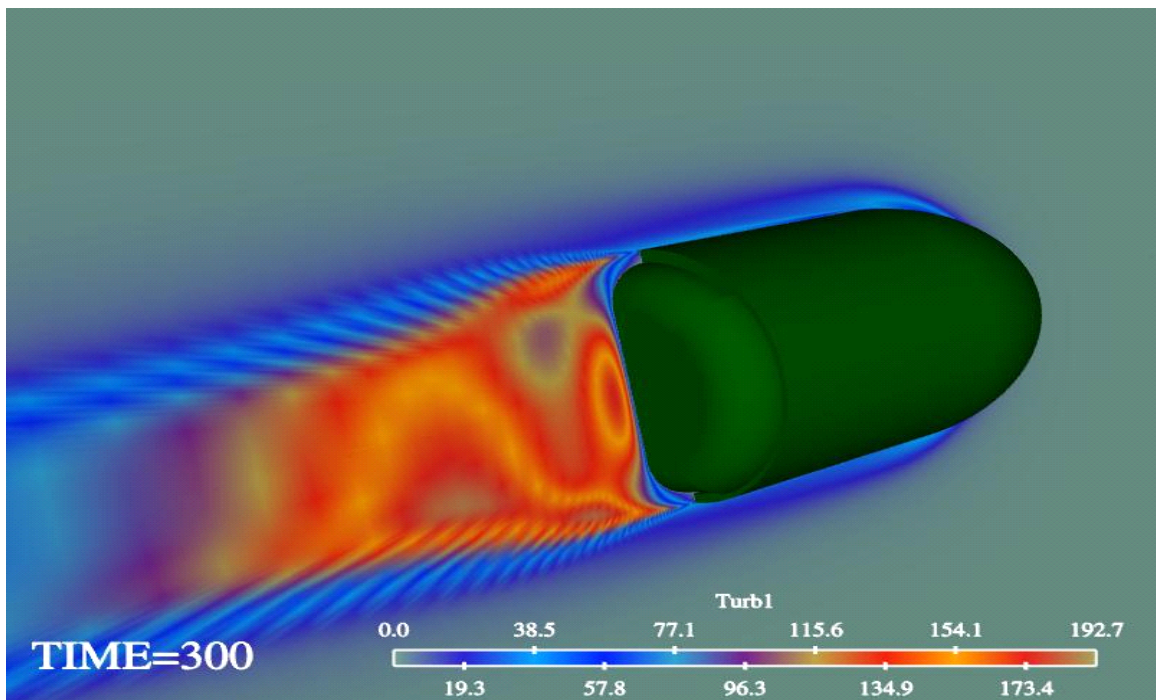


Figure 8. Clip plane colored with scalar value turbulence.

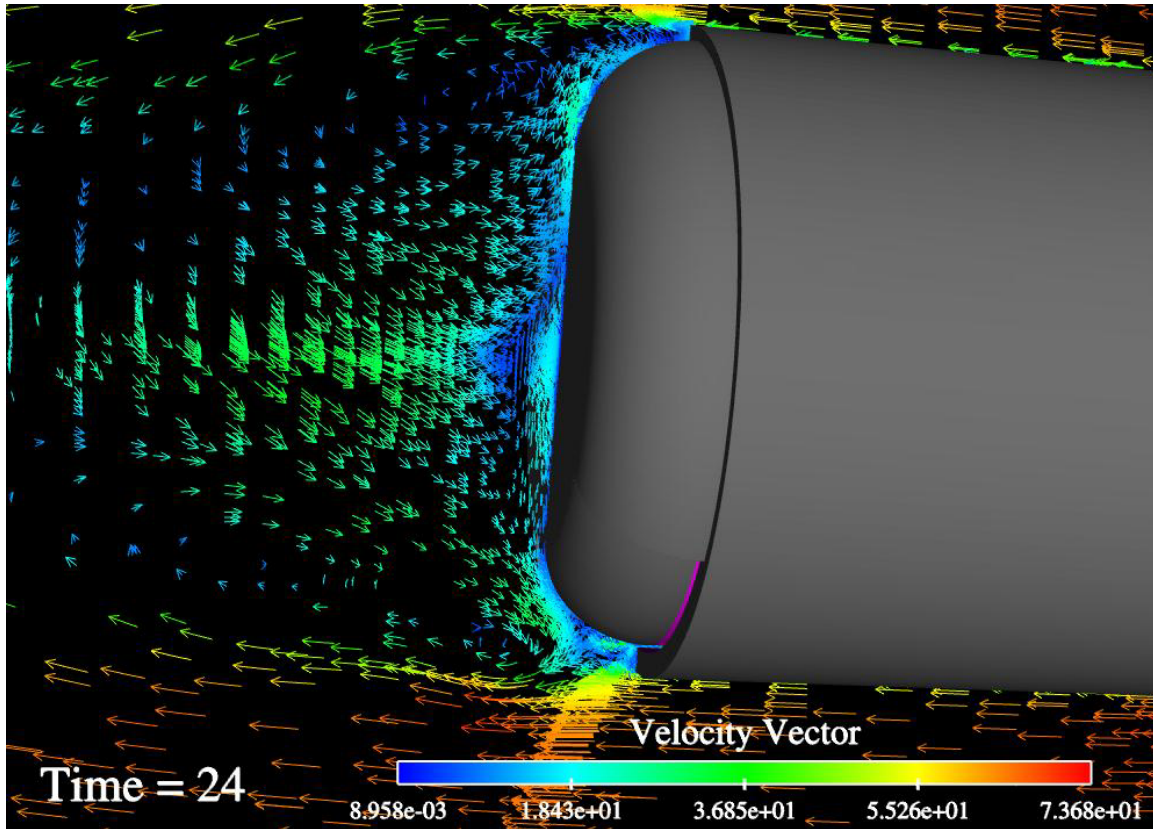


Figure 9. Vector arrows in base flow region of projectile (arrows colored by velocity).

4.4 Streamlines

In this example, the projectile rotates as it moves down the X-axis, and the CFD++ computation dumped a visualization file at 1-degree intervals, therefore providing 360 time steps representing one complete rotation of the projection. Dumping at 1-degree intervals provided adequate fidelity to represent the complex flow in and around the base flow region of the projectile as a result of the pulsing jet.

In order to demonstrate the presence of the pulsating synthetic jet and the effects of that jet, numerous techniques were evaluated to determine if they would allow for a concise visual representation of the physics. Isosurface generation and volume rendering techniques were employed; however, they could not clearly delineate the effects of the pulsating jet from the free stream flow around the body of the munition. It was determined that particle traces emitted from inside the diaphragm would depict the presence of a jet pulse during the course of the rotational period of the munition.

A series of streamlines was animated to show the rotation of the projectile over time. The technique used to create the streamline was to establish a set of seed points in the computational mesh “inside” the pulse jet diaphragm. We randomly selected a set of nodes inside the slot jet by temporarily cutting away all the mesh outside this region, allowing for visual selection of nodes.

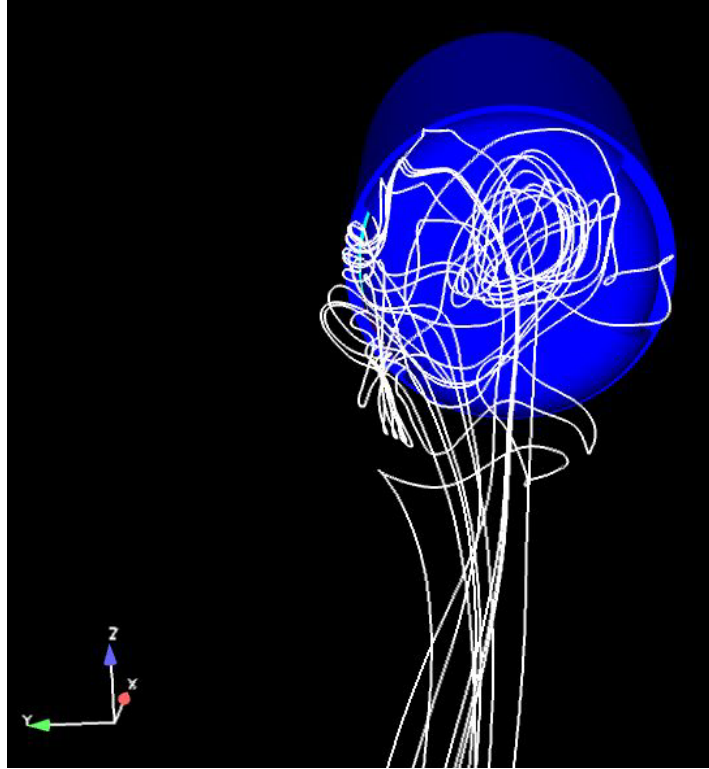


Figure 10. Streamline animation during “pulse” emission.

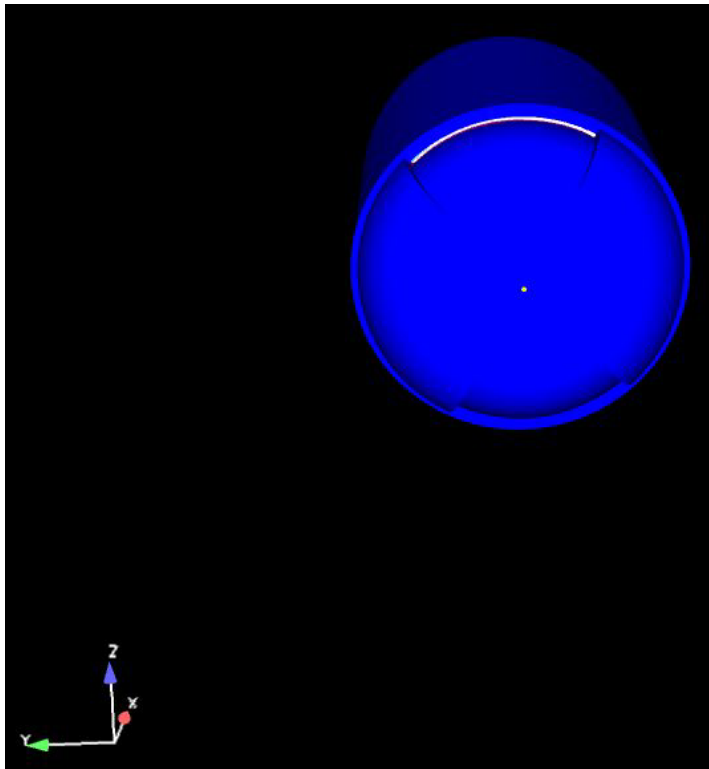


Figure 11. Streamline animation when the jet is not pulsing.

Once a set of nodes was identified and selected, those nodes were used to create an Enight subset part, from which path lines and streamlines can be seeded. This technique was chosen so that the streamlines could be seeded over time in a consistent manner. For instance, creating a streamline or path line from a rake/line tool or from XYZ coordinates in space would not be effective since it would not follow the rotation of the object. Placing the particle trace seeds “inside” the jet slot region at specific nodes reduces the chance of picking up velocity values from the free stream flow around the projectile. By placing the seeds inside the slot, we expect the results to be that the particles’ seeds will generate path lines only when the jet pulses (see figures 10 and 11).

4.5 Path Lines

The most complex visualization performed on this data set was the generation of path lines. Path lines differ from streamlines in that path lines are particle traces “over time,” while a streamline is a particle trace at a particular instance in time. Path line generation is computationally expensive in that the calculation is required to step through the entire data set to complete the computation. Because of the complexity of this data set, significant additional steps needed to be completed before the path line computation.

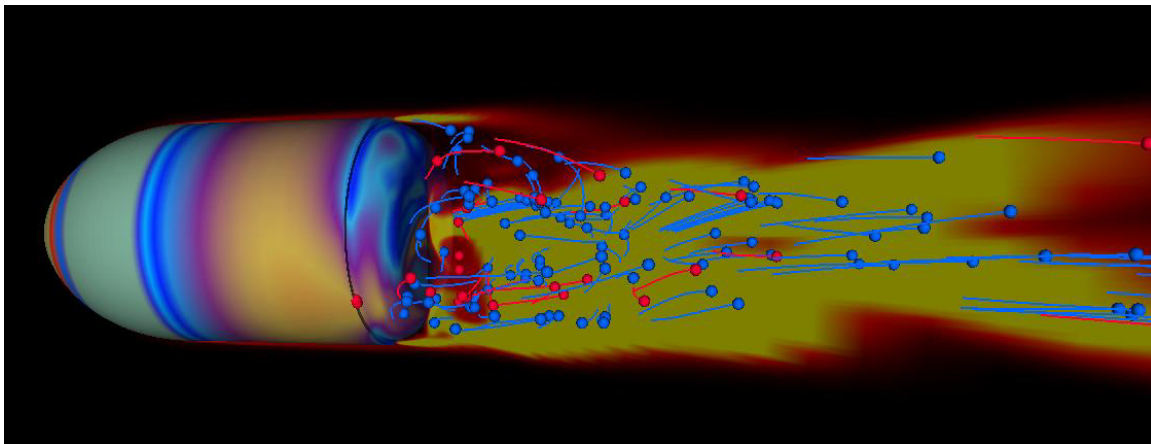


Figure 12. Path lines indicating those particles emitted during the first (red) cycle of the projectile and during the second (blue) cycle.

Because of the nature of this data set and the movement of the projectile object within the mesh, it was necessary to create an emitter file that contained the node point position at each time step for the subset part created to support the emitter position. The process for creating the emitter file is

1. Create a subset part that contains the emitter nodes (same part that was use to create the streamlines in the previous animation is sufficient)
2. Select this subset part in the Enight GUI, and issue the following command:

```
test: build_emitter 0 100 1 emitter.file
```

This command will build an emitter file, starting at time 0, ending at time 100, with an increment of 1. The emitter information will be dumped to a file called “emitter.file”.

3. Create path lines using an emit type of “file” and use the emitter.file as the input to this procedure. Ensign will churn through the data and create pathlines emitting from locations described in the file generated in the previous step.
4. Load a flipbook. This will create an animation of the visible objects over time.
5. Enable particle animation to display the flow of path line.

There were significant problems in completing the path line animations for this data set. After working with the vendor to explore a number of technical issues, we discovered that a change in the Ensign case file improved performance and algorithmic efficiency. The change in the case file was the following:

```
model: new.geo.***** change_coords_only
```

The use of the flag “change_coords_only” in the case file told Ensign that while this was a time-varying data set with a new geometry file at each time step, only the coordinates of the computational mesh changed, not the connectivity of the mesh. This significantly reduced the amount of I/O required to move between time steps and resolved an issue with the path line algorithm. The result of this technique can be seen in figure 12.

4.6 Passive Scalar

In an effort to isolate the fluid flow associated from the jet thruster from the free-stream airflow, a passive scalar variable was calculated. The passive scalar allowed the computation to trace the effects of the pulsating jet without influencing the dynamics of the free-stream airflow around the munition. Conceptually, the passive scalar simulates the injection of a dye that is then propagated by the local fluid without changing the fluid itself. When the “dye” (passive scalar) is visualized, the fluid emanating from the jet can be tracked. The passive scalar is a continuous variable ranging from 0 to 1, where a value of 1 indicates the presence of the influence of the jet thruster, and a 0 value indicates that the jet is off. In the CFD computations themselves, the passive scalar is an extra variable and an extra equation is solved in addition to the traditional continuity, momentum, energy, and turbulence equations. The actual flow variables and the calculated flow fields remain totally unaffected by the addition of this new variable. Initially, the passive scalar variable is set to 1 for the jet thruster and set to 0 everywhere else in the flow domain. As the calculation progresses, the scalar variable is allowed to change only in the vicinity of the jet thruster because of flow convection, etc., but the boundary condition at the jet thruster is always set to 1 for the passive scalar. When the passive scalar was calculated, a value was thus made available to the visualization process to more easily demonstrate the presence of the forces associated with the pulsating synthetic jet (see figure 13).

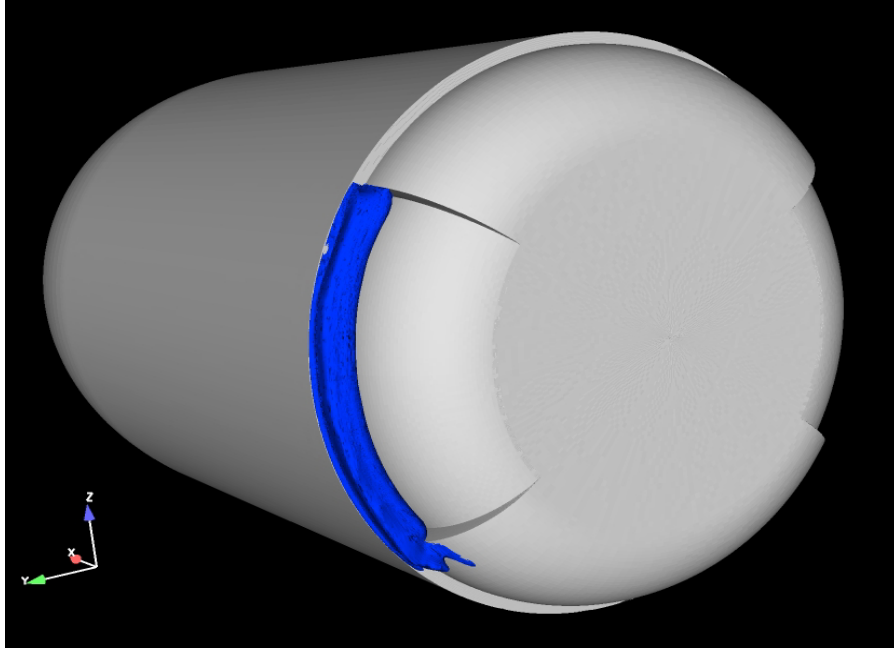


Figure 13. Blue isosurface representing the passive scalar calculated for the pulsating synthetic jet.

5. Conclusion

We have presented an efficient method for visualizing a large, DoD high performance computing (HPC) Challenge⁷ data set calculated with CFD++. Ensign provided a unique environment for analyzing the computational results while minimizing the movement of the large, time-dependent data set from the HPC platforms where the computation was run. Numerous challenges were associated with this data set, such as creating particle traces from a moving, rotating object, which were extremely difficult to accomplish. However, through the innovative use of Ensign's scripting language and other advanced features, we were able to develop the methodologies necessary to perform these complex visualizations.

This work has provided an increased fundamental understanding of the complex, 3-D, time-dependent, aerodynamic interactions associated with micro-jet control for yawing spin-stabilized

⁷The HPC Modernization Program allocates approximately 25% of its HPC resources each fiscal year to competitively selected DoD Challenge projects. These computationally intensive, high priority projects are selected annually through a rigorous technical and mission relevance evaluation. Senior scientists and engineers within defense science and technology and test and evaluation organizations, universities, and industry research partners head these high priority projects. Services and agencies allocate the remaining resources through service and agency unique processes. Challenge project efforts produce and support key enabling technologies, capabilities, and demonstrations expressed by the defense technology objectives (DTOs). These enabling DTOs, in turn, support Joint Vision 2020 and the 13 joint war-fighting capability objectives promulgated by the Joint Requirements Oversight Council of the Joint Chiefs of Staff.

munitions. Detailed flow physics simulations have captured all the flow structures with high fidelity and successfully identified the locations of synthetic micro-jets for optimum aerodynamic interference and control authority.

6. References

- Peroomian, O.; Chakravarthy, S.; Palaniswamy, S.; Goldberg, U. *Convergence Acceleration for Unified-Grid Formulation Using Preconditioned Implicit Relaxation*. AIAA Paper 98-0116, 1998.
- Batten, P.; Goldberg U.; Chakravarthy, S. *Sub-grid Turbulence Modeling for Unsteady Flow with Acoustic Resonance*. AIAA Paper 00-0473, 38th AIAA Aerospace Sciences Meeting, Reno, NV, January 2000.
- Sahu, J. *Unsteady Numerical Simulations of Subsonic Flow over a Projectile with Jet Interaction*. AIAA Paper 2003-1352, Reno, NV, 6-9 January 2003.
- Sahu, J. *Unsteady CFD Modeling of Aerodynamic Flow Control over a Spinning Body with Synthetic Jet*. AIAA Paper 2004-0747, Reno, NV, 5-8 January 2004.

Appendix A. Metacomp Environment Variables

```
setenv METACOMP_HOME /home/army/sukumar/linux/distrib
setenv MCFD_HOME /home/army/sukumar/linux/distrib/mlib/mcfd.4.1.1
setenv MCFD_VERSION 3.5.1
setenv MCFD_MAXMEM 8G
setenv MCFD_PROCMEM 2G
setenv MCFD_TCLTK
/home/army/sukumar/linux/distrib/mlib/mcfd.4.1.1/exec/gui_src
setenv MCFD_HTML /home/army/sukumar/linux/distrib/mlib/mcfd.4.1.1/html

setenv MCFD_TOGL yes
setenv TCL_LIBRARY /home/army/sukumar/linux/distrib/mlib/tcltk8/lib/tcl8.0
setenv TK_LIBRARY /home/army/sukumar/linux/distrib/mlib/tcltk8/lib/tk8.0

set mpath = (/home/army/sukumar/linux/distrib/sbin/ \
/home/army/sukumar/linux/distrib/mlib/mcfd.4.1.1/exec)

set path = ($mpath $path)
```

INTENTIONALLY LEFT BLANK

Appendix B. Sample Ensign Command File for Creating an Object-Centered Clip Plan

```
VERSION 7.64
data: binary_files_are big_endian
data: format case
data: path /scratch/data/support/ar1/pathline_work
data: geometry cei5.case
data: start_time 0
data: read
data_partbuild: begin
data_partbuild: data_type unstructured
data_partbuild: select_begin
  1 2 3 4
data_partbuild: select_end
data_partbuild: description
data_partbuild: create
data_partbuild: end
view_transf: rotate 7.131538e+00 4.657800e+01 0.000000e+00
part: select_begin
  3
part: select_end
part: modify_begin
part: visible OFF
part: modify_end
part: select_begin
  4
part: select_end
part: modify_begin
part: visible OFF
part: modify_end
part: modify_begin
part: visible ON
part: modify_end
variables: evaluate node1_x = Coordinates[x][1]
part: select_begin
  4
part: select_end
variables: evaluate node1_y = Coordinates[y][1]
part: select_begin
  4
part: select_end
variables: evaluate node1_z = Coordinates[z][1]
part: select_begin
  4
part: select_end
variables: evaluate node211_x = Coordinates[x][211]
part: select_begin
  4
part: select_end
variables: evaluate node211_y = Coordinates[y][211]
part: select_begin
  4
```

```

part: select_end
variables: evaluate node211_z = Coordinates[z][211]
part: select_begin
4
part: select_end
variables: evaluate midx = (node1_x+node211_x)/2.
part: select_begin
4
part: select_end
variables: evaluate midy = (node1_y+node211_y)/2.
part: select_begin
4
part: select_end
variables: evaluate midz = (node1_z+node211_z)/2.
part: select_begin
4
part: select_end
variables: evaluate normal_x = node1_x-midx
part: select_begin
4
part: select_end
variables: evaluate normal_y = node1_y-midy
part: select_begin
4
part: select_end
variables: evaluate normal_z = node1_z-midz
view_transf: function plane
tools: plane ON
view_transf: plane_origin $midx $midy $midz
view_transf: plane_normal $normal_x $normal_y $normal_z
part: select_all
clip: begin
clip: domain intersect
clip: tool plane
clip: end
clip: create
part: select_begin
1
part: select_end

#
# modify the clip plane to the plane tool's location
#
# The "clip: update_to_newtoollocation" command
# will update the clip to the tools (in this case the plane tool) # current
location #

part: modify_begin
clip: update_to_newtoollocation
part: modify_end

#
# set up to record jpeg images at 800x600 resolution
#

file: image_format jpeg
file: image_to_printer OFF

```

```

file: image_window_size user_defined
file: image_window_xy 800 600

#
# march through the timesteps that exist - my case I have 5
# When time is changed everything will update - including the clip # plane.
# However, the clip plane is not in the correct location and # has to be
# updated (the midx/y/z and normal_x/y/z constants would have # upated to new
# values after the time change). After everything updated # save an image
# file. Note that you will not be able to create a movie # format using this
# technique - you can only get one frame at a time. #

$int timestep
$timestep = 0

while ($timestep < 5)

solution_time: current_step $timestep
solution_time: update_to_current
view_transf: plane_origin $midx $midy $midz
view_transf: plane_normal $normal_x $normal_y $normal_z
part: modify_begin
clip: update_to_newtoollocation
part: modify_end
#
# save an image file. The $3timestep will create a file name
# containing 3 digits from the $timestep variable, i.e.,
# /tmp/clip_test000.jpg, /tmp/clip_test001.jpg, etc.
#
file: image_file /tmp/clip_test$3timestep
file: save_image
$timestep = $timestep + 1

endwhile

```

NO. OF
COPIES ORGANIZATION

- * ADMINISTRATOR
DEFENSE TECHNICAL INFO CTR
ATTN DTIC OCA
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218
*pdf file only
- 1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN IMNE AD IM DR MAIL & REC MGMT
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- 1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI OK TECH LIB
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- 1 US ARMY HIGH PERFORMANCE RSCH CTR
ATTN ANDREW JOHNSON
1200 WASHINGTON AVE S
MINNEAPOLIS MN 55415
- 1 MISSISSIPPI STATE UNIV ERC
ATTN R J MOORHEAD
PO BOX 9627
MISSISSIPPI STATE MS 39762-9627
- 1 AERONAUTICAL SYSTEMS CTR
ATTN RHONDA VICKERY
HIGH PERFORMANCE COMPUTING BR
ASC/HP BLDG 676 2435 FIFTH ST
WRIGHT PATTERSON AFB OH 45433-7802
- 1 METACOMP TECHNOLOGIES INC
ATTN SUKUMAR CHAKRAVARTHY
28632 ROADSIDE DRIVE SUITE 255
AGOURA HILLS CA 91301
- 2 RAYTHEON SYSTEMS
ATTN M A BOLSTAD
J C RENTERIA
939 I BEARDS HILL RD PMB 191
ABERDEEN MD 21001

ABERDEEN PROVING GROUND

- 1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI OK (TECH LIB)
BLDG 4600

NO. OF
COPIES ORGANIZATION

- 1 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI C NIETIUBICZ
BLDG 328
- 6 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL CI HC J CLARKE
R NAMBURU R ANGELINI
R NOACK R SCHUMEYER
W TAUBER
BLDG 394
- 7 DIRECTOR
US ARMY RSCH LABORATORY
ATTN AMSRD ARL WM BC J SAHU
P PLOSTINS S SILTON
K HEAVEY J DESPIRITO
P WEINACHT B FORCH
BLDG 390