Simulation of a Non-Physical Barrier in the Brandon Road Lock and Dam Approach Channel

by Andrew Bell, David L. Smith, and Matthew Farthing

PURPOSE: This Engineer Research and Development Center (ERDC) Technical Note documents the implementation of a non-physical barrier (NPB) simulation into a computational fluid dynamics model of the Brandon Road Lock and Dam (BRLD) system using the open source software, OpenFOAM (Open source Field Operation and Manipulation). The NPB consists of a bubble curtain, a bank of lights, and an array of speakers placed at the bottom of a channel to slow or stop the movement of fish into unwanted areas. The pimpleFoamBLS solver used to run the simulation is described and the source files explained. The initial and boundary conditions for each variable used in the four different modules (hydrodynamic flow, bubble, lights, and sound) are presented. The results of the simulation for the three modules are also presented. It was concluded that the operation of the BRLD has a negligible effect on the operation of the non-physical barrier.

NOMENCLATURE

BC	bubble curtain
c	speed of sound
clight	speed of light
C_{dg}	coefficient of drag of bubbles
C_f	coefficient of friction of the bubbles
D	diffusion parameter
D_b	bubble diameter
D_{light}	light extinction coefficient
E	total irradiance
E_i	angular Irradiance
ECEL	irradiance of reflection
ε	rate of dissipation of turbulent kinetic energy
g	gravity
k	turbulent kinetic energy
N	bubble number density
NPB	non-physical barrier
p	pressure
p_c	pressure at injection point
R	bubble radius
R_c	bubble radius at injection point
Re_b	Reynold's number of bubble
S	source intensity
Sc	bubble Schmidt number
S_i	angular Source Intensity
t	time

Approved for public release; distribution is unlimited.

U	velocity
Ui	reflection direction
U_b	bubble velocity
U_g	gas velocity
Ui	direction vector for sources
U_r	gas-liquid relative velocity
$UCEL$	direction vector for reflection
w	acoustic energy
x	downstream direction
y	cross stream direction
z	vertical direction

Greek

α	gas volume fraction
ρ_k	kinematic density for buoyancy force
ρ_b	density of bubble
ρ_g	density of gas
ρ_l	density of water
σ	interfacial tension
ν_b	volume of the bubble
ν_t	kinematic turbulent viscosity

INTRODUCTION: The Brandon Road Lock and Dam (BRLD) system is located on the Des Plaines River near Joliet, IL and serves as a connection between the Mississippi River basin and the Great Lakes basin. Over the past few decades an invasive species of fish, the Asian carp, has become established in the Mississippi River watershed and now threatens the Great Lakes. Lawmakers and environmental agencies fear a migration of the fish from the Mississippi River basin to the Great Lakes through the Chicago metropolitan water system. This species has been classified as an ecological threat and could damage the recreational and industrial interests in the area. This has led to investigations into ways to block the movement of these fishes.

The BRLD represents one location where systems could be implemented to inhibit migration. One method to inhibit migration is the bio-acoustic fish fence (BAFF), a type of non-physical barrier (NPB) (Ellison 2015). The BAFF has been implemented in rivers and other water systems around the world to block fish movement into unwanted areas. The NPBs are placed on the bottom of a river or channel, perpendicular to the direction of flow, and contain three components; a bubble injection strip, a series of speakers, and a series of lights. When in operation, the bubble injection strip creates a steady, very dense curtain of bubbles. The speakers produce a high intensity noise at a set frequency which can deter fish. Noatch and Suski (2012) describe different types of NPBs and their effectiveness in the control of aquatic nuisances. It was found that an acoustic system alone was able to reduce the number of Asian carp passing through a channel or river by 95%. Strobe lights create a rapid, repeating change in light level in the channel and have been found to deter fish by triggering an avoidance response. This method has not been found to be wholly-effective without being part of an NPB system. The bubble curtain (BC) forms a screen across a river or channel that can seem like a solid structure to fish. However, from a distance or in areas of low light the effectiveness of this component may decrease. To combat the individual inefficiencies of the components, integrated NPBs have been created to combine the advantages of each tool and allow them to work together. It was found that this coupled system was 95% effective in confining Asian carp movements (Noatch and Suski 2012). The goal of this research is to implement a non-physical barrier consisting of bubbles, lights, and sound in a model for the BRLD.

IMPLEMENTATION WITH OpenFOAM: The open source software chosen for this research was OpenFOAM. The directory structure used in OpenFOAM consists of *`constant`*, *`system`*, and *`time`*. The *`constant`* directory contains directories that describe the model and physical properties of the case and folders that contain the mesh for the case. The *`system`* directory must contain at least three files; `controlDict,` where run control parameters are set, including start/end time, time step and parameters for data output, `fvSchemes`, where discretization schemes used in the solution may be selected at run-time, and `fvSolution`, where the equation solvers, tolerances and other algorithm controls are set for the run (User Guide 2015). Finally, the *`time`* directories house the initial and boundary conditions for the variables used in the solver.

The work presented was largely based on the work completed by Politano and Martin (2014) at the University of Iowa who created an OpenFOAM case for a section of the Sacramento River and implemented a non-physical barrier model in one of its channels. The flow solver used in this research is called pimpleFoamBLS and was built by Xiaofeng Liu and based on the pisoFoamBLS solver created by Politano and Martin (2014).

PRE-PROCESSING

Geometry generation. The first step of a numerical simulation is the creation of the geometry and subsequent mesh. Therefore, to create an accurate representation of the shape of the system, a satellite image of the river was chosen and a blue outline was drawn around the BRLD system (Ellison 2015) (Figure 1). The outline includes the necessary sections of the lock and dam system (light blue arrows signifying the direction of water flow). The lock and dam define the edges of the mesh and they will both serve as inlets for the flow. The initial design for NPB placement is shown in Figure 2. All of the BAFF components were implemented within this area including the BC which is placed at a certain point in the channel (Figure 2). The outline of the system was converted into a two dimensional shape. Then to generate depth, the shape outline was extruded nine feet to represent the depth of the channel creating a 3D shape.

The SnappyHexMesh meshing tool, included in OpenFOAM, requires stereo lithography (STL) files for each of the potential patches in the geometry. Therefore, the full 3D model was divided into sections with specific patch names (Figure 3). The bottom of the object was named “walls,” the dam was named “inlet,” the lock was named “lock,” the bubble curtain was named “BC,” the outlet was named “Outflow,” the sides of the channel were named “Sides,” and the surface of the water was named “Top.” To convert from Digital Access Exchange (DAE) to STL, the open source software MeshLab was used. MeshLab was also used to check if the parts have non-manifold edges or vertices. These non-manifold parts can lead to a mesh that does not contain watertight geometry, an essential element to meshing and simulation. The STL files were saved individually and placed in the triSurface folder within the *`constant`* directory.

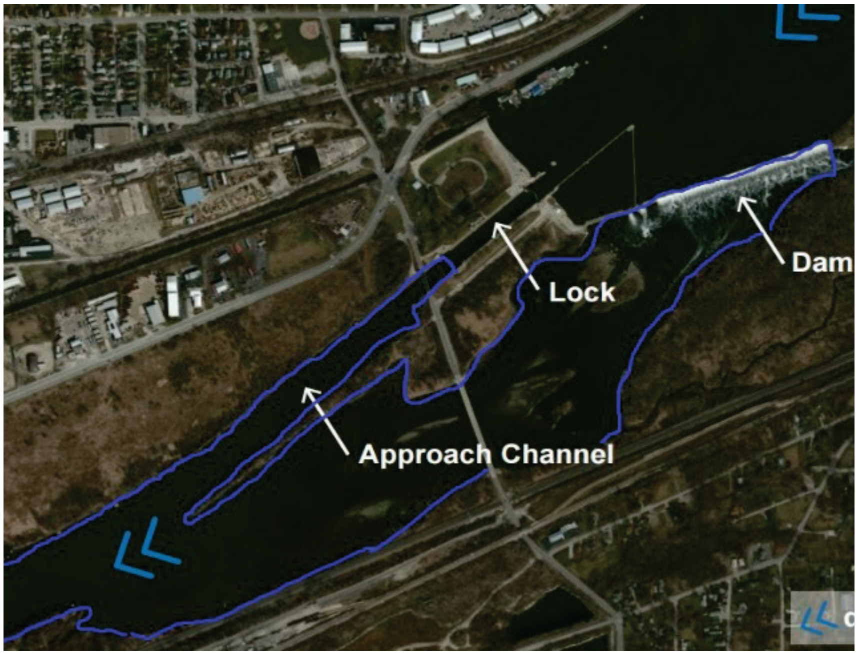

Figure 1. Outline of BRLD from satellite image.

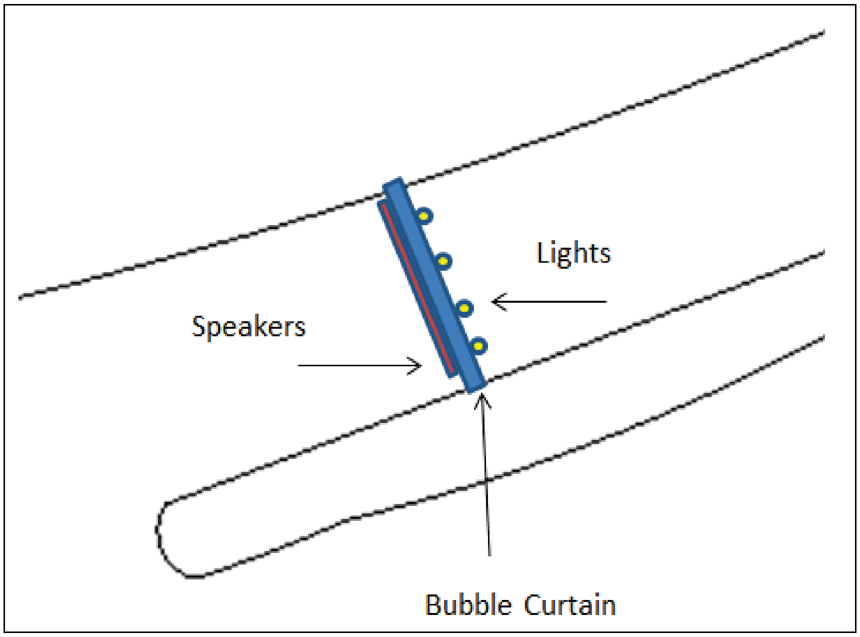

Figure 2. NPB initial placement.

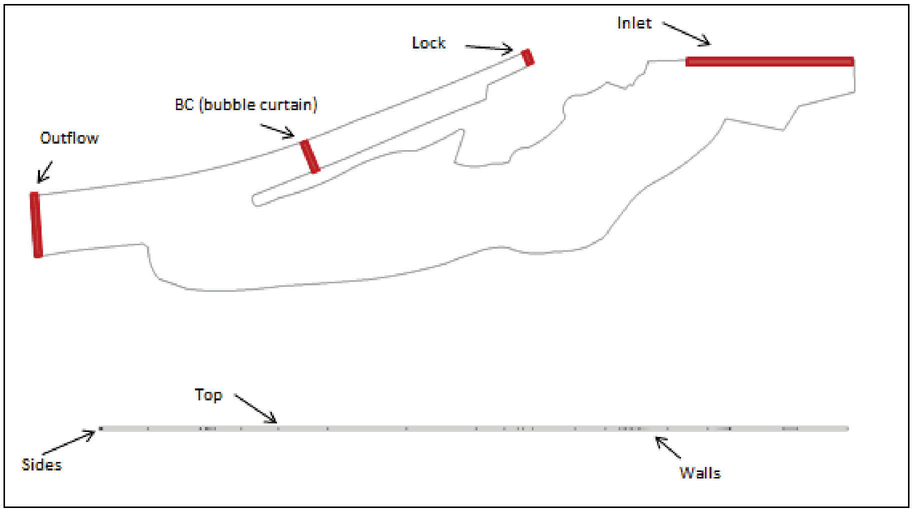

Figure 3. BRLD patch placement.

SnappyHexMesh. SnappyHexMesh is a tool used for mesh generation that sculpts a structured geometry from a pre-existing mesh created with blockMesh. The blockMesh command creates a large structured mesh around the geometry. An image of the blockMesh used for this mesh is shown in Figure 4. The SnappyHexMesh contains multiple components that increase the amount of refinement in the mesh, each having a multitude of quality control parameters. For this simulation the first two components were used, castellatedMesh and snap. The castellatedMesh control sculpts the block down to only the hexes that touch the surfaces of the geometry. Figure 5a shows the entire domain. Figure 5b shows the results of the first meshing. Finally, the snap feature moves the cell vertex points to the surface of the geometry and removes the excess from the castellated mesh. The result of this meshing, which for this simulation serves as the final mesh, is shown in Figure 5c. In snappyHexMesh, every patch can be set to mesh to a different level of refinement. This ability can be seen in Figures 5b and 5c. A patch that requires a finer mesh than the surrounding areas (OpenFOAM User Guide 2015) is shown in Figures 5a and 5c.

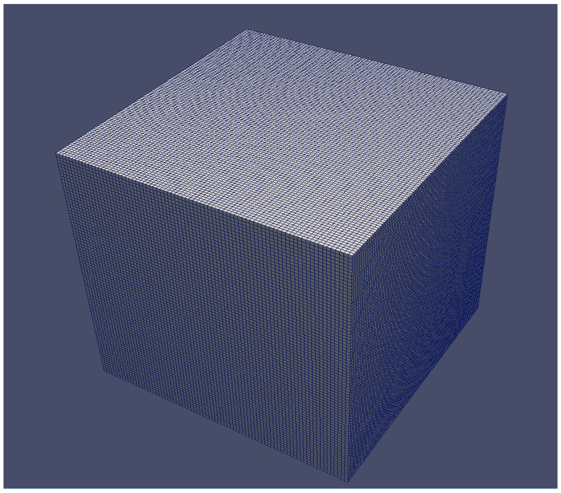

Figure 4. Block mesh.

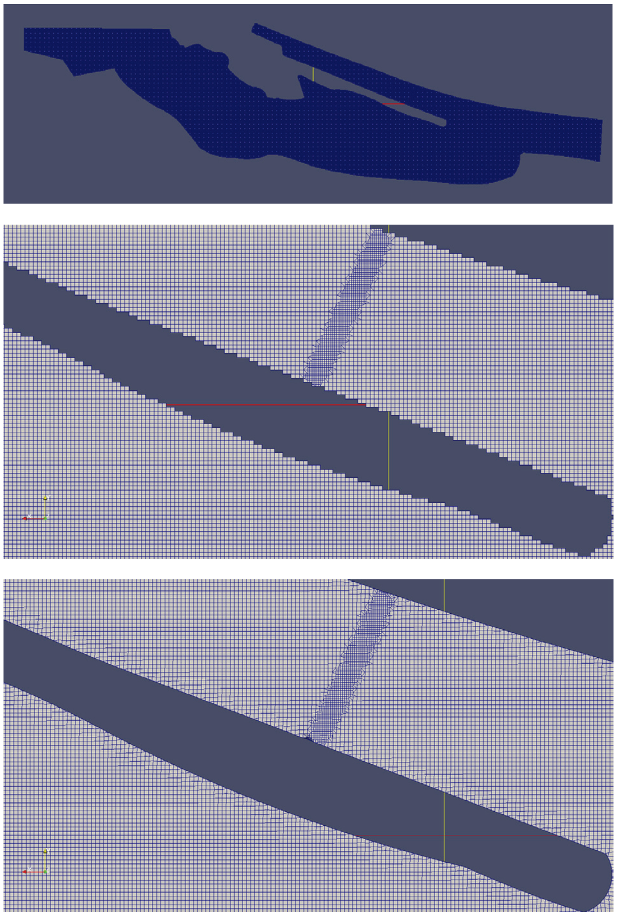

Figure 5. Mesh refinement study of snappyHexMesh. a) Entire domain with section marked for examination. b) Castellated mesh refinement. c) Snap feature refinement.

PimpleFoamBLS solver. The pimpleFoamBLS solver is a collection of modules that work together to simulate the flow of water through the channel and the propagation of bubbles, light and sound through the fluid. A graphical breakdown of the pimpleFoamBLS solver is shown in Figure 6.

The pimpleFoamBLS code contains a predictor loop named BLSproperties where the model properties are created and updated. The createfields module is used to read the variable files stored inside the time directories. These files declare the BLS variables in light, sound, and the bubble concentration *(α)* equations. The BLSproperties module creates physical and model parameters that are used in the light module, sound module, bubble curtain module, and pimpleFoam solver. The parameters listed in BLSproperties are presented in Table 1. The sound module pulls from the sound attenuation equation code and the speed of sound equation. The light module pulls values from the turbulent kinetic energy equation code which describes the attenuation of light (Politano and Martin 2014). Additionally, the `create fields` directory is a crucial part of the pimpleFoamBLS solver because it defines the values in all of the scalar and vector fields.

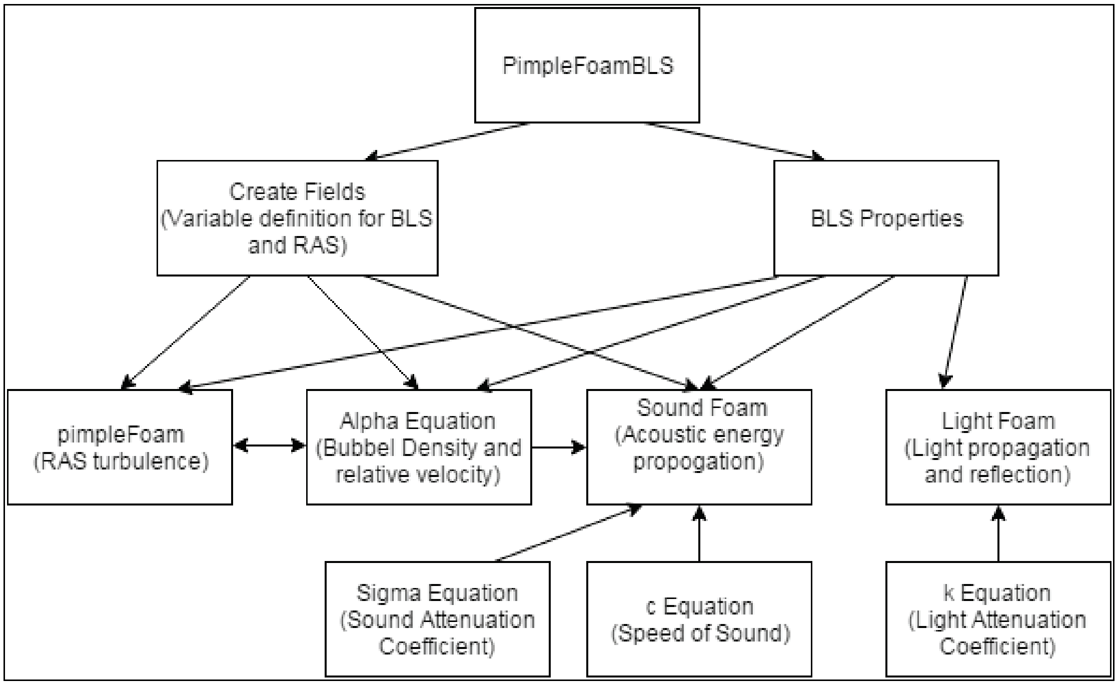
PimpleFoamBLS
Create Fields (Variable definition for BLS and RAS)
BLS Properties
pimpleFoam (RAS turbulence)
Alpha Equation (Bubbel Density and relative velocity)
Sound Foam (Acoustic energy propogation)
Light Foam (Light propagation and reflection)
Sigma Equation (Sound Attenuation Coefficient)
c Equation (Speed of Sound)
k Equation (Light Attenuation Coefficient)

Figure 6. Breakdown diagram of the pimpleFoamBLS solver. Reynolds-averaged stress (RAS) is used to determine turbulence within the pimpleFoamBLS solver.

Table 1. Physical and model parameters from BLSproperties.

Property	Value
Kinematic Viscosity	1e-6 m^2/s
Water density	1000 kg/m^3
Gas density	1.02 kg/m^3
Pressure at bubble curtain	1.91027e5 Pa
Bubble diameter at injection	8e-4 m
Temperature	298 K
Free surface elevation	2.68 m
Sound Diffusion parameter	150 m/s
Length scale of diffusion tensor in x-dir	200 m
Length scale of diffusion tensor in y-dir	200 m
Length scale of diffusion tensor in z-dir	10 m
Sound speed in water	1500 m/s
Sound speed in air	1 m/s
Bubble resonant frequency	400 1/s
Sound reference value	4.44e-22 Ws/m
Water absorption coefficient	0.32 1/m

PimpleFoam solver. The pimpleFoam solver is a large time-step, transient solver for incompressible flow and uses a combination of the pisoFoam and simpleFoam solvers. In this case, it is using a Reynolds Averaged Navier Stokes equation (RANS) scheme to calculate the values for the flow of water through the channel. The parameters for the pimpleFoam solver are set in the `fvSolution` file in the system directory. The pimpleFoam code is located in OpenFOAM's source code. First the velocity is calculated using Equation 1 where U is the velocity, p is the pressure, and R is the stress tensor.

$$\frac{\partial U}{\partial t}+\nabla\cdot\left(UU\right)+\nabla\cdot R=-\nabla p \tag{1}$$

Next, the pressure is calculated. For this simulation the nCorrector was set to 2, so the pressure field is corrected twice by calculating p in Equation 2 twice.

$$\nabla^{2}p=f\left(U,\nabla p\right) \tag{2}$$

Next, because the nOuterCorrectors value is set to two, the entire loop is repeated to find the corrected value for U and p. This loop solves for the pressure-velocity coupling (OpenFOAM guide/The PIMPLE algorithm in OpenFOAM 2016).

Bubble module. The BC is described as a “discrete gas phase in a continuous fluid” (Politano and Martin 2014). The bubbles are modeled by the introduction to the channel of water as a flow of lower density fluid. The bubbles accelerate as they exit the injection point and their diameter grows proportionally with the distance from the injection point until they reach an equilibrium condition. There are multiple ways to model a BC in a flow but the method chosen for this simulation was a two-fluid approach of an incompressible gas-liquid phase mixture. The gas volume fraction equation code uses parameters from BLS properties to model the BC with a series of variables. The first variable is the bubble number density (N) calculated using Equation 3, where t is time, Sc is bubble Schmidt number, v_t is the kinematic turbulent viscosity, and U_g is the gas velocity vector.

$$\frac{\partial N}{\partial t}+\nabla\cdot\left[u_{g}N\right]=\frac{v^{t}}{Sc}\nabla^{2}N_{g} \tag{3}$$

Next, the bubble volume is calculated for a given position in the mesh using Equation 4, where v_b is the bubble volume, P_c is the pressure at the injection point, and R_c is the radius of the bubble at the injection point.

$$v_{b}=\frac{P_{c}}{P}\cdot\frac{4}{3}\cdot\pi\cdot R_{c}^{3} \tag{4}$$

The gas volume fraction *(α)* is a non-dimensional ratio of the densities of the gas in the bubble and the surrounding fluid. To calculate the gas volume fraction, the bubble number density is multiplied by the volume of the bubble at a certain point. This is shown in Equation 5.

$$\alpha=N\cdot v_{b} \tag{5}$$

To calculate the bubble-liquid relative velocity (U_r) in the z direction (the direction perpendicular to the bottom of the channel) two equations must be used. First, to find the velocity while the bubbles are very small, with a D_b less than $4\cdot 10_4$, Equation 6 is used. In this equation D_b is the diameter of the bubble and g is the acceleration due to gravity.

$$U_r = \frac{-\left(\rho_l - \rho_g\right)\cdot D_b^{\frac{1}{2}}}{\frac{18\cdot v^l}{\rho_l \cdot g}} \tag{6}$$

As bubbles rise and leave the injection point they grow in diameter. Here, the coefficient of drag for the bubbles (C_{dg}) is needed to find the relative velocity. The equation for the drag coefficient is presented in Equation 7.

$$C_{dg} = 24\cdot\frac{\left[1+0.15\cdot\left(Re_b^{0.687}\right)\right]}{\frac{18\cdot v^l}{\rho_l \cdot g}} \tag{7}$$

The values for the drag coefficient and an initial value for relative velocity can be used to find the friction coefficient, C_f, using Equation 8.

$$C_f = \frac{3\cdot\rho_l.C_{dg}\ \cdot U_r}{4\cdot D_b} \tag{8}$$

The speed and direction of the bubbles, (i.e., the bubble velocity variable) (U_b), is calculated using Equation 9.

$$U_b = U_r + U\left(1-\alpha\right) \tag{9}$$

RUNNING THE SIMULATION

Initial and boundary conditions. The initial conditions and boundary conditions for all of the variables used in the modules were located in the time directory files. Because the simulation starts with the 0 time directory, the initial conditions are located there.

Hydrodynamic fields. The hydrodynamic fields define the flow of water through the channel, with a turbulence model turned on. The chosen turbulence model for this simulation is *k*-epsilon, so there are fields for *k* and epsilon values. Table 2 lists the variable, its symbol, and the initial condition defined for it in the 0 folder. The *p*, *k*, and ε fields are scalar fields while *U* is a vector field.

While initial conditions specify the initial value for the field, boundary conditions define what each variable is doing at each patch. Table 3 defines the boundary condition for each patch. The slip condition treats the top surface like it is free to move and is not locked in place. This is an easy way to model the free surface of the water without simulating the multi-phase flow of air and water.

Table 2. Hydrodynamic initial conditions.		
Variable	**Symbol**	**Initial Condition**
Pressure	p	0
Velocity	U	(0 0 0)
Turbulent kinetic energy	k	$1x10^{-6}$
Rate of dissipation of k	ε	$1x10^{-4}$

Table 3. Hydrodynamic boundary conditions.							
	Inlet	**Lock**	**Outflow**	**BC**	**Walls**	**Top**	**Sides**
U	Fixed value (0 -0.2 0)	Fixed value (-0.2 0 0)	pressureInletOutletVelocity (0 0 0)	Fixed value (0 0 0)	Fixed value (0 0 0)	Slip	Fixed Value (0 0 0)
p	calculated	calculated	calculated	calculated	calculated	calculated	calculated
k	Fixed value 6e-4	Fixed value 6e-4	Zero Gradient	Zero Gradient	kqRWall function 6e-4	slip	kqRWall Function 6e-4
ε	Fixed Value 1.3e-6	Fixed Value 6e-4	EpsilonWallfunction		epsilonWall function	Slip	Epsilon Wall function

Bubble fields. The initial and boundary conditions define the variables that make up the BC that resides at the bottom of the channel. The gas volume fraction is introduced at 0 for the initial condition and as the boundary condition for all of the patches except for the BC. At the BC, the gas volume fraction boundary condition is 0.2 which means that the density of the bubbles is 20% of the density of the surrounding fluid. The initial velocity of the bubbles in the *z* direction is 0.09 m/s. The gas liquid relative velocity changes as the bubble moves but at the injection where the flow of the river is 0 m/s the relative velocity is the same as the bubble velocity in the *z* direction, 0.09 m/s. The bubble number density and the number of bubbles per cubic meter has an initial condition of 0 but a boundary condition of $7.46 \cdot 10^{8}$ for the BC patch. Also, the bubble diameter is very small at injection, with a value of $8.00 \cdot 10^{-4}$ m. The initial conditions for the internal field for the bubble curtain are located in Table 4 and the boundary conditions are located in Table 5.

Table 4. Bubble curtain initial conditions.		
Variable	**Symbol**	**Initial Condition**
Gas Volume Fraction	α	0
Bubble Velocity	U_b	(0 -0.2 0.09)
Gas liquid relative Velocity	U_r	(0 0 0.09)
Bubble number density	N	0
Diameter of bubble	D_b	0.0008
Density of bubble	ρ_b	2.1818

Table 5. Bubble boundary conditions.

	Inlet	Lock	Outflow	BC	Walls	Top	Sides
α	zeroGradient	zeroGradient	zeroGradient	0.2	zeroGradient	zeroGradient	zeroGradient
D_b	zeroGradient	zeroGradient	zeroGradient	zeroGradient	zeroGradient	zeroGradient	zeroGradient
N	zeroGradient	zeroGradient	zeroGradient	7.46e08	zeroGradient	zeroGradient	zeroGradient
P_b	Calculated 2.23	Calculated 2.23	Calculated 2.23	Calculated 1.23	Calculated 2.23	Calculated 2.23	Calculated 2.23
U_b	Calculated (0 -0.2 0.09)	Calculated (-0.2 0 0.09)	Calculated (0 0 0)	Calculated (0 0 0.09)	Calculated (0 0 0)	Calculated (0 0 0)	Calculated (0 0 0)
U_r	Calculated (0 0 0.09)	Calculated (0 0 0.09)	Calculated (0 0 0.09)	Calculated (0 0 0.09)	Calculated (0 0 0.09)	Calculated (0 0 0.09)	Calculated (0 0 0.09)

Sound fields. The sound module models the sound energy, attenuation, and dissipation of sound created from 20 speakers on the bottom of the channel located downstream of the bubble curtain. The source variable defines where these speakers are and the locations are set using coordinates in the setFields Dictionary. The sound energy level is initially set to zero in both the initial and boundary conditions. The interfacial tension is the surface tension of the water on different areas with the introduction of bubbles. The initial and boundary conditions for this are also both set to zero. Finally, the diffusion parameter is set as a tensor with components in *x*, *y*, and *z*-directions for both the internal initial field and the boundary conditions for each patch. The initial conditions are presented in Table 6 and the boundary conditions are shown in Table 7.

Table 6. Sound module initial conditions.

Variable File	Symbol	Value
Sound Energy Level	w	0
Diffusion parameter	D	(30000 0 0 0 30000 0 0 0 1500)
Source	Source	setFields
Interfacial tension	σ	0

Table 7. Sound fields boundary conditions.

	Inlet	Lock	Outflow	BC	Walls	Top	Sides
w	Fixed Value 0	Fixed Value 0	Fixed Value 0	GroovyBC	GroovyBC	Fixed Value 0	Fixed Value 0
D	Calculated (30000 0 0 0 30000 0 0 01500)	Calculated (30000 0 0 0 30000 0 0 0 1500)	Calculated (30000 0 0 0 30000 0 0 0 1500)	Calculated (30000 0 0 0 30000 0 0 0 1500)	Calculated (30000 0 0 0 30000 0 0 0 1500)	Calculated (30000 0 0 0 30000 0 0 0 1500)	Calculated (30000 0 0 0 30000 0 0 0 1500)
Source	Zero Gradient	Zero Gradient	Zero Gradient	Zero Gradient	Zero Gradient	Zero Gradient	Zero Gradient
σ	Calculated 0	Calculated 0	Calculated 0	Calculated 0	Calculated 0	Calculated 0	Calculated 0

Light fields. There are a series of four lights located adjacent to the bubble curtain. These lights have seven reflection directions each. The irradiance of light is measured in units of W/m^2. The four light sources are set in the setFields dictionary. The irradiance (E_i) and the source intensity (S_i) files are scalar fields while the direction of the source is a vector field. The ECEL and UCEL files are irradiance and direction fields for the reflection direction for each patch. The extinction coefficient (D_{light}) is used to model the dissipation of light over distance through a medium. The initial and boundary conditions of the U, E, and S files are zero. The initial conditions for the UCEL files are set with the unit vector for the given reflection direction (Politano and Martin 2014).

Initial case. Running the simulation requires a “hot start” approach. This method requires two cases where the first is setup to run without the NPB active to prime the system. These cases are called the initial and real run cases. The cases must be set up with the *`time`* directories and the *`constant`*, and *`system`* folders. In the initial case, the switches for simulating bubbles, lights, and sound were turned off. These switches are located in the BLS properties folder. When the pimpleFoamBLS command is used to start the simulation only the pressure, velocity, and the k-ε turbulence model are calculated for the system. In Figure 7, the velocity on the free surface of the system is shown for the initial case setup after 5 seconds of simulation. The initial case only requires a physical run time between 1 and 5 seconds.

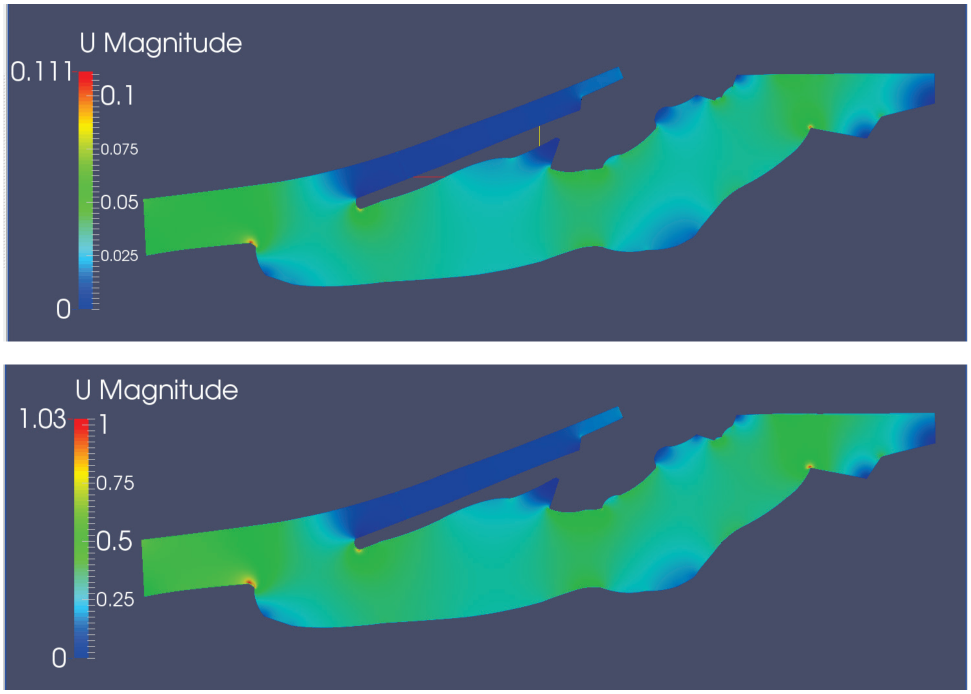

Figure 7. Velocity field for initial setup for a) 0.02m/s and b) 0.2m/s cases on surface.

Real run case. The second part of the hot start approach is the real run. After running the initial case for a short amount of time, the last time folder from the initial case must be copied to the real run case. The `controlDict` must then be edited to include the start time as the final time of the initial run. In the real run, the switches for bubbles, lights, and sound must be turned on. The hot start primes the channel with flow so that the NPB components have fluid in which to operate.

To analyze the effects of the operation of the lock and dam, two separate BRLD cases were set up. One case simulated flow through the system with the lock open. For this condition, the velocity through the inlet and lock was set as 0.2 m/s. The other case simulated the flow with the lock closed. This setup consists of nearly still water with inflow velocities of just 0.02 m/s. The depth of the channel remains constant for both setups.

RESULTS: The velocity through the two cases for the initial setup is shown in Figure 7. The velocity in the second case is an order of magnitude greater, due to the initial velocity through the inlet. Figure 8 shows what the BC looks like by mapping the gas volume fraction and observing the channel at four different depths from the surface: 2.6m, a depth slightly above the injection point; 1.9m; 0.9m; and 0.0m, a point at the free surface of the channel. These cuts through the water column in the channel show that the gas volume fraction is very large at the bottom of the channel and that the area of the BC is the same size as the BC patch. As the bubbles move up through the channel, the gas volume fraction shrinks as the bubble diameter grows. Also, the bubbles begin to spread out, causing the BC to be wider at the top. Figure 9 shows the velocity through the system with the NPB turned on. Part a) shows the velocity magnitude with the lock closed and the Part b) shows the magnitude with the lock open. These graphs show that for the slower case, the velocity of water created around the BC is significantly higher than anywhere else in the system. To analyze the effect of the lock operation on the bubble curtain, the flow velocity and bubble velocity are plotted against the distance from the bottom of the channel in Figure 10 for both velocity cases. These graphs are nearly identical, showing for the slow case and the fast case the effect of the operation the NPB is miniscule. Additionally, a contour of the sound energy is shown in Figure 11. From the figure, the sound energy level of the speakers is not strong enough to pass through the bubble curtain. The image of the contour of sound energy shows that the sound energy propagates from the speakers and stops at the BC. Fortunately however, the area that sound is required for effectiveness is on the downstream side of the bubble curtain. The change in velocity in the channel for the 0.2m/s and the 0.02m/s cases had no measurable effect on the sound energy level in the channel. Figure 12 shows a contour of the light energy in the channel.

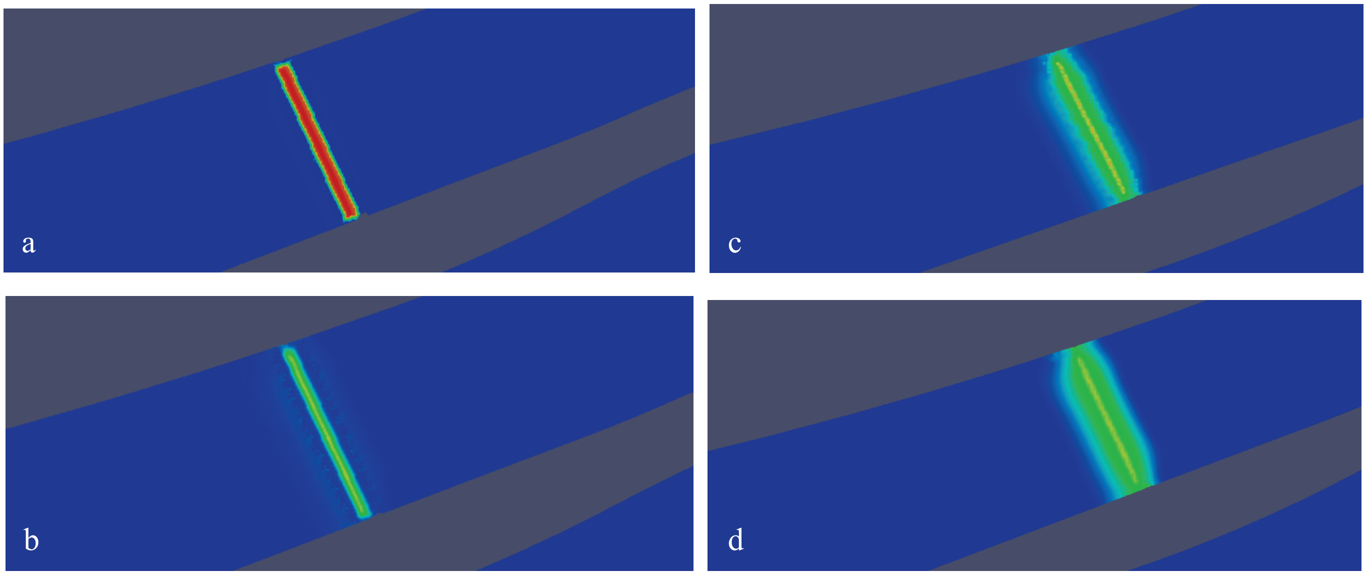

Figure 8. Gas volume fraction field at depth of a) 2.6m, b) 1.9m, c) 0.9m, and d) 0m.

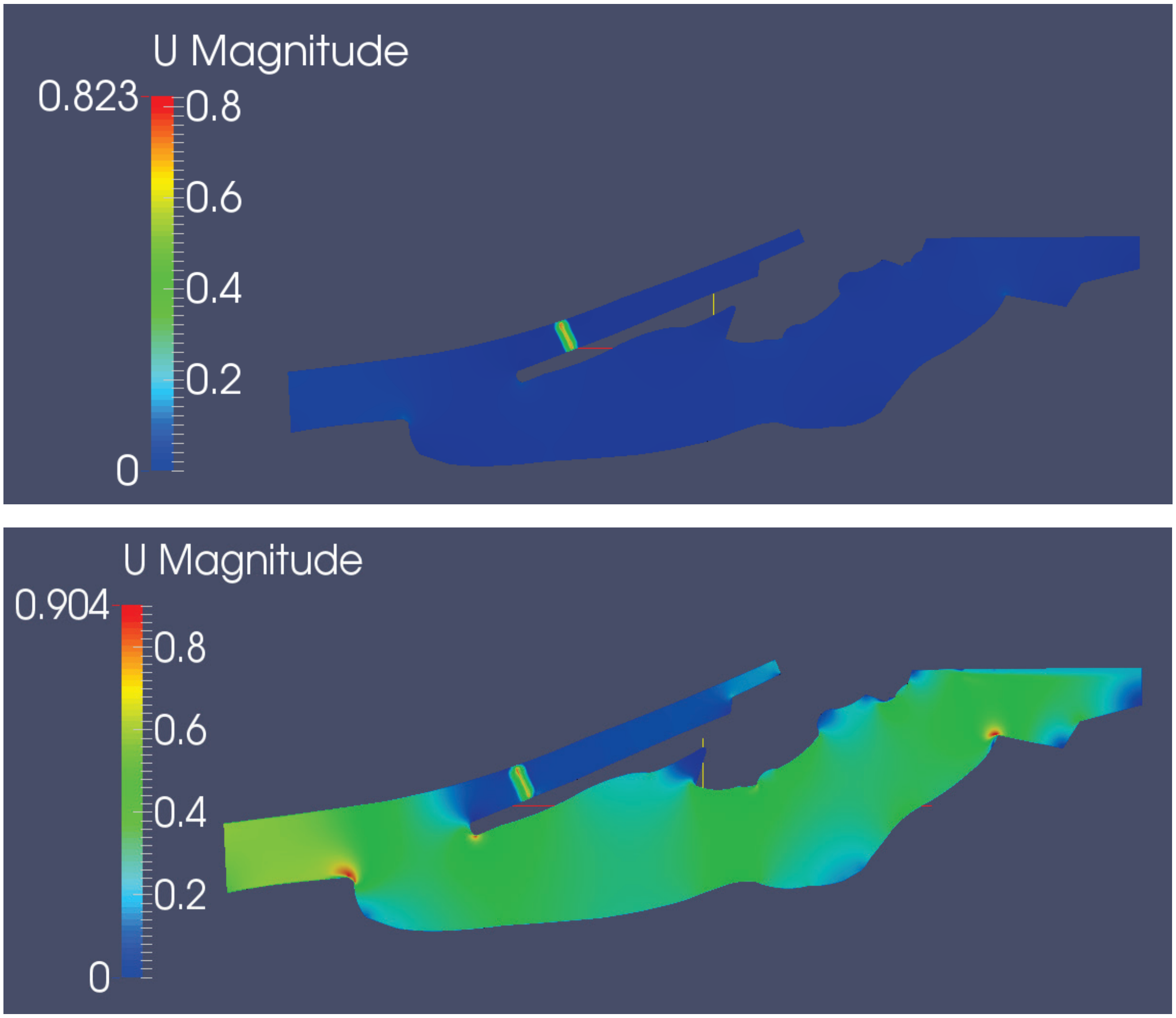

Figure 9. Velocityfield for real run case with a) 0.02 m/s and b) 0.2 m/s on surface.

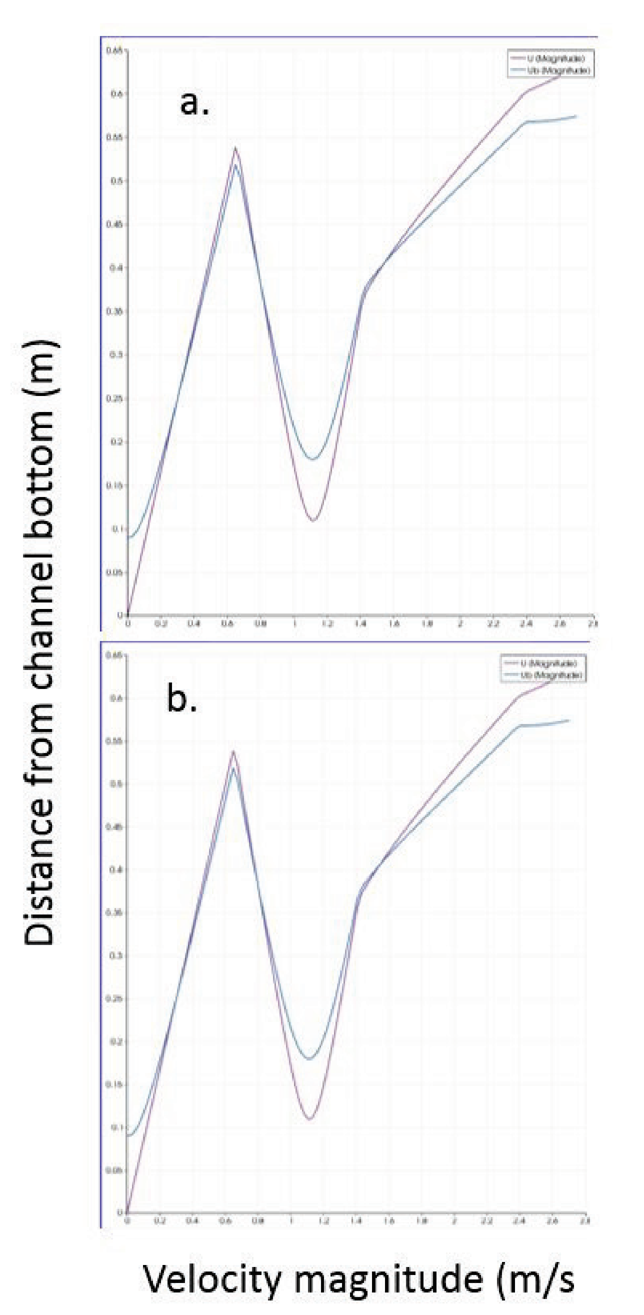

Figure 10. Flow velocity and bubble velocity across bubble curtain for a) 0.02m/s case and b) 0.2m/s case

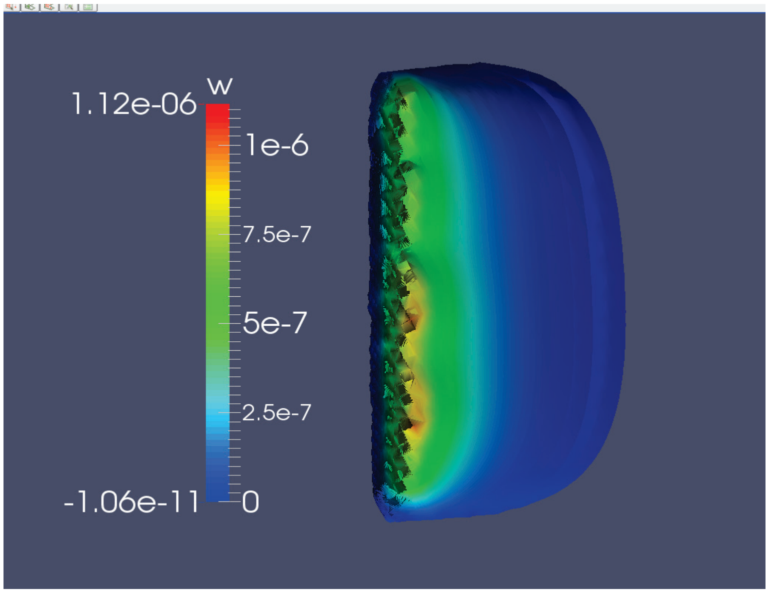

Figure 11. Sound energy in real run cases.

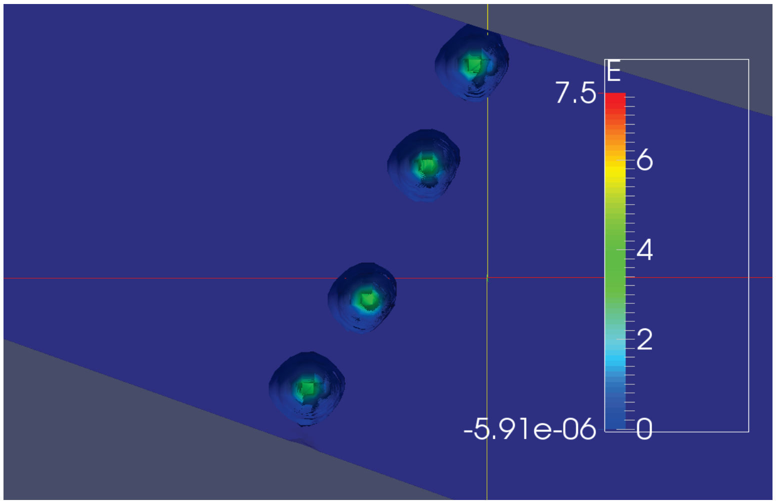

Figure 12. Light energy field in real run cases.

CONTINUING WORK AND CONCLUSIONS: There are multiple avenues that can be followed to continue this work. The first effort is in mesh generation. Two components in the SnappyHexMesh tool were used to create the mesh. A third, unused module could be added in further tests. This tool adds a layer of user-controlled mesh cells to given patches. This method can possibly improve the simulation's accuracy, but will surely increase the cell count and in turn increase the required run time. Another effort continuing this work would be to change the number of light sources, reflections, and speakers. Because the BC patch is the determining factor for the area of the bubble curtain in the NPB, if this patch were to change in size the results of the simulation would be altered. Determining if the BC can still be efficient if it is thinner than currently modeled would be an interesting study. Finally, more work can be completed to ensure that the boundary and initial conditions used in the numerical model correlate to the existing physical system. These inaccuracies may have contributed to the large velocities found at the sharp corners in the simulation. These points can be seen in Figure 7 and are indicated with red.

After completing the initial and real run cases for two velocities for simulation of a NPB in the approach channel of the BRLD system, several conclusions can be made. First, the contour of the light energy shows that the effect of the lights does not cover the entire width of the channel. Therefore, some sections of the BC are not lighted, which will likely decrease the barrier's efficiency. The number of light sources used in the barrier must be increased to cover this area. Next, the sound contour shows that the sound energy emitted by the speakers does cover a wide enough area in the channel; therefore, the number of speakers and their placement in the area is sufficient. The state of the channel with the lock open was simulated by increasing the velocity of one case by an order of magnitude over that of the case simulating a closed lock. The velocity contours of these two cases show that the effect of the operation of the lock is negligible on the NPB. Finally, the goal of completing all stages of the simulation; pre-processing, running, and post-processing with only open source software programs was achieved.

ACKNOWLEDGEMENTS: The authors would like to thank the valuable input and peer review of Dr. Xiaofeng Liu of Pennsylvania State University and Dr. Marcela Politano of the University of Iowa.

POINTS OF CONTACT: For additional information, contact Dr. David L. Smith (601-634-4267), david.l.smith@usace.army.mill, or the manager of the Aquatic Nuisance Species Research Program (ANSRP), Dr. Linda Nelson, (601-634-2656), Linda.S.Nelson@usace.army.mil. This technical note should be cited as follows:

> Bell, A., D. L. Smith, and M. Farthing. 2017. *Simulation of a non-physical barrier in the Brandon Road Lock and Dam approach channel*. ANSRP Technical Notes Collection. ERDC/TN ANSRP-17-1. Vicksburg, MS: U.S. Army Engineer Research and Development Center. http://el.erdc.usace.army.mil/ansrp/ansrp.html.

REFERENCES

Ellison, G. 2015. Blocking Asian carp at Illinois 'choke point' dam focus of new bipartisan bill. http://www.mlive.com/news/index.ssf/2015/02/defending_our_great_lakes_act.html. (accessed on 22 September 2015).

Noatch, M. R., and C. D. Suski. 2012. Non-physical barriers to deter fish movements. http://fishlab.nres.illinois.edu/Reprints/Noatch_Suski_2012.pdf. (accessed on 30 October 2015).

OpenFOAM guide/The PIMPLE algorithm in OpenFOAM. 2016. OpenFOAM guide/The PIMPLE algorithm in OpenFOAM. https://openfoamwiki.net/index.php/OpenFOAM_guide/The_PIMPLE_algorithm_in_OpenFOAM (accessed on 30 October 2015).

OpenFOAM User Guide: 4.1 File structure of OpenFOAM cases. 2015. http://cfd.direct/openfoam/user-guide/case-file-structure/. (accessed on 20 March 2015).

OpenFOAM User Guide: 5.4 Mesh generation, snappyHexMesh. 2015. http://cfd.direct/openfoam/user-guide/snappyhexmesh/. (accessed on 30 October 2015).

Politano, M., and J. E. Martin. 2014. Integrated bubble, sound, and light modeling for a non-physical Fish Barrier. *IIHR - Hydroscience and Engineering*. (accessed on 22 September 2015).

Politano, M., and J. E. Martin. 2014. OpenFoamBLS Tutorial. *IIHR - Hydroscience and Engineering*. (accessed on 22 September 2015).

***NOTE:** The contents of this technical note are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such products.*