

Progress towards a Microgravity CFD Validation Study using the ISS SPHERES-SLOSH Experiment

Jedediah M. Storey¹, Dr. Daniel Kirk²
Florida Institute of Technology, Melbourne, FL, 32901

Brandon Marsell³ and Dr. Paul Schallhorn⁴
NASA KSC, FL, 32899

Understanding, predicting, and controlling fluid slosh dynamics is critical to safety and improving performance of space missions when a significant percentage of the spacecraft's mass is a liquid. Computational fluid dynamics simulations can be used to predict the dynamics of slosh, but these programs require extensive validation. Many CFD programs have been validated by slosh experiments using various fluids in earth gravity, but prior to the ISS SPHERES-Slosh experiment¹, little experimental data for long-duration, zero-gravity slosh existed. This paper presents the current status of an ongoing CFD validation study using the ISS SPHERES-Slosh experimental data.

I. Introduction

Advancements in in-space, propellant storage, management, and transfer science and technologies are key to increasing safety, decreasing cost, and increasing payload mass of NASA's space missions. Since propellant usually makes up a large portion of a spacecraft's mass, predicting and controlling the motion of it is important. CFD programs are critical to predicting slosh dynamics and finding ways to mitigate these concerns, but CFD programs are complex and require extensive experimental validation before the results can be trusted.

Little experimental data for long-term zero- or micro-gravity slosh exists. This is mainly due to the costs associated with obtaining such data. While many reduced gravity drop tower experiments have been conducted, the short duration of these tests makes their use in CFD validation limited. Some long duration micro-gravity slosh testing has been conducted using sounding rockets [1] [2], parabolic flight profiles in an aircraft [3], orbital launch vehicles [4] [5], and spacecraft [6] [7]. The examples referenced here are not a complete list, but most of these experiments either involve imprecise low gravity, e.g. sounding rocket and parabolic flight aircraft, or a lack of high quality data and imagery (from modern sensors), e.g. the 1960's and 1970's tests. The ISS SPHERES-Slosh experiment [8] was designed to provide long duration, microgravity, coupled motion, high fidelity slosh data. The SPHERES-Slosh experiment consists of two MIT SPHERES robots attached to a frame that holds a small, pill-shaped tank partially filled with green-dyed distilled water. The SPHERES or an astronaut can push the experiment around inside of the ISS to simulate various launch vehicle upper stage or spacecraft maneuvers. Four inertial measurement units (IMU) capture six degree of freedom (6DoF) acceleration and rotation data, while two cameras capture high definition images of the fluid in the tank. The following figure is a picture of the experiment with the tank exposed.

¹ Graduate Student, Mechanical and Aerospace Department

² Professor, Mechanical and Aerospace Department

³ Fluids/CFD Engineer, Launch Services Program, NASA Kennedy Space Center

⁴ Chief, Launch Services Program, NASA Kennedy Space Center, VA-H3



Figure 1. ISS SPHERES-Slosh Experiment

The ultimate goal of this research effort is to validate a CFD program for microgravity water slosh using as much of the SPHERES-Slosh data as possible. The paper begins with a discussion of the data pipeline that processes the IMU data and outputs a tank trajectory. Next, the CFD methodology is presented. Qualitative comparisons between experiment images and images from the current CFD are made. Finally, conclusions and future work are discussed.

II. The SPHERES-Slosh Data Pipeline

A. Experimental Data

The blue SPHERE is considered the primary (“A” side), while the red SPHERE is considered the secondary (“B” side). The SPHERE local coordinate system’s origin is at the geometric center, with +x pointing in the direction of the expansion port (Slosh avionics box), +z in the direction of the pressure regulation knob, and +y completing the right-hand system. The two SPHERES’ x axes are pointing at each other and aligned with the center axis of the tank. The coordinate system used in the CFD for this project is the primary, A-side SPHERES’ coordinate system translated in +x to the tank center; this will be referred to as the “CFD body frame” in this paper.

The experiment had four IMUs: one in each SPHERE and one in each Slosh avionics box. The SPHERES accelerometers and gyroscopes have lower ranges than the Slosh ones, but are also less noisy. The Slosh IMUs are single package format, while the IMUs in the SPHERES are actually distributed single axis accelerometers and gyroscopes. The locations of these sensors are summarized in Tables 1 [9] and 2 (measured from CAD):

Table 1. SPHERE Accelerometer Locations

Sensor	Location in local SPHERE body frame: X,Y,Z [cm]
x-axis accelerometer	5.19, 2.17, 3.27
y-axis accelerometer	-2.66, 3.35, 3.30
z-axis accelerometer	3.28, -4.27, 3.35

Table 2. Relevant Locations in CFD body frame: X,Y,Z [cm]

A-side slosh IMU	-23.6, 4.9, -2.5
B-side slosh IMU	23.6, -4.9, -2.5
SPHERE A center	-42.0, 0.0, 0.0
SPHERE B center	42.0, 0.0, 0.0

The data is structured as follows. Each test session has a Science number, and each test has an A and B side folder. Inside of those are “run” folders for each test, and inside those are a folder for data and a folder for images. The IMU data from the SPHERE is saved to an “imu_data.txt” file, while the IMU data from the SLOSH avionics box is saved to a “slosh_imu.txt” file. Each data set was saved in the local IMU’s coordinate system, so it had to be rotated and transformed.

Note that the SPHERES metrology system was not used for this experiment because the experiment often traveled outside of the volume defined by the beacons. Thus, no absolute position data is available.

There are two 5MP machine vision cameras looking at the tank, 90 degrees apart. These take high definition bitmap images at a variable rate of about 0.5-2 fps. Knowing their locations is important for accurately recreating the correct views during CFD post-processing. Table 3 contains the locations of the cameras’ image sensors in the CFD body frame (measured from CAD).

Table 3. Camera Locations: X,Y,Z [cm]

A-side Camera	0, -17.4, 17.4
B-side Camera	0, 17.4, 17.4

One of the primary challenges with working with these data sets is the fact that no clocks were synchronized. In other words, the A side SPHERE and B side SPHERE were on different clocks, but so were the A side SPHERE and SLOSH avionics box, as well as the A side camera. This made aligning the data signals in time difficult and aligning the images to the data imprecise. More data challenges and their solutions are presented in Table 5.

Selecting cases for simulation was time consuming. Some tests were missing some or all data and/or images; these were excluded. The motion in many was too low to induce significant sloshing. Some had non-steady initial conditions; these had to be excluded because the data pipeline trajectory computation algorithm requires steady initial conditions. A variety of maneuvers was desired. The cases in the following table were the final selections.

Table 4. Selected Test Cases

Science Mission	Test Number	A-side Folder	B-side Folder	Maneuver Description
2	11	run_2014_06_18_16_34_33	run_2014_06_18_16_28_08	x-axis periodic translation
2	13	run_2014_06_18_16_44_23	run_2014_06_18_16_37_58	y-axis periodic translation
3	4	run_2014_09_09_11_37_51	run_2014_09_09_11_30_39	single push along +x axis
3	16	run_2014_09_09_12_29_35	run_2014_09_09_12_22_25	spin about +x axis

The IMU data is used to recreate the 6DoF trajectory of the experiment, which is then used as an input for the meshed tank in the CFD. Rendered videos of the CFD fluid surface are then compared to videos created from the experiment images to qualitatively assess agreement. Another option might be to use the SPHERES’ thruster firings as force inputs to the CFD. Then the coupled slosh-motion 6DoF accelerations and rotations (or trajectory) could be compared to the experimental data. However, the SPHERES-actuated cases did not excite significant fluid, so all cases considered for this project have been astronaut-actuated. Since the forces and moments applied by the astronaut are unknown, the second method cannot be done.

B. Pipeline

The data pipeline was created in MATLAB [10]. It reads, interpolates, and filters the individual IMU data sets. Since the IMUs are not located at the center of the tank, the accelerations have to be transformed to the tank center (the body axes origin). Most inertial navigation systems have an external absolute position/orientation correction, e.g. GPS, because of cumulative integration error. Since no absolute reference was available for corrections, computing the position and orientation of the tank in time is purely inertial/dead reckoning. External videos are used to roughly check the predicted trajectories. Despite all attempts to reduce it, the cumulative error from dead reckoning often results in noticeably incorrect positions by approximately 30 s, depending on the amount of motion (particularly rotation) in the test.

The following table summarizes the challenges faced while writing the data pipeline, their consequences, and how they were addressed.

Table 5. Data Pipeline Challenges

Challenge	Consequences	Resolution
Some tests missing spheres data and images for A and/or	Difficult to find which A side test folders correspond to B side test folders; cannot use some tests	Manually find which tests have data problems and do not use them.

B side; sometimes data folders were not written		
No clocks synchronized (A/B side SPHEREs, A/B side SLOSH, cameras all on different clocks)	Difficult to find which A side test folders correspond to B side test folders; time alignment difficult	Manual folder corresponding; time alignment algorithms; data time-alignment algorithms; video alignment must be done manually by eye
Non-constant time steps for data	Automated file reading difficult, computations difficult	Flexible reading scripts; resample and interpolate
Low (20-30Hz) sample rate	Aliased signals; noise	Resample and interpolate; filtering
SLOSH IMU file format has both raw and scaled, gyro and accelerometer data in randomly changing order	Automated file reading difficult; separate time vectors; many text reads necessary	Flexible reading script; time alignment algorithms; parallel reading loops
Missing data points, negative time steps	Automated file reading difficult	Flexible reading scripts; resample and interpolate
SLOSH signal dropouts	Large (up to 2s) apparent time step due to pause in data writing. Data during that pause is written in a burst of short (<1msec) time step data after pause. Causes unrealistic results after interpolation.	Signal dropout detection algorithm and warnings; redistribution correction algorithm for compressed data
SPHERE signal dropouts	Large (up to 3s) apparent time step. Likely due to CPU overloading; data lost. Causes unrealistic results after interpolation.	Signal dropout detection algorithm and warnings. Gyro and accelerometer merging algorithms do not use SPHERE data in signal dropout regions.
SPHEREs accelerometers have lower acceleration limit than SLOSH accelerometers	SPHEREs accelerometers saturate during high acceleration maneuvers	Detect saturation and warn user. Accelerometer merging algorithm does not use SPHERE data in saturated regions.
SPHERE gyros have lower rotation rate limit than SLOSH gyros	SPHEREs gyros saturate during high rotation maneuvers	Detect saturation and warn user. Gyro merging algorithm does not use SPHERE data in saturated regions.
SLOSH accelerometer and gyro noise. Higher saturation limits than SPHEREs, but higher signal-to-noise ratio.	High integration error leads to nonsense results (experiment ending up in space)	Filtering; defaulting to SPHEREs accelerometers when not at or near saturation.
All signal channels have different offsets	Must offset all channels to zero, which requires no motion initial conditions.	Offsetting and amplitude alignment algorithms. Cannot use tests that have motion during first few seconds.
A/B side gyro/accel X/Y/Z (all 24 channels) were not necessarily aligned in time	High computation error; nonsense results	Time alignment algorithms
SLOSH IMUs not at center of tank	Cannot simply average the accelerometer data	3D kinematic transformations to center of tank
SPHERE IMUs are made of distributed single axis sensors not at center of tank	Cannot simply average the accelerometer data	3D kinematic transformations of individual axes to center of SPHERE, then all to center of tank
IMUs in body axes	Must be transformed to inertial frame for position and orientation calculations	Direction Cosine Matric (DCM) – based integration algorithm
Camera frame rates are low (typically <2Hz)	Cannot see flow feature development for some slosh events. Time alignment difficult.	None.

Camera frame rates are variable	Cannot simply write all images to a movie at a constant frame rate	Automated video writing script that repeats images to achieve a real time video.
Experiment leaves box defined by sonic beacons; hardware pin error turns off beacon detection. No gravity reference vector.	No absolute position information: cannot use sensor fusion/correction algorithms. Resulting position and orientation after some time (~30s) unreliable.	Dead reckoning/inertial navigation algorithm for position and orientation. Must use SPHEREs sensors where possible and filters to minimize accumulating error, which is ultimately unavoidable.
Error in inertia calculations during experiment design or setup	Center of tank not center of rotation: slight +Z component. X-rotations (roll) results in geometric center of tank following a circle.	Nothing. Calculations must be accurate enough to resolve the rotation of tank center in the Y-Z plane.

C. Experimental Video Creation

Because the image framerate was variable (usually between 0.5 and 2 frames/s), a custom real-time video creation script was written in MATLAB. It works by repeating a frame an appropriate number of times before calling up the next frame. If the MATLAB video frame rate was 25 fps, then the maximum frame time error of this method is approximately $0.5/25 = 0.02$ s. The error introduced from attempting to manually align CFD and experiment videos is about an order of magnitude higher (see Section IV). The time differences between A and B side images is on the order of the CFD-experiment alignment error, or about +/- half the cameras' true frame rates (which were variable).

III. CFD Approach

All simulations were performed using OpenFOAM [11]. This section will briefly describe some of the initial validation and error work, the final mesh and simulation settings, and the post-processing.

A. Initial Work

Before any simulations with motion were attempted, OpenFOAM's surface tension and static contact angle implementations were checked against SE-FIT [12], a minimum surface energy calculator, and STAR-CCM+ [13]. Starting with initially flat fluid interfaces in OpenFOAM and STAR-CCM+, transient simulations were run until surface oscillations were within one cell height. Static fluid surfaces from all three programs were extracted for multiple contact angles and compared in MATLAB. The results for each contact angle were all within one cell height of each other.

The "real" contact angle was difficult to obtain. Contact angle experiments with a sample of the same 3D printed material used in the tank and deionized water droplets were performed at FIT. A MATLAB program was written to measure the contact angle from high resolution images, and average repeated cases. The results are a static contact angle of 62.4° , advancing contact angle of 66.8° , and receding contact angle of 33.7° . Overall uncertainty on all contact angle measurements is approximately +/- 10° . The ISS SPHERES-Slosh Experiment images reveal a thin film of fluid coating the inside surfaces of the tank when the tank is being accelerated below some unknown threshold. This fluid film decreases the contact angle significantly. The same MATLAB program was used to estimate the modified contact angle from multiple experiment images (prior to motion). The result was a static contact angle of about 28° . This was used in all "real" CFD cases because it appeared to give the best initial fluid distribution. Unfortunately, accurate fluid film formation could not be recreated in the CFD. If the static contact angle was reduced to 15° or below, then a (thick) fluid film would form and coat the tank walls. However, due to the thickness of this film, the initial fluid distribution did not agree well with the initial distribution in the experiments, so the fluid film was neglected in the CFD. Finer meshes were checked and resulted in thinner films, but a high enough resolution mesh for (qualitatively) accurate film formation was not tested due to computational constraints. Another potential improvement would be to use a dynamic contact angle model coupled with a fluid film model, but this has not been investigated by the authors.

Two options for obtaining an initial condition for the fluid surface were tried. Option 1 involves initializing the fluid to be roughly in the correct location, running a transient, no motion simulation in OpenFOAM for about 60 s, then saving the final solution. The final solution is then used as the initial solution for all cases with that initial fluid distribution, e.g. evenly split or all on one side. Option 2 uses SE-FIT to generate the initial condition. This is usually faster and gives a more axisymmetric surface. However, SE-FIT can be difficult to use and sometimes has trouble converging. This method requires extracting the fluid surface in Paraview and then applying it to an OpenFOAM case using a topoSetDict in setFields. After trying both options extensively, option 1 is preferred.

During initial/practice CFD test cases, extraneous fluid surface and force oscillations were observed. These oscillations were traced to three sources: 1. parasitic currents due to surface tracking scheme, 2. numerical instabilities, and 3. low precision tabulated motion data. The parasitic currents were only present with no motion and less diffusive, i.e. 2nd order, numerical schemes. When motion is added, and the fluid dynamics become inertial dominated, these currents become negligible. Numerical instabilities were removed with careful selection of schemes and settings. The tabulated trajectories for the initial test cases were generated with OpenFOAM's 6DoF generator, which writes values with 6 digit precision (default C++ stream operator precision). Due to the incompressibility assumption and the slight inaccuracies introduced by using only 6 digit precision for position input, the resulting force measurements were "noisy". When the motion table was generated with 12 decimal point precision, all noise in the force waveforms was eliminated. The trajectories derived for the real cases were written with 12 decimal point precision.

B. Settings

The mesh was created using snappyHexMesh, OpenFOAM's built-in mesher. It is hexahedral dominant with prism layer cells along the wall and has a smooth transition from the wall layer cells to the core mesh. Cell count is approximately 800,000. A mesh independence study was attempted using a 1DoF sinusoidal motion test case and meshes of 800,000, 2.4M, and 6M cells. Force in the axial direction and images of the fluid surface were used to compare the cases. It was expected that the smallest mesh case would begin to differ from the medium and large mesh cases first, and then the medium case would begin to differ from the large mesh case a few seconds later. However, all cases began showing differences starting around 6s (first fluid impact due to direction reversal), which means the mesh study was inconclusive. The smallest mesh was selected to reduce simulation times.

The following settings were used for all simulations: second order accurate time and space formulations, PIMPLE solution scheme, multiphase volume-of-fluid (VOF), laminar, constant density fluids (air and water), static contact angle. All residuals were driven to 1E-4 or lower for every time step. The position and orientation were commanded, and isosurfaces at a volume fraction of 0.5 were recorded every 0.02 s. Time step was automatically adjusted based on CFL number. The CFL number was set at 1.5 because any higher usually resulted in instabilities.

C. Simulations and Post-Processing

Four simulations of four tests (see Table 4) were performed using OpenFOAM on the "america" compute cluster at KSC.

Paraview [14] was used to process the isosurfaces and create videos. Simple opacity, diffuse shading, and specular shading were used, so the CFD does not look particularly realistic. Note that the CFD images are just the clear tank walls and an isosurface at a volume fraction of 0.5. This may or may not accurately capture the true location of the fluid surface due to surface reconstruction smearing. Also, the bulk fluid is not shown or colored, so volumes that appear not to have any green fluid, but are bound by an isosurface, should in fact be green. This can be accomplished in Paraview using the Threshold filter, but it requires OpenFOAM to write a full output (as opposed to just an isosurface) at every time step. The final folder sizes for each test would have been over 300 GB, which was too large. Thus, only isosurfaces were written for every time step. Python batch scripts were written to create the proper camera perspectives (6DoF camera motion) and automate the rendering process.

IV. Results

In this section, a few experimental images and images extracted from the CFD videos will be compared. Due to the aforementioned lack of synchronization, the experimental videos and CFD videos had to be aligned in time manually. Every effort was made to ensure the best possible time alignment between experimental and CFD images presented here. Despite this, the time alignment error should be considered approximately +/-0.5 s. The below images are specified in time relative to the start of the CFD. Therefore, the A and B side views of the CFD are at the same time (within 0.02 s). However, due to the non-synchronized and low frame rate of the images captured by the cameras, the A and B side experiment images may *not* be at the exact same real time. The axes in the CFD images are the inertial (initial) axes.

Figures 2 and 3 are of Science 2 Test 11 at 14.2 s and 27 s respectively. In Figure 2, the bulk fluid distribution seems to agree fairly well. The bulge feature is present in all images. Even though it appears to be larger in the CFD, it is likely that the true maximum of the bulge was not captured by the cameras due to low frame rates. While the thicker fluid coating on the wall is captured in the CFD, the thin fluid film is not (see Section III.A). The cause for the drops along the wall in the CFD is unknown. They seem to appear and disappear at random in the videos. One hypothesis is that a thin fluid film is present, but is smeared by the surface scheme; a lower volume fraction isosurface may provide more insight on this phenomenon. In the experiment B-side of Figure 3, a large drop has broken off of a

prominence. In the video, this traverses the tank centerline to the other side of the tank. The next frame of the experiment A-side also shows the drop, but the frame shown in Figure 3 is of the prominence. The prominence forms in the CFD prior to this time, but instead of a drop breaking off, it collapses into the fluid along the side of the tank.

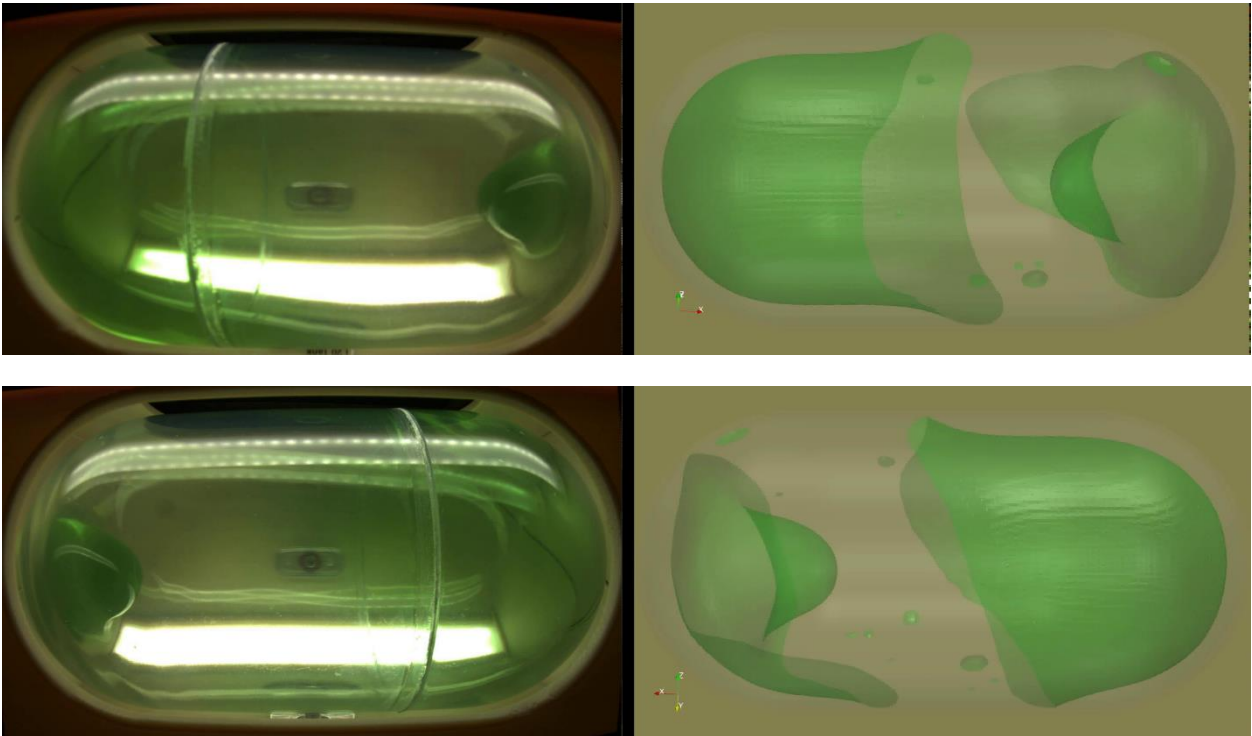


Figure 2. Science 2 Test 11 at ~14.2 s. Top: A-side. Bottom: B-side

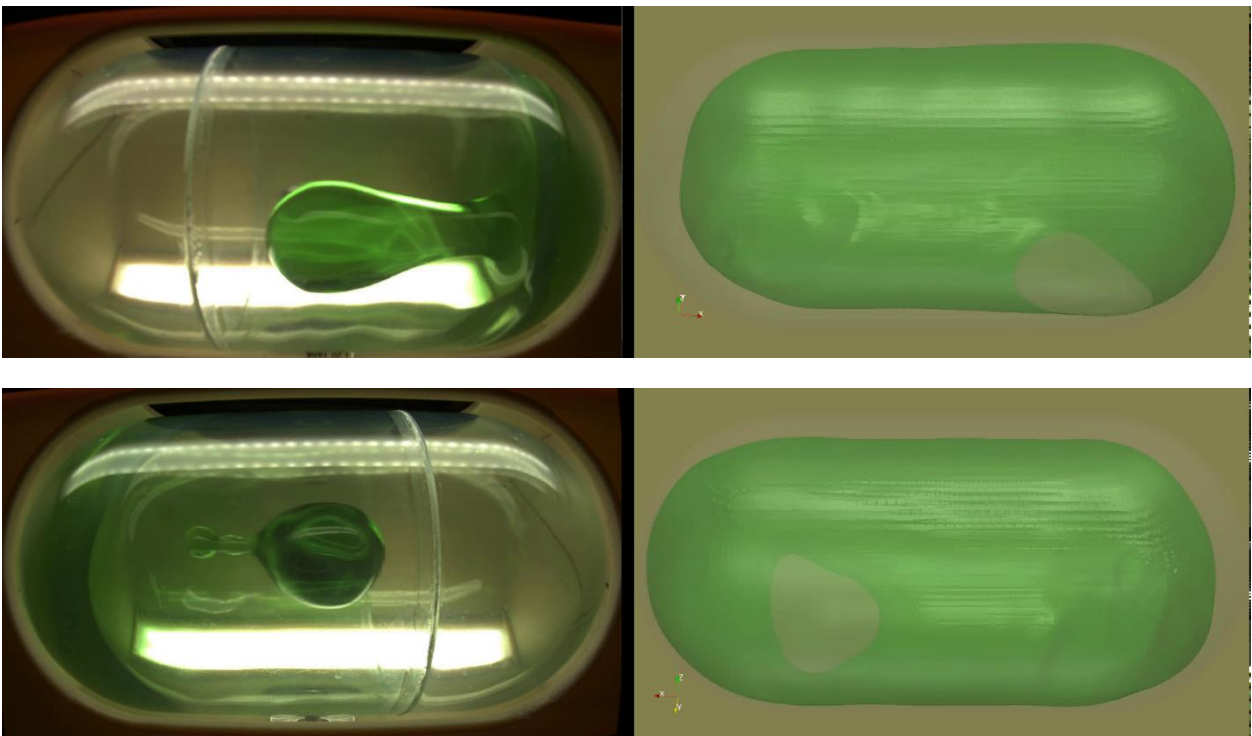


Figure 3. Science 2 Test 11 at ~27 s. Top: A-side. Bottom: B-side

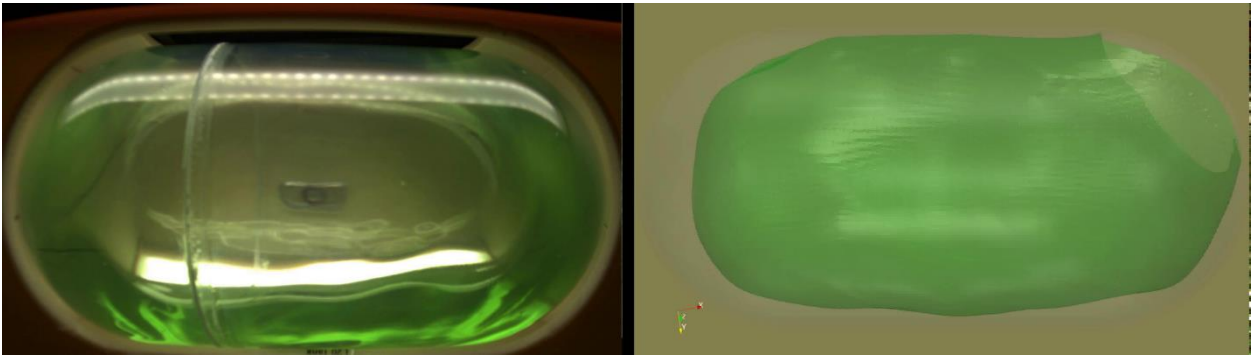


Figure 4. Science 3 Test 16 at ~14.6 s. Top: A-side. Bottom: B-side

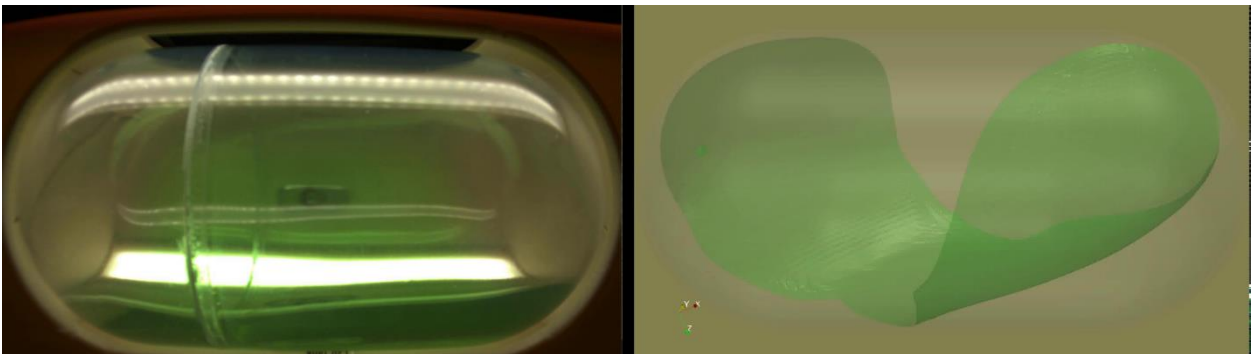
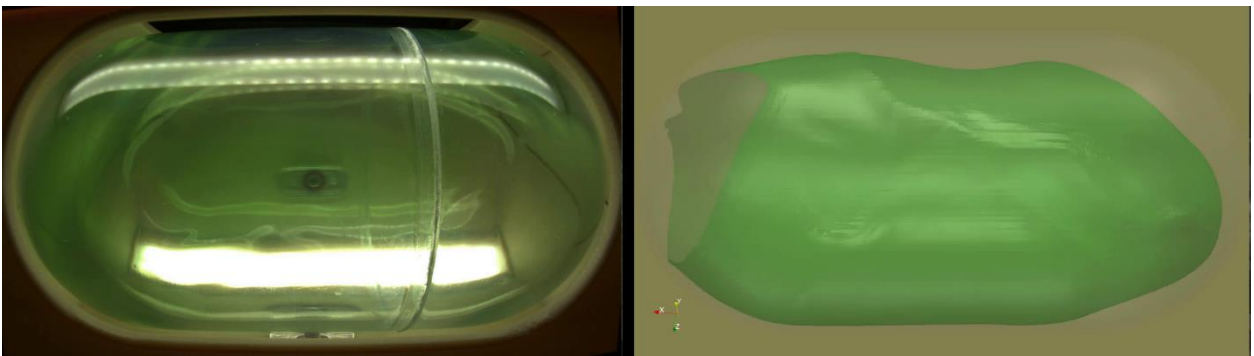


Figure 5. Science 3 Test 16 at ~37.3 s. Top: A-side. Bottom: B-side

Figures 4 and 5 are of Science 3 Test 16 at 14.6 s and 37.3 s respectively. In Figure 4 A-side, the bulk of the fluid is near the bottom of the images, and near the top of the images for the B-side. The experiment images also show some

fluid in the tank hemispheres, but the right side of the image for the A-side CFD, and the left side of the image for B-side CFD are mostly devoid of fluid. By the time of Figure 5, most of the fluid has settled in the $-z$ direction (CFD body frame). This is because this case was a spin about the x -axis and the center of mass of the experiment was slightly above ($+z$) the tank center. The CFD appears to be rotating about a slightly different axis than the experiment, though the bulk of the fluid is still on the $-z$ side of the tank. Another thing to note is the lack of a fluid film in the experiment images due to the centripetal acceleration from the spinning.

From these two cases, as well as videos of the other two cases, it was clear that there was something wrong with the rotation components of the 6DoF trajectory. After examining the order and signs of rotations required by OpenFOAM, it was determined that the data pipeline was providing an incorrect order of rotations. Because the Science 2 Test 11 case is primarily x -axis translational motion, it was repeated with 1DoF, x -axis translation only, motion. The trajectory was created by assuming that there were no rotations, and no translations in the other axes. When the x -axis acceleration from the 1DoF case is compared to that of the 6DoF case, they appear almost identical. Nothing else in the case was modified. Unlike the 6DoF case at $\sim 27s$, a drop *did* break off in the 1DoF case. This lends support to the theory that an error with the rotation order was causing the discrepancies between the CFD and the experiments. The data pipeline has been updated, and these four cases are being re-run with corrected trajectories.

V. Future Experiments

Various recommendations for future experiments similar to this one have been compiled. An absolute reference for trajectory corrections, e.g. the metrology system or optical tracking, is essential. The SPHERES-Slosh experiment collected data at about 20-30Hz and video at about 0.5-2 fps due to bandwidth limitations; these rates need to be higher (and constant) in order to eliminate data collection errors, implement better filters, and to resolve fast fluid flow features. In fact, reducing the resolution of the images collected by the cameras would significantly improve the framerate in the current setup. The IMU data files should be in a consistent format. All clocks need to be synchronized in future experiments to reduce errors introduced by time alignments.

VI. Conclusions and Future Work

The data pipeline is essentially complete. It will be cleaned up and uploaded to the KSC LSP Electronic Slosh Data Catalog (ESCD).

Various avenues for improving the CFD exist. OpenFOAM recently added a convenient tool for generating O-grid-type cylindrical meshes. That, plus mesh refinement, would probably improve solution accuracy. In terms of modelling, a fluid film and/or dynamic contact angle model would likely result in significant improvements, though including these requires writing custom code and recompiling OpenFOAM. The visualization could be improved by saving fluid bounding surfaces that include where the fluid contacts the wall, instead of simple isosurfaces. This functionality may already exist in OpenFOAM, but if not, it can be added. Using Blender instead of Paraview for video rendering would result in more realistic-looking images and videos. Controlling lighting, shadows, reflections, and refractions are all possible in Blender.

After some of these improvements have been made, and the experiment and CFD agree well for these four cases, many other cases from Science 2, 3, and 4 will be run and compared.

Acknowledgments

The SPHERES-Slosh experiment was funded through the NASA Game Changing Development Program. This research was partially funded by a NASA Space Technology Research Fellowship (NSTRF) through the Space Technology Mission Directorate (STMD).

References

- 1 Fraure, J., Vergalla, M., Zhou, R., Chintalapati, S., Gutierrez, H., and Kirk, D. (2010). Experimental Platform for the Study of Liquid Slosh Dynamics using Sounding Rockets. *IREASE*.
- 2 Gold, H., McArdle J., and Petrash, D. (1967). Slosh Dynamics Study in Near Zero Gravity. NASA TN D-3985. Cleveland, OH.
- 3 Vergalla, M., Zhou, R., Gutierrez, H., & Kirk, D. (2009). Experimental and Numerical Framework for Characterization of Slosh Dynamics. *IREASE*.
- 4 Lacovic, R., Yeb, F., Szabo, S., Brun, R., Stofan, A., and Berns, J. (1968). Management of Cryogenic Propellants in a Full-Scale Orbiting Space Vehicle. NASA TN D-4571. Cleveland, OH.
- 5 Buchanan, H. and Bugg, F. (1967). Orbital Investigation of Propellant Dynamics in a Large Rocket Booster. NASA TN D-3968. Huntsville, AL.

- 6 Petrash, D., Nussle, R., and Otto, E. (1963). Effect of the Acceleration Disturbances Encountered in the MA-7 Spacecraft on the Liquid-Vapor Interface in a Baffled Tank during Weightlessness. NASA TN D-1577. Cleveland, OH..
- 7 Crawley, E., Van Schoor, M., and Bokhour, E. (1993). The Middeck 0-Gravity Dynamics Experiment. NASA CR-4500.
- 8 Lapilli, G., Hollicker, C., Gutierrez, H., and Kirk, D. (2015). Design of a liquid sloshing experiment to operation in the International Space Station. *AIAA JPC*, Orlando, FL.
- 9 Hilstad, M., Enright, J., Richards, A., and Mohan, S. (2010). The SPHERES Guest Scientist Program. MIT. <http://ssl.mit.edu/spheres/gsp/>.
- 10 Mathworks, *MATLAB* 2014.
- 11 OpenFOAM V3.0.1. www.openfoam.org
- 12 Portland State University. SE-FIT V1.2 beta. www.se-fit.com
- 13 CD-Adapco. STAR-CCM+ V9.04. 2014.
- 14 Paraview V5.0.1. 2016. www.paraview.org