How the three-dimensional geometry of computational domain(s) affects the accuracy of non-reflective boundary conditions in acoustic simulation

Janelle Resch
jresch@uwaterloo.ca

Summary
Describing and simulating acoustic wave propagation can be difficult and time consuming; especially when modeling three-dimensional (3D) problems. As the propagating waves exit the computational domain, the amplitude needs to be sufficiently small otherwise reflections can occur from the boundary influencing the numerical solution. This paper will attempt to quantify what is meant by ‘sufficiently small’ and investigate whether the geometry of the computational boundary can be manipulated to reduce reflections at the outer walls. The 3D compressible Euler equations were solved using the discontinuous Galerkin method on a graphical processing unit. A pressure pulse with an amplitude equivalent to 10% of atmospheric pressure was simulated through a modified trumpet within seven different geometries. The numerical results indicate that if the amplitude of the pulse is less than 0.5% of atmospheric pressure, reflections are minimal and do not significantly influence the solution in the domain. Furthermore, the computational region behind the bell can be neglected greatly reducing the required memory and run time.

Keywords: Computational domain; computational boundary geometry; acoustic wave propagation; non-reflective boundary conditions; pass-through boundary conditions; DGM; trumpet flare; spherical symmetry; GPU.

1 Introduction and Background
Systems of partial differential equations (PDEs) which model acoustic wave propagation can typically be written as a hyperbolic system of conservative laws. Numerically solving these equations of motion can be expensive because discontinuities can arise in the solution, even if the initial wave profile is smooth [1]. A common approach when numerically simulating small-amplitude wave propagation (i.e., acoustic waves) is to use a finite volume (FV) method; whereas for finite-amplitude waves (i.e., nonlinear acoustic waves), finite element (FE) methods are more standard [2]. A high-order accurate method which combines aspects from both FV and FE methods is the discontinuous Galerkin method (DGM) which was developed by Cockburn and Shu [3] [4]. This method is particularly useful for modeling hyperbolic systems because numerical fluxes defined along the cell interface guarantees stability and local solvability, i.e., at each time step, only information about the neighbouring cells are required. DGM can also accommodate unstructured meshes, adaptive refinement, [4] and is open to parallel implementations [5].

To ensure acoustic simulations can be completed in a reasonable amount of time, the computational domain needs to be truncated in an efficient manner. A computational boundary, denoted by $\partial \Omega$, is defined to construct a computational domain, denoted by $\Omega$. In order to have a well-posed problem, boundary conditions (BCs) must be defined on $\partial \Omega$ [6]. In acoustoacoustic simulations, this is one of the most difficult endeavors [8]. The outflow BCs must be imposed such that the acoustic waves can propagate directly out of the domain with minimal reflections. Such BCs are typically referred to as pass-through or non-reflecting [7].

If the BCs defined on $\partial \Omega$ are poorly prescribed, spurious waves may be generated and reflected from the boundary [7]. This can specifically be an issue for nonlinear problems or higher-order schemes due to the low dispersion and dissipation properties of the DGM [8]. Spurious waves are typically classified as “unphysical and supposedly make elements of high-order useless for accurate computations” [2]. Physically, they represent energy from the far-field solution propagating into $\Omega$ [10]. Though in [2], it has been suggested by Mulder that for linear systems, the spurious modes contribute “to the numerical error that behaves in a reasonable manner, and that higher-order elements can be more accurate than lower-order elements.” Nonetheless, over the years several non-reflecting or pass-through BCs have been implemented to better model far-field solutions for various problems [9].
One of the main approaches to prescribe far-field BCs is by using characteristics or Riemann invariants [7]. This approach however is intrinsically best suited for one-dimensional (1D) problems. Thomas [11] was one of the first to use characteristics to examine multi-dimensional non-reflecting BCs for time dependent hyperbolic systems. Such an approach is useful for disturbances that propagate through the boundary at a 90 degree angle. However, using characteristics is not very efficient for problems where slanted perturbations (i.e., oblique incidences) hit the boundary [7].

Another technique is to use asymptotic expressions of the equations of motions for the far-field solution. Considering the asymptotic expansion allows for a more continuous transition as waves propagate through the boundary. Bayliss and Turkel [10] considered this approach for non-reflecting BCs for far-field radiation. Others have further extended such work. For instance, Bogey and Bailly formulated non-reflecting BCs for three-dimensional (3D) geometries for the linearized Euler equations. For the radiation condition however, linearizing the BC usually is not accurate enough when modeling physical wave phenomena [12]. To deal with this, a buffer zone or absorbing layer around \( \Omega \) can be considered as Colonius presents in [12]. In this buffer region, damping is applied to the far-field solution in which the ghost cells or points calculate the pressure gradient normal to the wall [12]. Regardless of the chosen approach to impose pass-through BCs, if the acoustic waves which reach \( \partial \Omega \) have a sufficiently small amplitude, the disturbance should leave \( \Omega \) with little to no reflections.

Another factor that influences the accuracy of pass-through BCs is the geometric shape of the boundary itself. In fact, one of the most challenging aspects when imposing pass-through BCs is how to deal with corners effectively [2]. In 3D this can be especially difficult since reflections can occur from multiple directions [13]. Therefore, depending on the problem being studied, using a sphere for instance as the \( \partial \Omega \) shape rather than a cube or rectangular prism may be more beneficial. Although, unless curved elements are considered at the outer boundary, it is likely that the mesh may need to be refined potentially slowing down run time. Furthermore, deciding on the appropriate geometry can be challenging for acoustic simulations through ducts (for instance, in musical instruments) because the “position and geometry of the domain boundary is governed by the shape of the duct” [14].

1.1 Overall Objectives

It is of interest to the author to study the production and propagation of finite-amplitude sound waves in brass musical instruments, specifically the trumpet. To simulate such wave propagation in 3D within a timely manner, it would be optimal to construct the smallest possible \( \Omega \). In addition, manipulating the geometric shape of \( \partial \Omega \) could further reduce run time while producing a more accurate solution. However, we would need to quantify how small the parameters of the outgoing waves must be upon interacting with \( \partial \Omega \) to ensure that the solution inside \( \Omega \) is not influenced by any reflections from the boundary as the waves exit the domain. This is precisely the the primary objective of this paper: to quantify how ‘small’ the amplitude of the outgoing waves must be. The second purpose of this paper is to determine whether there is an optimal geometric shape for the boundary of the computational domain, i.e., if reflections from \( \partial \Omega \) can be reduced by manipulating the geometry so that the angle of the outgoing waves is small relative to the boundary.

The numerical method that will be used for this investigation will be the DGM due to its usefulness and gaining popularity in aeroacoustic simulations. For instance, the DGM has been successfully used for acoustic simulations on the trumpet and bassoon [15] [16] [17] as well as acoustic pulse studies [15] [18] [19]. Since long simulations need to be run in 3D, serial computation is too slow. Instead, a graphical processing unit (GPU) implementation of the method is used. The GPU implementation used is described in [20] and takes advantage of the numerical method’s parallelization features. It was done using NVIDIA’s Compute Unified Device Architecture (CUDA). All simulations presented in this paper were run on the NVIDIA GeForce 750 Ti 2 GB graphics card. A brief summary of the numerical scheme is provided in Appendix A and a more detailed description can be found in [20] [28]. Numerical experiments presented in Section 3 were computed using a second order accurate linear approximation in space and verified by running the same tests with a quadratic basis. Due to the very fine meshes used, the results were similar. Runge-Kutta (RK2) discretization in time was used with the local Lax-Friedrichs Riemann solver.

1.2 Geometric Shapes Considered

Since the trumpet is one of the main instruments being studied by the author, a modified geometry shown in Fig. 1 will be considered in various \( \Omega \). The geometry of the shortened instrument was made to accurately model the slowly increasing diameter of the bell and corresponding cylindrical tube leading up to the flare. It was important to have a realistic representation of the trumpet bell because as mentioned in [13], the optimal
The geometry of the domain boundary will be governed by the shape of the wavefront which exists the duct. The portion of the waveform that radiates from the flare mostly contains higher frequency components and is roughly spherically symmetric with a curved wavefront [24]. However, there will be a bias as the energy leaves the flare: more will be focused along the central axis rather than along the edge of the flare.

Figure 1: The geometry of the modified trumpet used for the presented simulations.

The computational trumpet will be modeled with various domains. In particular, two sets of geometries for \( \Omega \) will be considered:

*Type A*: Domains that completely encapsulate the modified trumpet.

*Type B*: Domains that only have an area directly in front of the modified trumpet.

Figure 2: A side view of the dimensions of the computational domains.

The Type A geometries include a cube, a sphere, a cone, and a spherical cone. Whereas, the Type B geometries consist of a rectangular prism, a rotated hemisphere and a cone all to the right of the trumpet flare. The goal of examining these different volumes for \( \Omega \) is to address the following inquiries:

1. Is it advantageous to simulate a curved acoustic wavefront in a domain with spherical symmetry?

2. Is it necessary to consider \( \Omega \) to the left of the flare, or can it be neglected since only a small portion of energy propagates behind the bell?

With the same modified trumpet placed in the same position for all the \( \Omega \) considered, as shown in Fig. 2, seven different meshes were generated using the GMSH software. All meshes consist of tetrahedral elements with adaptive element sizes to accurately resolve the geometric features of the modified trumpet. The points that were used to trace out the trumpet boundary are depicted in Fig. 1. The trumpet shape in each domain
was created as follows: the cylindrical bore region connects two points with a line, while the remaining points outlining the flare were interpolated using cubic splines. A rotational extrusion about $x = 0$ was applied to the curve outlining the entire trumpet shape (above the central axis) to create the full 3D instrument. The meshes were scaled so that one unit on the axes represents 1 cm. The meshes constructed from the Type A and Type B geometries can be seen in Fig. 3 and Fig. 4 respectively. The names of each computational domain and corresponding number of elements can be found in Table 1.

Figure 3: A modified trumpet completely inside a cube, sphere, cone and spherical cone for $\Omega$. These are the Type A domains.

Figure 4: A rectangular prism, hemisphere and cone for $\Omega$ to the right of the trumpet flare. These are the Type B domains.
Table 1: Names of each computational domain and the corresponding number of elements.

| Geometry type | Domain Shape | Name of $\Omega$ | Number of elements |
|---------------|--------------|------------------|--------------------|
| Type A        | Cube         | $\Omega_{A\text{-}cube}$ | 207537             |
| Type A        | Sphere       | $\Omega_{A\text{-}sphere}$ | 198001             |
| Type A        | Cone         | $\Omega_{A\text{-}cone}$ | 214176             |
| Type A        | Spherical cone | $\Omega_{A\text{-}scone}$ | 194338             |
| Type B        | Rectangular prism | $\Omega_{B\text{-}half-cube}$ | 96805              |
| Type B        | Hemisphere   | $\Omega_{B\text{-}half-sphere}$ | 123849            |
| Type B        | Cone         | $\Omega_{B\text{-}half-cone}$ | 98146              |

2 Numerical Set-Up

2.1 Euler Equations
The mathematical model used in this paper was based from the 3D compressible Euler equations. The nonlinear equations are given by

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + P) \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E + P) \end{bmatrix} + \frac{\partial}{\partial z} \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E + P) \end{bmatrix} = 0 \quad (1)$$

where $\rho$ is the density of air; ($\rho u, \rho v, \rho w$) are the momenta in the $x$, $y$ and $z$ direction, respectively; $p$ is the internal pressure; and $E$ is the total energy. The equation of state for the ideal gas connects $E$ to the other variables and closes the system

$$E = \frac{p}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2 + w^2). \quad (2)$$

For air, the specific heat ratio $\gamma \approx 1.4$. The system (1) describes the motion of an inviscid, compressible gas. Hence, the model neglects viscosity of the air and wall losses in the trumpet. To solve equations (1 - 2), initial and boundary conditions need to be specified.

2.2 Initial Conditions and Pass-Through Boundary Condition
It was assumed initially that the flow was at rest. The remaining variables are scaled from physical values to ones more convenient for computation. In particular, the speed of sound $c_0$, which is approximately 343 m/s in air, and atmospheric pressure $p_0$, which is 101,325 Pa (i.e., one atmosphere), are scaled to be equal to 1. Assuming that the flow is isentropic, i.e., $c_0 = \sqrt{\frac{p}{\rho}}$, the initial density should then be taken to be 1.4. In summary,

$$\text{Initial State:} \begin{cases} p_0 = 1.0, \\ \rho_0 = 1.4, \\ u_0 = 0.0, \\ v_0 = 0.0, \\ w_0 = 0.0, \\ E_0 = 2.5. \end{cases} \quad (3)$$

Along the computational domain $\partial \Omega$, pass-through boundary conditions are used in which the ghost state is prescribed to be the initial state given in (3). If the computational domain is large enough, the amplitude of the wave should be small enough to completely pass through $\partial \Omega$ without being reflected back into the domain.

2.3 Solid Wall Boundary Conditions
On the inner and outer walls of the trumpet, excluding the left vertical boundary, reflective boundary conditions were prescribed. A ghost state is specified so that the normal velocity is reflected with respect to the wall, i.e.,
taken with a change of sign. The density, pressure and tangential velocity are unchanged from the corresponding value inside the cell. In summary,

\[
\text{Reflective Condition:} \begin{cases}
    p_{\text{right}} = p_{\text{left}}, \\
    p_{\text{right}} = p_{\text{left}}, \\
    u_{\text{right}} = u_{\text{left}} - 2u_{\text{left}} n_x, \\
    v_{\text{right}} = v_{\text{left}} - 2v_{\text{left}} n_y, \\
    w_{\text{right}} = w_{\text{left}} - 2w_{\text{left}} n_z, \\
    E_{\text{right}} = E_{\text{left}}.
\end{cases}
\]  

(4)

2.4 Inflow Conditions

At the left vertical boundary of the bore, the ghost state is specified to the be the inflow condition which will now be discussed. For all numerical test cases considered, a simple pressure pulse is generated at the left boundary of the modified instrument and is defined as

\[
p_1 = \begin{cases}
    1.0 + (0.05 - 0.05 \cos(\alpha t)), & \text{if } t < \frac{2\pi}{\alpha} \\
    1.0, & \text{else}
\end{cases}
\]

which corresponds to a unipolar pressure pulse with an amplitude of approximately 10,000 Pa where \(\alpha = 565\text{ Hz}\). In the left plot of Fig. 5, the generated pulse at the far left end of the cylindrical bore is shown. In the right plot of Fig. 5, the corresponding frequency spectrum is depicted.

![Simulated Pressure Pulse](image1)

![FFT of Simulated Pressure Pulse](image2)

Figure 5: Left: Initial pressure pulse in the time domain. Right: The corresponding pressure pulse in the frequency domain.

As this pulse propagates through the modified trumpet, a portion of the incident wave will be reflected before and within the bell and the other portion will be transmitted from the flare into \(\Omega\). As the transmitted wave propagates toward \(\partial \Omega\), its amplitude will drop off inversely proportional to the distance traveled. The purpose of considering such a large pulse is to ensure it does not completely diffuse before reaching the computational boundary.

The wave entering the bore is assumed to be planar. To relate pressure and velocity at the inlet boundary, the 1D expression derived from linear acoustic theory is used, i.e.,

\[
p - p_0 = \rho_0 c u. \tag{5}
\]

Note that this linearization is used only for the mouthpiece boundary condition, not inside the domain where the velocity is governed by (1 - 2). The density is prescribed by assuming the adiabatic relation between pressure and density from compressible flow theory, i.e.,

\[
\rho = C p^\gamma, \tag{6}
\]

where \(C = \gamma\) is the proportionality constant \(\gamma\).

Therefore, in summary, the dimensionless boundary condition at the mouthpiece of the computational trumpet
is given by

\[
\text{Inflow Condition: } \begin{cases} 
\dot{p} = p_1, \\
\dot{\rho} = \gamma \dot{p}, \\
\dot{u} = \frac{\rho_0 u_0}{p_0 c}, \\
\dot{v} = 0, \\
\dot{w} = 0, \\
\dot{E} = \frac{\dot{p}}{\gamma - 1} + \frac{\dot{\rho}}{2} (\dot{u}^2 + \dot{v}^2 + \dot{w}^2). 
\end{cases}
\]

(7)

3 Numerical Results and Discussion

The numerical results of solving (1) with the initial and boundary conditions on all the computational domains mentioned above will now be examined. For each simulation, the generated pressure was sampled at various positions throughout the computational domain. Although all the sensors’ data will not be shown here, Fig. 6 (left) illustrates the placement of the sensors on the 3D coordinate grid. The junction of the flare and room is positioned at \( x = 1.48 \) m and the central axis of the flare is at \( (x, y, z) = (1.48 \text{ m, 0 m, 0 m}) \). The flare reaches the height of \( y = 0.06738 \text{ m} \). For convenience, we will name the sensors that will be examined here. The position of these specific sensors are depicted in Fig. 6 (right) and the corresponding positions relative to the bell and boundary are listed in Table 2. In terms of Cartesian coordinates, the sensors are positioned at \((1.49 \text{ m, 0.99 m, 0 m})\) for sensor 1; \((2.48 \text{ m, 0 m, 0 m})\) for sensor 2; \((2.49 \text{ m, 0 m, 0 m})\) for sensor 3; and \((1.85 \text{ m, 0.57 m, 0 m})\) for sensor 4.

![Figure 6](image)

Figure 6: Left: A picture of where the sensors where placed in the computational domains, where the junction of the flare and room is positions at \( x = 1.48 \) m. Right: Names and corresponding positions of specific sensors.

Table 2: Location of specific sensors relative to the flare of the modified trumpet and the outer boundaries. The \((x,y,z)\) coordinates of the sensors are \((1.49 \text{ m, 0.99 m, 0 m})\) for sensor 1; \((2.48 \text{ m, 0 m, 0 m})\) for sensor 2; \((2.49 \text{ m, 0 m, 0 m})\) for sensor 3; and \((1.85 \text{ m, 0.57 m, 0 m})\) for sensor 4.

| Name   | Location from bell & boundary (x-axis) | Location from bell & boundary (y-axis) |
|--------|-------------------------------------|-------------------------------------|
| sensor 1 | 1 cm from bell, 101 cm from boundary | 99 cm from bell, 1 cm from boundary |
| sensor 2 | 100 cm from bell, 2 cm from boundary | 0 cm from bell, 100 cm from boundary |
| sensor 3 | 101 cm from bell, 1 cm from boundary | 0 cm from bell, 100 cm from boundary |
| sensor 4 | 37 cm from bell, 65 cm from boundary | 57 cm from bell, 43 cm from boundary |

For both geometry types, the portion of the pulse that will propagate out of the bell along the central axis has a maximum and minimum pressure peak of approximately 400 Pa and -400 Pa, respectively. This corresponds to a deviation from atmospheric pressure by approximately 0.3947%. However, only half that amount of energy travels vertically from the bell (i.e., from very top or bottom of the bell) compared to that along the central axis. Plots illustrating this in the time domain can be seen in Appendix B in Figs. 16 and 17. By the time the wavefront reaches the rightmost or uppermost boundary of \( \Omega \) (i.e., near sensor 1 and sensor 3), it has a
Figure 7: Simulation results in the frequency domain for the Type A geometries at sensor 1, sensor 2 and sensor 3 for \(\Omega_{A\text{-sphere}}\) (top left); \(\Omega_{A\text{-cone}}\) (top right); \(\Omega_{A\text{-scone}}\) (bottom left); and \(\Omega_{A\text{-cube}}\) (bottom right).

maximum and minimum pressure peak of approximately 10 Pa and -10 Pa, respectively. This is equivalent to 0.009869\% variation from atmospheric pressure. With these values in mind, we will now examine the numerical results near \(\partial \Omega\).

Figure 8: Simulation results in the frequency domain for the Type B geometries at sensor 1, sensor 2 and sensor 3 for \(\Omega_{B\text{-half-sphere}}\) (top); \(\Omega_{B\text{-half-cone}}\) (middle); and \(\Omega_{B\text{-half-cube}}\) (bottom).

In Figs. 7 and 8, the frequency spectra of the simulation results for the Type A and Type B geometries, respectively, at sensors 1 - 3 are shown. With the exception of \(\Omega_{B\text{-half-cone}}\), we observe that the spectrum for each \(\Omega\) at sensor 1 and sensor 3 are almost identical, especially for components less than 2200 Hz. The entire spectra however at sensor 1 and sensor 3 for \(\Omega_{A\text{-sphere}}\) and \(\Omega_{B\text{-half-sphere}}\) match for all the frequency components (which is not surprising because of the spherical symmetry). \(\Omega_{A\text{-cube}}\) and \(\Omega_{B\text{-half-cube}}\) also give extremely similar results at these sensor locations with slight deviations for a few of the components between 3000 Hz - 4000 Hz. We see the most discrepancy in the spectra for \(\Omega_{A\text{-scone}}\), \(\Omega_{B\text{-half-cone}}\) and \(\Omega_{A\text{-scone}}\) (though \(\Omega_{A\text{-scone}}\) is more accurate).
Figure 9: A comparison of the simulation results for the similar Type A (solid lines) and Type B (dotted lines) geometries at sensor 2 (black and blue line), sensor 3 (red and green) and sensor 4 (cyan and magenta) for $\Omega_{A\text{-sphere}}$ and $\Omega_{B\text{-half-sphere}}$ (top left); $\Omega_{A\text{-cone}}$ and $\Omega_{B\text{-half-cone}}$ (top right); $\Omega_{A\text{-cone}}$ $\Omega_{B\text{-half-cone}}$ (bottom left); and $\Omega_{A\text{-cube}}$ and $\Omega_{B\text{-half-cube}}$ (bottom right).

Figures 7 and 8 also depict the spectrum at sensor 2, which is 2 cm from the rightmost boundary along the central axis. Comparing sensors 2 and 3 can give some indication whether or not reflections occurred from $\partial \Omega$. In particular, for $\Omega_{A\text{-sphere}}$ and $\Omega_{A\text{-cone}}$, we observe that the spectra sampled 1 cm from the boundary is roughly 1-3 dB larger for components above 300 Hz relative to the spectra sampled 2 cm from the boundary. This implies that a portion of the pressure pulse was reflected from the boundary in the mentioned domains. Potentially this observation could be due to refinement since curved elements along the boundary were not considered. Reflections do not seem to be evident from any of the Type B geometries. However, the harmonic components corresponding to the $\Omega_{B\text{-half-cone}}$ solution are almost 20 dB higher than all other solutions. We will discuss some possible reasons for this shortly.

Figure 10: Simulation results for Type A (black solid lines) and Type B (dotted magenta lines) geometries inside the modified trumpet at the left vertical boundary after reflection.
In Fig. 9 a comparison of the solution curves obtained from sensors 2 - 4 are given for $\Omega_{A,\text{sphere}}$ and $\Omega_{B,\text{half-sphere}}$ (top left); $\Omega_{A,\text{cone}}$ and $\Omega_{B,\text{half-cone}}$ (top right); $\Omega_{A,\text{scone}}$ and $\Omega_{B,\text{half-cone}}$ (bottom left); and $\Omega_{A,\text{cube}}$ and $\Omega_{B,\text{half-cube}}$ (bottom right) is given. The Type A geometries are denoted with a solid line, whereas Type B is illustrated with a dotted line. We observe that for the most part, the data for both geometries match well for the sphere/hemisphere and cube/rectangular prism. However, $\Omega_{B,\text{half-cone}}$ does not align with $\Omega_{A,\text{cone}}$ or $\Omega_{A,\text{scone}}$.

Reflections seem to occur from the boundary of the domains with spherical symmetry, though this does not appear to greatly influence the solution inside $\Omega$. This observation can be made by examining the data sampled inside the modified instrument. If the simulated pressure for all $\Omega$ is consistent, this implies the numerical solutions were not influenced by any reflections that may have taken place. Thus, the frequency spectrum of each pressure pulse was sampled at the left vertical wall inside the bore after the disturbances were reflected from the bell. These results are shown in Fig. 10. In particular, solutions of $\Omega_{A,\text{sphere}}$ and $\Omega_{B,\text{half-sphere}}$ (top left); $\Omega_{A,\text{cone}}$ and $\Omega_{B,\text{half-cone}}$ (top right); $\Omega_{A,\text{scone}}$ and $\Omega_{B,\text{half-cone}}$ (bottom left); and $\Omega_{A,\text{cube}}$ and $\Omega_{B,\text{half-cube}}$ (bottom right) were juxtaposed. Although the $\Omega$’s with flat boundary walls (i.e., plane surfaces) match slightly better than the $\Omega$’s with curved boundary walls (i.e., ruled surfaces), there is little variation overall.

Next, we will examine the numerical solutions at the bell for each domain. If reflections from the boundary influence the numerical solution in $\Omega$, it will be more evident at the bell rather than inside the smaller cylindrical-tube portion. Figure 11 plots all the simulation results for both geometry types at the midpoint of the bell (top left), 1 cm outside the midpoint of the bell (top right), at the top of the bell (bottom left) and 1 cm outside the top of the bell (bottom right). We notice that all the simulation results seem to align with each other rather well. This should not be surprising for stable simulations because one would expect that the simulated pulse propagating out of the flare should be the same for each $\Omega$. We do however notice that the solution for $\Omega_{B,\text{half-cone}}$ deviates rather drastically from the other solutions for frequency components less than 500 Hz. This indicates that the solution in $\Omega_{B,\text{half-cone}}$ was influenced by reflections from the domain boundary. And we will now discuss some possible reasons why such reflections took place in $\Omega_{B,\text{half-cone}}$.

The angle of the cone in $\Omega_{B,\text{half-cone}}$ which extends from the flare (relative to the top of the bell) is 45 degrees. In the top right and bottom left plot of Fig. 9 we saw that the numerical solution of $\Omega_{B,\text{half-cone}}$ at sensor 4 (which is only 37 cm outside the bell) was approximately 40 dB higher than any of the other solutions. However, 100 cm outside the bell, the harmonic distribution was only 20 dB higher. This indicates that the amplitude of the pressure pulse in $\Omega_{B,\text{half-cone}}$ was increasing before it reached $\partial \Omega$. The only way for this to happen is if a portion of the disturbance was being reflected along the side of the cone. If this was the case, reflections would
occur in all directions due to spherical symmetry. It would then seem reasonable that the amplitude increases so quickly. If the cone instead followed the angle of the flare, such a computational domain may potentially give the best results because the propagating pressure would interact with the side of the cone at (roughly) a 90 degree angle. By the time the pressure pulse would reach the rightmost boundary, it would be equivalent to a plane wave propagating into a plane surface, i.e., it could propagate through the wall without reflections occurring.

4 Conclusion

A comparison of all the numerical results obtained from simulating a pressure pulse in the Type A and Type B geometries is shown in Fig. 12 (in addition to Fig. 11). In particular, the top left and top right plots in Fig. 12 shows all the Type A results at sensor 3 and at the left vertical boundary inside the bore, respectively. The bottom left and bottom right graphs illustrate the corresponding results for the Type B geometries.

![Figure 12: A comparison of all the simulation results for Type A geometries (top plots) and Type B geometries (bottom plots) at sensor 3 (left plots) and at left vertical boundary of the modified trumpet after reflection.](image)

With the exception of $\Omega_{B\text{-}\text{half-cone}}$, we see little variation between the two different computational domain types. Therefore, it does not seem necessary to construct $\Omega$ to encapsulate the entire instrument. This is very encouraging because this significantly reduces the required memory and run time. In fact, from examining Table 3 we can see that the number of cells and memory required is almost cut in half if the domain to the left of the flare is neglected. Furthermore, if the amplitude of the pressure pulse is less than 0.5% of atmospheric pressure, and the mesh is sufficiently refined, there is little to no reflections from the boundaries of the computational domain. More specifically, the reflections that do occur do not seem to greatly influence the solution inside of $\Omega$.

However, it is inconclusive from the results presented here that considering a computational domain with spherical symmetry is advantageous. But, the spherically symmetric domains that were considered were not well refined (i.e., the refinement was the same for all geometries for easier comparison) and curved elements along $\partial \Omega$ were not used. If the meshes for $\Omega\text{sphere}$, $\Omega_{B\text{-}\text{half-sphere}}$ and $\Omega_{A\text{-}\text{cone}}$ were not so coarse, better results may have been produced. Further simulations will have to be done using a GPU with more memory.

Acknowledgment

This research was supported in part by the Alexander Graham Bell PGS-D grant 365873. Also, thank you to Andrew Giuliani for his implementation of the DG method used for the simulations presented in this paper.

References

[1] B. Cockburn & C. W. Shu (1989). “TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework.” Mathematics of Computation, 52(186):411-435.
Table 3: Required memory and number of cells for each computational domain.

| Computational domain | Number of elements | Required memory     |
|----------------------|-------------------|--------------------|
| $\Omega_A$-cube      | 207537            | 212.857992 MB      |
| $\Omega_A$-sphere    | 198001            | 206.619888 MB      |
| $\Omega_A$-cone      | 214176            | 222.658788 MB      |
| $\Omega_A$-scone     | 194338            | 204.042924 MB      |
| $\Omega_B$-half-cube | 96805             | 96.623664 MB       |
| $\Omega_B$-half-sphere | 128349           | 137.651184 MB      |
| $\Omega_B$-half-cone | 98146             | 98.348604 MB       |

[2] W. A. Mulder (1999). “Spurious modes in finite-element discretizations of the wave equation may not be all that bad.” *Applied Numerical Mathematics*, **30**(4): 425-445.

[3] B. Cockburn & C. W. Shu (1998). “The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems.” *J. Computational Physics* **141**: 119-224.

[4] J. Yan & C. W. Shu (2002). “Local discontinuous Galerkin methods for partial differential equations with higher order derivatives.” *J. Scientific Computing*, **17**: 27-47.

[5] M. Fuhrty (2013). “An implementation of the discontinuous Galerkin method on graphics processing units.” Thesis.

[6] J. R. Dea, F. X. Giraldo & B. Neta (2009). “High-order non-reflecting boundary conditions for the linearized 2-D Euler equations: no mean flow case.” *Wave Motion* **46**(3): 210-220.

[7] C. Bogey, & C. Bailly (2002). “Three-dimensional non-reflective boundary conditions for acoustic simulations: far field formulation and validation test cases.” *Acta Acustica united with Acustica*, **88**(4): 463-471.

[8] I. Spisso & A. Rona. “Acoustically non-reflecting boundary conditions.” *University of Leicester*. [http://www2.le.ac.uk/departments/engineering/people/academic-staff/aldo-rona/research/methods/computational-aeroacoustics/boundary-conditions](http://www2.le.ac.uk/departments/engineering/people/academic-staff/aldo-rona/research/methods/computational-aeroacoustics/boundary-conditions)

[9] D. Appel’o (2003). “Non-reflecting Boundary Conditions for Wave Propagation Problems.” Thesis.

[10] A. Bayliss & E. Turkel (1982). “Far field boundary conditions for compressible flows.” *J. Computational Physics* **48**(2): 182-199.

[11] K. W. Thompson (1987). “Time dependent boundary conditions for hyperbolic systems.” *J. Computational Physics* **68**(1): 1-24.

[12] T. Colonius (2004). “Modeling artificial boundary conditions for compressible flow.” *Annu. Rev. Fluid Mechanics* **36**: 315-345.

[13] M. B. Schlaffer (2013). “Non-reflecting boundary conditions for the lattice Boltzmann method.” Thesis.

[14] S. K. Richards, X. Zhang, X. X. Chen, & P. A. Nelson (2004). “The evaluation of non-reflecting boundary conditions for duct acoustic computation.” *J. Sound and Vibration* **270**(3): 539-557.

[15] J. Resch, K. Krivodonova, & J. Vanderkooy (2014). “A two-dimensional study of finite amplitude sound waves in a trumpet using the discontinuous Galerkin method,” *J. Computational Acoustics*, **22**(3): 27 pages.

[16] J. Resch, K. Krivodonova, & J. Vanderkooy (2016). “A comparison between two and three-dimensional simulations of finite amplitude sound waves in a trumpet.” In *Mathematical and Computational Approaches in Advancing Modern Science and Engineering*. Springer International Publishing, 481-492.

[17] A. Richter, J. Stiller, & R. Grundmann (2007). “Stablized discontinuous Galerkin method for flow-sound interaction.” *J. Computational Acoustics* **15**(1): 123-143.

[18] A. V.Wolkov & N. B. Petrovskaya (2010). “Higher order discontinuous Galerkin method for acoustic pulse problem,” *Computer Physics Communications* **181**(7): 1186-1194.
[19] R. R. Erickson & B. T. Zinn (2003). “Modeling of finite amplitude acoustic waves in closed cavities using the Galerkin method.” *J. Acoustical Society of America* **113**(4): 1863-1870.

[20] M. Fuhry, A. Giuliani & L. Krivodonova (2014). “Discontinuous Galerkin methods on graphics processing units for nonlinear hyperbolic conservation laws.” *International Journal for Numerical Methods in Fluids* **76**(12): 982-1003.

[21] C. K. W. Tam, & Z. Dong (1996). “Radiation and outflow boundary conditions for direct computation of acoustic and flow disturbances in a nonuniform mean flow.” *J. Computational Acoustics* **4**(2): 175-201.

[22] V. Pagneux, N. Amir, & J. Kergomard (1996). “A study of wave propagation in varying cross-section waveguides by modal decomposition. Part I. Theory and validation.” *J. Acoustic Society of America* **100**(4): 2034-2048.

[23] J. E. Flaherty, L. Krivodonova, J. F. Remacle & M. S. Shephard (2002). “Some aspects of discontinuous Galerkin methods for hyperbolic conservation laws.” *J. Finite Elements in Analysis and Design* **38**(10): 889-908.

[24] A. H. Benade, *Fundamentals of Musical Acoustics* (Dover Publications, New York, 1900).

[25] Liepmann, H. W., & Roshko, A., (2001). *Elements of Gasdynamics* (Dover Publications, Mineola, NY).

[26] L. Krivodonova & M. Berger (2002). “High-order accurate implementation of solid wall boundary conditions in curved geometries.” *J. Computational Physics* **38**(2): 492-512.
Appendix A: The Discontinuous Galerkin method

The equations of motion found in Eq. (1) form a hyperbolic system of conservation laws which we solved numerically using the discontinuous Galerkin method (DGM). In order to describe the method, we write a general conservation law

\[
\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0, \quad x \in \Omega, \quad t > 0, \tag{8a}
\]

\[
u = u^0, \quad t = 0, \tag{8b}
\]

with the solution \( u(x, t) = (u_1, u_2, ..., u_m)^t \), where \( (x, t) \in \Omega \times [0, T] \), and numerical flux \( F(u) \). The computational domain \( \Omega \) is partitioned into a collection of nonoverlapping elements

\[
\Omega = \bigcup_{j=1}^{N_h} \Omega_j. \tag{9}
\]

We then construct a Galerkin problem on element \( \Omega_j \) by multiplying Eq. (8a) by a test function \( v \in (H^1(\Omega_j))^m \), where \( m \) is the number of equations in the system, integrating the result on \( \Omega_j \), and using the divergence theorem to obtain

\[
\int_{\Omega_j} v \frac{\partial u}{\partial t} ds + \int_{\partial \Omega_j} v F(u) \cdot \hat{n} d\tau - \int_{\Omega_j} \nabla v \cdot F(u) ds = 0, \quad \forall v \in (H^1(\Omega_j))^m, \tag{10}
\]

where \( \hat{n} \) is the unit outward normal vector to \( \partial \Omega_j \).

The solution \( u(x, t) \) on \( \Omega_j \) is approximated by a vector function \( U_j = (U_j,1, U_j,2, ..., U_j,m)^t \), where

\[
U_{j,k} = \sum_{i=1}^{N_p} c_{j,k,i} \varphi_i, \quad k = 1, 2, ..., m, \tag{11}
\]

is a finite-dimensional subspace of the solution space. The basis \( \{ \varphi_i \}_{i=1}^{N_p} \) is chosen to be orthonormal in \( L^2(\Omega_j) \).

Due to the discontinuous nature of the numerical solution, the normal flux \( F_n = F(u) \cdot \hat{n} \), is not defined on \( \partial \Omega_j \). The usual strategy is to define it in terms of a numerical flux \( F_n(U_j, U_k) \) that depends on the solution \( U_j \) on \( \Omega_j \) and \( U_k \) on the neighboring element \( \Omega_k \) sharing the portion of the boundary \( \partial \Omega_{jk} \) common to both element [23]. Finally, the \( L^2 \) volume and surface inner products in Eq. (10) are computed using \( 2p \) and \( 2p + 1 \) order accurate Gauss quadratures [23], respectively, where \( p \) is the order of the orthonormal basis. The resulting system of ODEs is

\[
\frac{dc}{dt} = f(c), \tag{12}
\]

where \( c \) is the vector of unknowns and \( f \) is a nonlinear vector function resulting from the boundary and volume integrals in Eq. (10).

Appendix B: Additional Plots

Below are some additional plots and larger versions of some of the figures above.
Figure 13: Close up: Comparison of all the simulation results for Type A geometries (top plots) and Type B geometries (bottom plots) at sensor 3.
Figure 14: Close up: Comparison of all the simulation results for Type A geometries (top plots) and Type B geometries (bottom plots) at left vertical boundary of the modified trumpet after reflection.
Figure 15: Simulation results for Type A (black solid lines) and Type B (dotted magenta lines) geometries inside the modified trumpet at the left vertical boundary after reflection where the spectra are plotted from frequencies to 8000 Hz.
Figure 16: A comparison of all the simulation results at the bell in the time domain for the Type A geometries (top) and the Type B geometries (bottom)
Figure 17: A comparison of all the simulation results 1cm outside the bell in the time domain for the Type A geometries (top) and the Type B geometries (bottom)